WYMAGANIA EDUKACYJNE NA POSZCZEGÓLNE OCENY  
Z INFORMATYKI ROZSZERZONEJ

**Ocenianie za pomocą stopni szkolnych ma na celu:**

1. Ocenianie osiągnięć edukacyjnych ucznia z informatyki rozszerzonej polega na rozpoznaniu przez nauczyciela poziomu i postępów w opanowaniu przez ucznia wiadomości i umiejętności w stosunku do wymagań określonych w podstawie programowej i wynikających z realizowanego w danym oddziale programu nauczania.
2. Na początku roku szkolnego nauczyciel informuje uczniów o wymaganiach edukacyjnych niezbędnych do otrzymania przez ucznia ocen bieżących i końcowych z informatyki rozszerzonej, określonych w podstawie programowej i wynikających z realizowanego przez siebie programu nauczania, o sposobach sprawdzania osiągnięć edukacyjnych oraz o warunkach i trybie otrzymania wyższej niż przewidywana rocznej oceny klasyfikacyjnej.
3. Nauczyciel indywidualizuje pracę z uczniem na zajęciach edukacyjnych i dostosowuje wymagania edukacyjne odpowiednio do potrzeb rozwojowych i edukacyjnych oraz możliwości psychofizycznych ucznia.

Przedmiotem oceniania w całym roku szkolnym są następujące obszary aktywności ucznia:

* sprawdzian — obejmuje większy zakres przerobionego materiału (na przykład dział), zapowiadana jest co najmniej z tygodniowym wyprzedzeniem, może być poprawiana jeden raz; może być praktyczna (przy komputerze) lub teoretyczna (bez użycia komputera);
* kartkówka— obejmuje materiał dwóch, trzech ostatnich lekcji; może być praktyczny (przy komputerze) lub teoretyczny (bez użycia komputera);
* projekt indywidualny — indywidualnie wykonana prezentacja multimedialna, strona internetowa, program komputerowy bądź inna forma pracy zrealizowanej przez ucznia z wykorzystaniem komputera;
* odpowiedź ustna — obejmuje zagadnienia dotyczące aktualnie realizowanego materiału;
* referat — praca opisowa na określony temat;
* praca na lekcji — aktywność na lekcji, ćwiczenia na lekcji;
* praca w domu — zadanie domowe, praca dodatkowa wykonana przez ucznia w domu;
* projekt zespołowy — praca nad projektem grupowym realizowanym pod kierunkiem nauczyciela; ocena uzależniona jest zarówno od efektu końcowego dotyczącego zrealizowanego zadania, jak i umiejętności współdziałania w grupie oraz zaangażowania wszystkich uczestników grupy;
* korelacja między przedmiotowa — realizacja projektów z innych przedmiotów z wykorzystaniem narzędzi informatycznych; nauczyciel informatyki ocenia wykonaną pracę od strony informatycznej, natomiast nauczyciel przedmiotu, którego dotyczy wykonywana praca, sprawdza efekty działań ucznia od strony tego przedmiotu.

**Obowiązują następujące kryteria dla poszczególnych ocen:**

* stopień celujący otrzymuje uczeń, który: spełnia wymagania na stopień bardzo dobry oraz posiada wiedzę i umiejętności znacznie wykraczające poza program nauczania przedmiotu w danej klasie, samodzielnie i twórczo rozwija własne uzdolnienia, biegle posługuje się zdobytymi wiadomościami w rozwiązywaniu problemów, proponuje rozwiązania nietypowe, rozwiązuje także zadania wykraczające poza program. Posiada wiedzę wykraczającą poza zakres materiału programowego, umie samodzielnie zdobywać wiedzę z różnych mediów, wykazuje inicjatywę rozwiązywania konkretnych problemów w czasie lekcji i pracy pozalekcyjnej, wykonuje z własnej inicjatywy dodatkowe prace, wykonuje prace na rzecz szkoły i pracowni (np. witryny Web). Jego wypowiedzi są przemyślane i nie zawierają żadnych błędów. Chętnie bierze udział i osiąga sukcesy w konkursach i olimpiadach przedmiotowych, kwalifikuje się do finałów na szczeblu wojewódzkim (regionalnym) albo krajowym lub posiada inne porównywalne osiągnięcia;
* stopień bardzo dobry otrzymuje uczeń, który: opanował cały zakres wiedzy i umiejętności określonym programem nauczania przedmiotu w danej klasie, sprawnie posługuje się zdobytymi wiadomościami, rozwiązuje samodzielnie problemy informatyczne lub z zakresu TI. Dobrze zna pojęcia informatyczne i swobodnie je stosuje. Pracuje samodzielnie w zakresie zadań teoretycznych oraz praktycznych. Na lekcje przychodzi przygotowany i bierze w niej czynny udział;
* stopień dobry otrzymuje uczeń, który: nie opanował w pełni wiadomości określonych programem w danej klasie, ale opanował je na poziomie przekraczającym wymagania zawarte w podstawach programowych, poprawnie stosuje wiadomości, rozwiązuje (wykonuje) samodzielnie typowe zadania z zakresu informatyki i TI. Wie, czym zajmuje się informatyka, zna podstawowe pojęcia informatyczne, umie sprawnie komunikować się z komputerem swobodnie posługując się oprogramowaniem użytkowym do wykonywania typowych zadań. Jednak w wypowiedziach oraz ćwiczeniach praktycznych popełnia sporadyczne błędy;
* stopień dostateczny otrzymuje uczeń, który: opanował wiadomości i umiejętności określone programem nauczania w danej klasie na poziomie nieprzekraczającym wymagana zawartych w podstawach programowych, rozwiązuje (wykonuje) zadania teoretyczne i praktyczne o średnim stopniu trudności. Zna podstawowe pojęcia informatyczne, wykonuje proste zadania, umie uruchomić programy komputerowe. Jednak ma problemy z samodzielna pracą i bez pomocy nauczyciela gubi się w toku lekcyjnym. W wypowiedziach oraz ćwiczeniach popełnia błędy;
* stopień dopuszczający otrzymuje uczeń, który: ma braki w opanowaniu materiału programowego określonego w podstawach programowych, ale braki te nie przekraczają możliwości uzyskania przez ucznia podstawowej wiedzy z danego przedmiotu w ciągu dalszej nauki, rozwiązuje (wykonuje) zadania teoretyczne i praktyczne o niewielkim stopniu trudności. Uczeń umie komunikować się z komputerem w elementarnym zakresie. Na lekcjach wymaga ciągłego nadzoru ze strony nauczyciela, popełnia poważne błędy;
* stopień niedostateczny otrzymuje uczeń, który: nie opanował wiadomości i umiejętności określonych w wymaganiach edukacyjnych, a braki w wiadomościach uniemożliwiają dalsze zdobywanie wiedzy z tego przedmiotu (w przypadku informatyki), nie jest w stanie rozwiązać (wykonać) zadań o elementarnym stopniu trudności. Nie zna pojęć informatycznych występujących w materiale nauczania, nie umie stosować posiadanych wiadomości do wykonywania elementarnych czynności praktycznych w bardzo prostych sytuacjach, nie rozumie pytań i poleceń, nie umie uruchamiać programów komputerowych. W wypowiedziach popełnia bardzo poważne błędy, które uniemożliwiają dalszą jego pracę.

Wystawianie ocen klasyfikacyjnych

* Ocenę roczną wystawia nauczyciel na podstawie uzyskanych ocen w ciągu całego roku.
* Termin wystawienia ustalany jest przez dyrektora szkoły stosownym zarządzeniem.
* O zagrożeniu oceną niedostateczną nauczyciel informuje ucznia i jego rodziców poprzez wychowawcę klasy na miesiąc przed klasyfikacją.

Stosowane są kryteria punktowe przeliczane na ocenę według tabeli:

|  |  |
| --- | --- |
| **Nazwa oceny** | **Progi procentowe** |
| celujący | 98 - 100% |
| bardzo dobry | 90 - 97% |
| dobry | 75 - 89% |
| dostateczny | 55 - 74% |
| dopuszczający | 40 - 54% |
| niedostateczny | 0 - 39% |

**Dostosowanie wymagań edukacyjnych do możliwości psychofizycznych wynikających z indywidualnych potrzeb uczniów**

Wobec uczniów ze specjalnymi potrzebami edukacyjnymi konieczne jest dostosowanie wymagań edukacyjnych oraz form i metod pracy, w zależności od indywidualnych potrzeb oraz zaleceń poradni psychologiczno – pedagogicznej lub lekarza. Dostosowanie wymagań nie oznacza ich obniżenia i dotyczy głównie form i metod pracy z uczniem. Nie może prowadzić do zejścia poniżej podstawy programowej, a zakres wiedzy i umiejętności powinien dać szansę na sprostanie wymaganiom kolejnego etapu edukacyjnego.

Sposoby dostosowania wymagań edukacyjnych:

1. **uczeń u którego stwierdzono zaburzenia i odchylenia rozwojowe lub specyficzne trudności w uczeniu się:**
2. stworzenie przyjaznej, spokojnej atmosfery
3. dawać więcej czasu na opanowanie danej umiejętności
4. w trakcie wykonywania ćwiczeń praktycznych sprawdzać, czy uczeń prawidłowo zrozumiał instrukcje, w razie potrzeby udzielać dodatkowych wskazówek
5. materiał sprawiający trudność dłużej utrwalać, dzielić na mniejsze porcje
6. pomoc w odczytywaniu testów ze zrozumieniem, prowadzeniu notatek oraz wykonywaniu ćwiczeń na lekcji (wpisywaniu danych do komputera)
7. częste odwoływanie się do konkretu
8. podawanie poleceń w prostszej formie ( dzielenie złożonych treści na proste, bardziej zrozumiałe części)
9. proszenie ucznia o powtórzenie poleceń
10. rozbijanie dużych partii materiału/poleceń na mniejsze
11. unikanie pytań problemowych, przekrojowych
12. przypominanie o terminowych zadaniach
13. posadzenie ucznia w pracowni komputerowej blisko nauczyciela w celu lepszego kontaktu nauczyciela z uczniem
14. zapewnienie poczucia bezpieczeństwa.
15. **uczeń zdolny:**
16. poszerzenie zainteresowań i zdolności ucznia poprzez udział w przedsięwzięciach szkolnych i pozaszkolnych
17. indywidualizacja procesu dydaktycznego
18. przygotowanie ucznia do konkursów i olimpiad przedmiotowych
19. krótkie, kilkuminutowe rozmowy z uczniem, zwykle komentujące w sposób rozszerzający bieżący materiał lub kończące się formułowaniem problemu, a potem rozwiązaniem go
20. zadawanie dodatkowych zadań podczas prac klasowych, do domu
21. zachęcanie ucznia do realizacji dodatkowych projektów
22. zachęcanie do czytania fachowej literatury, czasopism
23. praca w grupach z uczniami o podobnym poziomie uzdolnień, gdzie zadawane są zadania trudniejsze
24. praca w grupach, gdzie uczniowie zdolni pełnią rolę liderów.

**Uwagi końcowe**

1. Każdy uczeń oceniany jest z zasadami sprawiedliwości.
2. Prace sprawdzające z każdego działu i odpowiedzi ustne są obowiązkowe.
3. Prace klasowe są zapowiadane, z co najmniej tygodniowym wyprzedzeniem, podany jest zakres sprawdzanych umiejętności i wiedzy i poprzedzone są utrwaleniem materiału.
4. Uczeń nieobecny na pracy klasowej jest zobowiązany, zaraz po powrocie do Szkoły, umówić się z nauczycielem na zaliczenie określonego zakresu materiału w terminie nie dłuższym 10 dni roboczych następujących po powrocie ucznia.
5. Pracę klasową uczeń może poprawić. Poprawa jest dobrowolna i odbywa się w ciągu 2 tygodni od dnia podania informacji o ocenach. Uczeń poprawia pracę tylko raz i brana jest pod uwagę ocena z pracy poprawionej.
6. Nauczyciel ma prawo przerwać sprawdzian uczniowi lub całej klasie, jeżeli stwierdzi, że zachowanie ucznia (uczniów) nie gwarantuje samodzielności pracy. Stwierdzenie faktu odpisywania podczas sprawdzianu pisemnego jest podstawą ustalenia oceny niedostatecznej.
7. Uczeń ma wgląd do pracy na lekcji, na której są oddawane prace, a rodzice w dowolnym czasie pobytu w szkole.
8. Na każdej lekcji można sprawdzić ustnie lub pisemnie wiadomości ucznia z bieżącego materiału.
9. Nieobecność ucznia na lekcji zobowiązuje go do niezwłocznego uzupełnienia materiału we własnym zakresie.
10. Uczeń ma prawo 2 razy być nieprzygotowany do lekcji w ciągu okresu bez uzasadniania przyczyny, jeżeli na dane zajęcia edukacyjne przypada minimum 2 godziny tygodniowo. Jeżeli przypada jedna godzina tygodniowo – to 1 nieprzygotowanie. Nieprzygotowanie obejmuje także zadania domowe oraz brak zeszytu.
11. Uczeń może korzystać z kalkulatora prostego. Zabrania się korzystania z telefonu komórkowego jako kalkulatora.
12. Przy ocenianiu nauczyciel uwzględnia możliwości intelektualne ucznia, wkład pracy   
    i zaangażowanie oraz opinie z Poradni Psychologiczno – Pedagogicznej.
13. Egzaminy poprawkowe i klasyfikacyjne są przeprowadzane zgodnie z §135 Statutu Szkoły.

**Tryb i warunki uzyskania wyższej niż przewidywana rocznej oceny z zajęć edukacyjnych**

1. Za przewidywaną ocenę roczną przyjmuje się ocenę zaproponowaną przez nauczyciela zgodnie z terminem ustalonym w Statucie Szkoły.
2. Uczeń może ubiegać się o podwyższenie przewidywanej oceny tylko o jeden stopień i musi spełniać warunki:
   1. jego frekwencja na zajęciach była nie niższa niż 55% (z wyjątkiem długotrwałej choroby);
   2. przystąpił do wszystkich przewidzianych przez nauczyciela form sprawdzianów i prac pisemnych również w trybie uzupełniania braków;
   3. skorzystał z wszystkich oferowanych przez nauczyciela form poprawy, w tym – konsultacji indywidualnych.

WYMAGANIA NA POSZCZEGÓLNE OCENY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Temat** | **Ocena dopuszczająca** | **Ocena dostateczna** | **Ocena dobra** | **Ocena bardzo dobra** | **Ocena celująca** |
| **I.** **Przestrzeganie prawa i zasad bezpieczeństwa podczas pracy przy komputerze** | | | | | |
| Bądź uczciwy, czyli przestrzeganie prawa w świecie informatyki | ‒ wie, że udostępnianie treści chronionych prawem autorskim jest przestępstwem  ‒ podaje przykłady łamania praw autorskich  ‒ szanuje własność intelektualną | ‒ wie, kiedy nie narusza prawa podczas korzystania z utworów z sieci  ‒ wie, czym charakteryzuje się licencja CC i na jakich zasadach można używać takich dzieł  ‒ określa, czym w świetle prawa jest utwór | ‒ wskazuje legalne źródła muzyki, grafiki, animacji itp. rozpowszechnianej na licencji CC  ‒ wskazuje różnice pomiędzy plagiatem a cytatem | ‒ określa i przedstawia zasady legalnego korzystania z dzieł objętych prawami autorskimi | ‒ omawia niektóre metody działania instytucji i kancelarii prawnych w zakresie ścigania osób łamiących prawo autorskie |
| 8, 16, 32, 64, czyli jak rozwój technologii wpływa na rozwój społeczeństw | ‒ wie, że istnieją inne systemy liczbowe poza dziesiętnym i tłumaczy ich zastosowanie  ‒ zna pojęcia *bajt* i *bit*  ‒ wie, jak powstają wagi poszczególnych pozycji w kodzie binarnym  ‒ wie, jaki wpływ na zastosowanie komputerów ma postęp technologiczny | ‒ umie wykorzystać kalkulator do prezentacji liczb w różnych systemach liczbowych  ‒ omawia zalety zdalnego nauczania i jego wpływu na rozwój społeczny  ‒ podaje przykłady wpływu postępu technologicznego na rozwój informatyki | ‒ wie, dlaczego do projektowania układów komputera używa się kodu dwójkowego  ‒ omawia zalety zdalnego nauczania i jego wpływu na rozwój społeczny  ‒ omawia wpływ rozwoju technologii informacyjnych na rozwój społeczeństw | ‒ omawia zmiany technologiczne poszczególnych elementów komputerów i ich wpływ na zastosowanie komputerów | ‒ analizuje tendencje rozwoju społeczeństwa w kontekście rozwoju technologicznego |
| Chrońmy informacje, czyli techniki uwierzytelniania, kryptografia i podpis elektroniczny | - wie czym są techniki uwierzytelniania,  - wie czym jest kryptografia  - wie czym jest i gdzie jest wykorzystywany podpis elektroniczny | - zna techniki uwierzytelniania  - zna kody jedno- i dwuczęściowe  - zna kody jednorazowe  - zna superszyfrowanie  - zna podpis cyfrowy | - umie rozróżnić techniki uwierzytelnienia w zależności od tego, co jest uwierzytelnianie  - omawia kody jedno- i dwuczęściowe,  - omawia kody jednorazowe  - omawia superszyfrowanie  - omawia podpis cyfrowy | - zna i stosuje zasady uwierzytelniania dwuskładnikowego,  - określa techniki kryptografii  - zna i omawia kodowanie i szyfrowanie  - omawia procedury prawne podpisu elektronicznego  - stosuje podpis elektroniczny w życiu codziennym | - stosuje łącznie dwie różne techniki uwierzytelniania |
| Praktycznie i teoretycznie, czyli trendy historyczne w rozwoju technologii | - zna rozwój technologii informacyjnej | - zna wpływ rozwoju sprzętu komputerowego na rozwój technologii  - określa jaki miał wpływ rozwoju technologii na rozwój e-usług | - omawia historię rozwoju technologii informacyjnej  - omawia rozwój e-usług  - wskazuje różnice w rozwoju sprzętu komputerowego | - omawia wpływ rozwoju technologii informacyjnej na życie człowieka  - opracowuje możliwości wykorzystania e-usług w życiu codziennym  - zna zagrożenia w komunikacji interpersonalnej wynikające z rozwoju nowych technologii | - zna rozwój możliwości komunikowania się za pomocą nowych technologii oraz zagrożenia, jaki ten rozwój niesie za sobą |
| Kim jestem, czyli jak bezpiecznie budować wizerunek w sieci | ‒ określa, czym są przepisy oparte na RODO i jaki jest cel ich wprowadzenia | ‒ określa, na czym polegają prawa obywatela do ochrony wizerunku i wskazuje źródła prawa  ‒ określa, czym grozi upowszechnianie wizerunku bez zgody danej osoby  ‒ wie, co zrobić w przypadku wykrycia naruszenia swoich praw do wizerunku | ‒ umie opisać cyberzagrożenia i wskazać najgroźniejsze z punktu widzenia przepisów o ochronie wizerunku  ‒ omawia zasady bezpiecznego korzystania z sieci i usług sieciowych w kontekście ochrony własnego wizerunku i niewykorzystywania cudzego bez odpowiedniej zgody | ‒ wymienia działania, które służą ochronie wizerunku w sieci, i wie, jak nie naruszać tych praw  ‒ opracowuje własne zasady ochrony wizerunku na podstawie przepisów prawa  ‒ wie, czym może skutkować kradzież tożsamości | ‒ określa i objaśnia kolegom na lekcji zagrożenia płynące z możliwości kradzieży tożsamości w kontekście oszustw i wyłudzeń |
| **II. Podstawy programowania i środowisko programistyczne** | | | | | |
| Przypomnij sobie, czyli podstawy tworzenia algorytmów | ‒ wie, ze algorytm można zapisać w różnych postaciach i wymienia nazwy tych sposobów  ‒ zna podstawowe zasady tworzenia schematów blokowych w tym dozwolone i niedozwolone połączenia | ‒ prawidłowo interpretuje działanie bloku decyzyjnego i wie, jaką rolę odgrywa on w algorytmie  ‒ wie, jaka jest różnica pomiędzy blokiem decyzyjnym a wejściowym/wyjściowym | ‒ samodzielnie analizuje przykład algorytmu, np. z podręcznika z pomocą opisu  ‒ buduje algorytmy prostych zagadnień z różnych dziedzin lub przedmiotów szkolnych  ‒ umie ułożyć prosty algorytm w postaci schematu blokowego na podstawie algorytmu typu lista kroków zawierający blok decyzyjny | ‒ samodzielnie tworzy algorytmy na podstawie specyfikacji zawierające bloki decyzyjne  ‒ samodzielnie testuje algorytmy dla różnych przypadków  ‒ samodzielnie dyskutuje sposób rozwiązania problemu na podstawie algorytmu, np. z podręcznika, i proponuje jego modyfikacje | ‒ analizuje różne algorytmy i testuje je na samodzielnie i odpowiednio dobranych danych  ‒ samodzielnie proponuje modyfikacje przykładowych algorytmów |
| Algorytm, czyli utrwalamy swoją wiedzę | - wie czym jest schemat blokowy  - wie co to jest pseudokod  - wie czym jest algorytm zapisany w formie listy kroków | - umie zapisać algorytm wykorzystując schemat blokowy  - umie zapisać algorytm wykorzystując listę kroków  - umie zapisać algorytm przy pomocy pseudokodu | - stosuje tworzenie algorytmów w postaci schematu blokowego  - stosuje tworzenie algorytmów w postaci listy kroków  - stosuje tworzenie algorytmów w postaci pseudokodu | - samodzielnie analizuje złożone algorytmy zapisane w dowolnej postaci | - samodzielnie tworzy, analizuje i testuje algorytmy dla bardziej złożonych problemów |
| Pozycyjne reprezentacje liczb, czyli tablice w algorytmie | - wie co to są liczby doskonałe  - wie czym są tablice  - wie czym jest schemat Hornera | - opisuje czym jest doskonałość  - umie nazwać czym są liczby doskonałe  - opisuje czym są tablice | - umie tworzyć algorytmy z wykorzystaniem tablic  - umie tworzyć algorytmy z wykorzystaniem liczb doskonałych, parzystych i pierwszych | - umie tworzyć algorytmy wykorzystujące schemat Hornera  - - umie tworzyć algorytmy z wykorzystaniem pozycyjnych reprezentacji liczb | - samodzielnie tworzy, analizuje i testuje algorytmy dla bardziej złożonych problemów wykorzystujących schemat Hornera oraz liczby doskonałe |
| Narzędzia, czyli jak dobrać i skonfigurować środowisko programistyczne | ‒ wymienia nazwy kilku języków programowania  ‒ wie, gdzie sprawdzić poziom popularności niektórych z nich  ‒ wie, co powinno wpływać na dobór odpowiedniego języka programowania | ‒ zna podstawowe cechy języków JAVA i C++  ‒ wie, czym jest środowisko programistyczne  ‒ umie edytować kod programu w Eclipse | ‒ umie instalować Eclipse i przystosować środowisko do danego języka programowania | ‒ umie uruchamiać w Eclipse program ułożony w JAVA  ‒ umie uruchomić polski pakiet językowy dla Eclipse | ‒ umie omówić działanie poszczególnych linii przykładowego kodu programu w JAVA |
| C z plusami, czyli uruchamiamy pierwsze programy w C++ | ‒ wie, że Eclipse należy przystosować do układania programów w C++ poprzez zainstalowanie pakietu MinGW | ‒ z pomocą nauczyciela instaluje Eclipse dla C++ i pakiet MinGW  Na podstawie podręcznika:  ‒ umie utworzyć nowy projekt w C++ w IDE Eclipse  ‒ uruchamia program testowy Hello World | Na podstawie podręcznika:  ‒ omawia działanie i znaczenie poszczególnych linii programu testowego Hello World  ‒ sprawnie posługuje się środowiskiem Eclipse w zakresie edycji, uruchamiania i zachowywania programów w C++ | ‒ samodzielnie instaluje i uruchamia środowisko programistyczne Eclipse dla C++  ‒ samodzielnie uruchamia program testowy i umie go modyfikować | ‒ umie instalować inne środowiska programistyczne dla C++ oraz edytować i uruchamiać w nich programy |
| C z plusem, czyli podstawy programowania | ‒ wymienia i charakteryzuje kolejne etapy tworzenia programu komputerowego  ‒ wie, czym jest badanie warunku w programie i kiedy się je stosuje w kontekście bloków warunkowych algorytmu  ‒ wie, że istnieją różne typy operatorów i na podstawie podręcznika omawia rolę niektórych z nich  ‒ wie, że w programie mogą występować biblioteki i funkcje | ‒ korzystając z podręcznika, omawia podstawową strukturę programu w języku C++  ‒ zna znaczenie nawiasów klamrowych i "//" oraz ich rolę w programie C++  ‒ na podstawie tabeli z podręcznika omawia znaczenie operatorów  ‒ odróżnia operatory arytmetyczne od relacyjnych i logicznych i symboli porównawczych  ‒ wie, czym są zmienne i stałe w programie komputerowym i wskazuje ich deklarację w przykładowym programie  ‒ zna znaczenie i rolę funkcji i bibliotek  ‒ zna pojęcie pętli i warunku | ‒ samodzielnie omawia działanie poszczególnych operatorów wszystkich typów  ‒ umie zapisać warunki dla instrukcji warunkowej  ‒ umie dołączyć bibliotekę do kodu programu  ‒ wymienia różnice pomiędzy instrukcją warunkową a pętlą | ‒ samodzielnie układa proste programy ze strumieniowaniem  ‒ samodzielnie charakteryzuje najczęściej używane typy zmiennych liczbowych i logicznych  ‒ samodzielnie zapisuje fragment programu z instrukcją warunkową na podstawie algorytmu  ‒ wskazuje, w których przypadkach należy użyć danej instrukcji warunkowej lub pętli | ‒ samodzielnie stosuje instrukcje warunkowe i pętle w prostych programach układanych na podstawie algorytmów |
| C z plusem, czyli podstaw programowania ciąg dalszy | ‒ omawia na zasadzie analogii budowę zmiennych tablicowych 1 lub 2-wymiarowych  ‒ umie graficznie przedstawić tablice | ‒ omawia różnice pomiędzy zmiennymi a zmiennymi tablicowymi  ‒ podaje przykłady zastosowania indeksowania pozycji, np. w sali kinowej  ‒wie, że do wyprowadzania i wprowadzania informacji w programie komputerowym stosuje się strumieniowanie | ‒ w programie komputerowym w C++ umie zastosować strumieniowanie danych do wprowadzenia wartości danych lub wyprowadzenia na ekran tekstów lub wartości zmiennych  ‒ korzystając z podręcznika, umie analizować programy, w których zastosowano strumieniowanie danych  ‒ z pomocą podręcznika stosuje w prostych programach pętle i instrukcje warunkowe, np. do wypełniania tablicy danymi | ‒ samodzielnie układa proste programy, w których używa strumieniowania danych i instrukcji warunkowych oraz pętli  ‒ samodzielnie opracowuje sposób wyświetlania i wprowadzania danych  ‒ umie stosować instrukcje zagnieżdżone | ‒ proponuje rozwiązania problemów informatycznych wskazanych przez nauczyciela zawierających instrukcje warunkowe, pętle i strumieniowanie |
| **III. Programowanie i algorytmy** | | | | | |
| Wspólna miara, czyli podstawy złożoności obliczeniowej | - wie na czym polega złożoność obliczeniowa  - wie czym jest notacja omikron  - wie czym jest operacja dominująca | - umie objaśnić czym jest złożoność obliczeniowa  - objaśnia operację dominującą  - objaśnia złożoność pamięciową  - omawia czym jest złożoność optymistyczna | - omawia czym jest złożoność optymistyczna  - omawia notację omikron | ‒ samodzielnie stosuje w praktyce czasową złożoność obliczeniową  ‒ samodzielnie tworzy algorytm wykorzystujący znak klasy złożoności | ‒ samodzielnie układa algorytm dla bardziej złożonych problemów wykorzystujący czasową złożoność obliczeniową |
| Przez 1 i siebie, czyli jak zbadać, czy liczba jest liczbą pierwszą | ‒ zna definicję liczby pierwszej i umie wymienić kilka z nich, wskazując spełnienie podstawowej cechy  ‒ wymienia nazwę metod badania, czy liczba jest liczbą pierwszą | ‒ na podstawie tabeli z podręcznika umie objaśnić metodę sita Erastotenesa  ‒ objaśnia algorytmy badania, czy liczba jest liczbą pierwszą  ‒ zna zastosowanie liczb pierwszych | ‒ zna algorytmy zapisane w różnych postaciach wykrywające liczby pierwsze  ‒ sprawdza na przykładach działanie algorytmów wykrywających liczby pierwsze | ‒ samodzielnie tworzy i omawia działanie algorytmów wykrywających liczby pierwsze  ‒ przedstawia algorytmy w różnych zapisach w tym schematu blokowego i listy kroków  ‒ samodzielnie testuje algorytmy i dobiera odpowiednie dane wejściowe | ‒ samodzielnie układa algorytm testowania liczb na podstawie opisu metody |
| Która pierwsza, czyli jak program bada liczby | ‒ rozumie działanie algorytmów przedstawionych w podręczniku | ‒ na podstawie algorytmów z podręcznika omawia metodę badania liczby opartą na obliczaniu reszty z dzielenia  ‒ wskazuje fragmenty przykładowego programu zgodne z poszczególnymi fragmentami algorytmu | ‒ analizuje programy z podręcznika, omawiając znaczenie poszczególnych instrukcji  ‒ testuje programy i algorytmy na przykładowych danych  ‒ omawia modyfikacje algorytmu dla większej liczby danych | ‒ samodzielnie układa algorytmy i modyfikuje je wg wskazań nauczyciela  ‒ samodzielnie układa program na podstawie algorytmu  ‒ samodzielnie testuje program i poprawia ewentualne błędy, omawiając przyczynę ich powstania | ‒ samodzielnie układa program badający liczby wpisane do tablic o 1 lub więcej wymiarach |
| Więcej pierwszych, czyli generujemy liczby | - wie do czego w informatyce wykorzystuje się liczby pierwsze | - umie wygenerować liczbę pierwszą z niedużych liczb naturalnych za pomocą algorytmu | - analizuje algorytm, który wyłania liczby pierwsze z zadanego zakresu liczb | ‒ samodzielnie układa i modyfikuje je wg wskazań nauczyciela, który wyłania liczby pierwsze z zadanego zakresu liczb | - samodzielnie układa i modyfikuje program który wyłania liczby pierwsze z zadanego zakresu liczb |
| Największy i najmniejszy, czyli jak znaleźć NWD i NWW | ‒ definiuje NWD i omawia jego zastosowanie w matematyce  ‒ podaje kilka przykładów NWD dla wybranych liczb | ‒ na podstawie gotowego zapisu przykładu algorytmu Euklidesa, np. z podręcznika, omawia istotę tej metody  ‒ podaje i uzasadnia dziedzinę liczb, dla których przeznaczony jest algorytm Euklidesa | ‒ omawia różnicę pomiędzy metodą rekurencyjną a iteracyjną  ‒ analizuje gotowy przykład zastosowania metod Euklidesa  ‒ przedstawia algorytmy Euklidesa, np. w formie schematu blokowego, i tłumaczy ich istotę | ‒ analizuje obie metody Euklidesa pod kątem wydajności i szybkości działania dla różnych zestawów zmiennych wejściowych | ‒ samodzielnie przeprowadza analizę wydajności algorytmu Euklidesa dla różnych danych i przewiduje wyniki swojej analizy |
| Euklides i rekurencja, czyli szukamy NWD | - wie czym jest największy wspólny dzielnik (NWD)  - wie czym jest rekurencja | - umie wyszukać NWD  - układa algorytm umożliwiający wyszukanie NWD  - umie zastosować w praktyce rekurencję | - omawia ciąg Fibonacciego  - układa algorytmy i tłumaczy ich działanie z zastosowaniem ciągu Fibonnaciego | - omawia algorytm Euklidesa  - umie zastosować w praktyce algorytm Euklidesa układając algorytm rekurencyjny NWD | ‒ samodzielnie układa algorytm i pisze program na jego podstawie dla bardziej złożonych problemów wykorzystujący algorytm Euklidesa |
| Metoda iteracji, czyli algorytm Euklidesa w innej postaci | - wie czym jest iteracja | - omawia algorytm iteracyjny | - umie zastosować pętle w praktyce  - zna instrukcje pętli w językach C++ i Java | - umie zastosować iteracje w algorytmie Eukidesa | ‒ samodzielnie układa algorytm i pisze program na jego podstawie dla bardziej złożonych problemów wykorzystujący iterację w algorytmie Euklidesa |
| Dodawanie ułamków, czyli jak wykorzystać NWW i NWD w programie komputerowym | ‒ zapisuje algorytm dodawania ułamków w postaci listy kroków lub schematu blokowego | ‒ analizuje program ułożony na podstawie algorytmu dodawania ułamków | ‒ analizuje przykładowy program z podręcznika z użyciem funkcji i omawia ich rolę | ‒ samodzielnie układa program dodający ułamki i organizuje sposób wprowadzania danych i wyświetlania wyników | ‒ biegle posługuje się samodzielnie ułożonymi funkcjami ‒ obliczającymi wyniki pośrednie, a także organizującymi wydruk i wprowadzanie danych |
| Skracamy i wyłączamy, czyli dodawania ułamków ciąg dalszy | ‒ zapisuje algorytm upraszczania ułamków na podstawie ogólnej wiedzy matematycznej | ‒ określa rolę NWW i NWD w algorytmie upraszczania ułamków  ‒ omawia algorytm wyłączania całości z ułamka | ‒ analizuje program upraszczający ułamki i wskazuje w nim rolę funkcji obliczających NWW i NWD  ‒ proponuje różne rozwiązania problemu wyświetlania wyników  ‒ analizuje program wyłączania całości z ułamka | ‒ samodzielnie układa program wykorzystujący funkcje NWW i NWD do upraszczania ułamków  ‒ samodzielnie układa program wyłączający całości z ułamka | ‒ samodzielnie układa program dodający ułamki |
| Szyfrowanie, czyli poznajemy szyfr Cezara i szyfr przedstawieniowy | ‒ omawia cele szyfrowania danych i informacji  ‒ tłumaczy, na czym polega podstawieniowy sposób szyfrowania informacji  ‒ wie, jak odróżnić strony internetowe z szyfrowaną transmisją danych od pozostałych | ‒ na przykładzie tabeli tłumaczy metodę przestawieniową i umie zaszyfrować tekst tą metodą  ‒ omawia na podstawie rysunku z podręcznika metodę szyfrowania szyfrem Cezara | ‒ wie, na czym polega szyfrowanie szyfrem wieloalfabetowym  ‒ tłumaczy potrzebę szyfrowania niektórych transmisji w sieci | ‒ samodzielnie układa algorytm dla szyfru Cezara | ‒ samodzielnie układa program komputerowy szyfrujący szyfrem Cezara |
| Konspiracja, czyli tworzymy własne szyfry | ‒ wie, jaką rolę odgrywa klucz w metodzie szyfrowania | ‒ odróżnia metodę szyfrowania symetryczną od niesymetrycznej  ‒ na przykładzie przedstawia działanie klucza  ‒ wie, jak generować liczbę pseudolosową | ‒ prawidłowo, na podstawie podręcznika, analizuje działanie programu szyfrującego stałym kluczem  ‒ tłumaczy na przykładzie z podręcznika rolę tablicy stałej kluczy  ‒ objaśnia metodę stosowania losowo generowanych kluczy raz ich wady i zalety | ‒ samodzielnie układa programy szyfrujące z kluczem zapisanym w tablicy  ‒ samodzielnie układa program oparty na losowo generowanych danych w tablicach kluczy | ‒ samodzielnie modyfikuje program szyfrujący losowo generowanym kluczem w taki sposób, by szyfrował także klucze przesyłane odbiorcy informacji |
| Z kluczem publicznym, czyli bezpieczniejsze szyfrowanie | - wie czym jest szyfrowanie kluczem publicznym | - omawia czym jest szyfr asymetryczny (RSA)  - wie, ze szyfrowanie asymetryczne działa w trzech etapach | - zna etapy działania algorytmu RSA | - umie stworzyć algorytm szyfr wykorzystujący algorytm RSA | - samodzielnie pisze program na podstawie algorytmu szyfru wykorzystującego algorytm RSA  - samodzielnie pisze program deszyfrujący wykorzystujący algorytm RSA |
| Szyfrowanie zachłanne, czyli Kody Huffmana | - wie czym jest metoda zachłanna i w jakim celu ją się stosuje | - omawia zastosowanie metody zachłannej w szyfrowaniu  - omawia metodę optymalną | - stosuje metodę optymalną w tworzeniu algorytmów  - omawia zastosowanie Kodów Huffmana | - tworzy algorytmy wykorzystujące Kody Huffmana | - stosuje Kody Huffmana podczas kodowania znaków ASCII |
| Tworzenie szyfrów przedstawieniowych, czyli kodowanie znaków o zmiennej długości | - wie czym jest szyfr przestawieniowy | - wie jak wykorzystać szyfr przestawieniowy do kodowania znaków o zmiennej długości  - wie czym jest szyfr przestawieniowy z tablicą  - wie czym są kody bezprzystankowe | - umie kodować znaki o zmiennej długości za pomocą szyfru przestawieniowego  - umie stworzyć algorytm szyfrujący znaki o zmiennej długości wykorzystując szyfr przestawieniowy | - umie kodować znaki o zmiennej długości z wykorzystaniem szyfru przestawieniowego z tablicą  - umie stworzyć algorytm z wykorzystaniem kodu bezprzystankowego | - umie stworzyć program na podstawie algorytmu kodującego znaki o zmiennej długości wykorzystując szyfr przestawieniowy z tablica |
| Automat wydaje monety, czyli jak dynamicznie wydawać resztę | ‒ wie, na czym polega metoda zachłanna wydawania reszty  ‒ tłumaczy działanie algorytmu z podręcznika | ‒ samodzielnie układa algorytm wydawania reszty metodą zachłanną dla nominałów większych lub równych 1 zł  ‒ testuje programy i algorytmy z podręcznika na przykładowych zmiennych | ‒ samodzielnie układa algorytm wydawania reszty dla wszystkich możliwych nominałów  ‒ objaśnia działanie programu wydającego resztę w kontekście algorytmu podanego w podręczniku | ‒ samodzielnie układa algorytm i program wydający resztę metodą zachłanną korzystający ze wszystkich dostępnych nominałów  ‒ omawia problemy, które należało rozwiązać podczas układania algorytmu i programu dla dowolnych nominałów | ‒ modyfikuje program dla innego układu monetarnego, np. z USA |
| Wydajemy resztę, czyli jak wykorzystać metodę zachłanną | - wie czym jest metoda zachłanna | - objaśnia zasadę wydawania reszty przez automat z wykorzystaniem różnych nominałów | - układa algorytm zachłanny do stworzenia algorytmu rozwiązującego problem wydawania reszty w różnych nominałach przez automat | - pisze program z wykorzystaniem metody zachłannej | - układa program wykorzystujący algorytm zachłanny do stworzenia programu rozwiązującego problem wydawania reszty w różnych nominałach przez automat |
| Co z tym groszem, czyli błąd zaokrąglenia i błąd przybliżenia | - wie w jak zaokrągla się liczby  - wie w jakim celu zaokrągla się liczby | - omawia błąd zaokrąglenia  - omawia błędy względne i bezwzględne w zaokrągleniu | - wie dlaczego stosujemy zaokrąglanie liczb z nadmiarem i niedomiarem | - umie wykorzystać zaokrąglanie liczb z nadmiarem i niedomiarem  - umie wykorzystać problemy błędu zaokrąglania, błędu bezwzględnego i błędu względnego do rozwiązania zadań | - układa samodzielnie i modyfikuje programy, które rozwiązują problem zaokrąglania oraz błędy występujące przy zaokrąglaniu |
| Sprawdzam, czyli porównywanie tekstów i wyszukiwanie wzorca w tekście metodą naiwną | - wie czym jest wzorzec  - wie czym jest implementacja podczas szukania wzorca | - umie zastosować wzorzec do rozwiązania zadanego problemu  - omawia czym jest metoda naiwna w algorytmice | - umie stworzyć algorytm naiwny w celu wyszukiwania wzorca w łańcuchu tekstowym | - stosuje implementację podczas wyszukiwania wzorca w tekście  - umie stworzyć algorytm wykorzystując implementacje podczas wyszukiwania wzorca | - umie stworzyć program i go zmodyfikować wykorzystujący implementację podczas wyszukiwania wzorca |
| Od ostatniego, czyli szukamy wzorca tekstu metodą Boyera-Moore’a | - zna algorytm Boyera-Moore’a  - wie czym jest czym jest podliniowa klasa złożoności | - umie zastosować wyszukiwanie wzorca algorytmem BM | - umie wykorzystać wzorzec BM do wyszukiwania w tekście z użyciem tablicy | - umie wykorzystać uproszczony algorytm BM do wyszukiwania wzorca | - umie stworzyć program i go zmodyfikować wykorzystujący algorytm BM do wyszukiwania wzorca |
| Z lewej i z prawej, czyli szukamy palindromów | - wie czym jest palindrom  - wie co to jest algorytm naiwny | - omawia zastosowanie palindromu w informatyce | - umie stworzyć algorytm naiwny wyszukujący palindromy | - umie zastosować algorytm naiwny do sprawdzenia, czy dany ciąg znaków jest palindromem | - umie stworzyć program i go zmodyfikować wykorzystujący algorytm naiwny do sprawdzenia, czy dany ciąg znaków jest palindromem |
| Przestawianie liter, czyli jak stworzyć anagram | - wie czym jest anagram | - wie jakie zastosowanie ma anagram w informatyce | - umie stworzyć algorytm sortowania bąbelkowego | - umie stworzyć algorytm sprawdzający czy tablica jest anagramem z wykorzystaniem sortowania bąbelkowego | - umie stworzyć program i go zmodyfikować wykorzystujący algorytm sprawdzający czy tablica jest anagramem z wykorzystaniem sortowania bąbelkowego |
| Od A do Z, czyli alfabetyczne sortowanie wyrazów | - wie czym jest sortowanie alfabetyczne  - wie czym jest sortowanie jednopoziomowe | - umie zastosować sortowanie alfabetyczne w celu posortowania elementów ciągu znaków  - wie czym jest sortowanie przez wybór | - umie zastosować sortowanie jednopoziomowe w celu posortowania ciągu znaków | - umie stworzyć algorytm wykorzystujący sortowanie przez wybór | - umie stworzyć program i go zmodyfikować wykorzystujący algorytm wykorzystujący sortowanie przez wybór |
| Odwrotna notacja polska, czyli zapisujemy wyrażenia arytmetyczne | - wie czym jest odwrotna notacja polska | - umie zastosować odwrotną notację polską w działaniach arytmetycznych  - wie czym jest zapis prefiksowych | - umie zastosować zapis prefiksowy | - umie stworzyć algorytm obliczenia wartości wyrażenia odwrotnej notacji polskiej | - umie stworzyć program i go zmodyfikować wykorzystujący algorytm obliczenia wartości wyrażenia odwrotnej notacji polskiej |
| Naiwny jak algorytm, czyli szukamy najmniejszego i największego elementu metodą naiwną | - wie czym jest sortowanie naiwne  - wie czym jest rekurencja  - wie czym jest iteracja | - umie stworzyć algorytm wykorzystujący sortowanie naiwne | - umie stworzyć algorytm sortowania naiwnego w wersji rekurencyjnej | - umie stworzyć algorytm sortowania naiwnego w wersji iteracyjnej | - umie stworzyć program i go zmodyfikować wykorzystujący algorytm sortowania naiwnego w wersji rekurencyjnej  - umie stworzyć program i go zmodyfikować wykorzystujący algorytm sortowania naiwnego w wersji iteracyjnej |
| Dwie dodatkowe tablice, czyli stosujemy algorytm optymalny | - wie czym jest metoda „dziel i rządź”  - wie czym jest algorytm optymalny | - umie stworzyć algorytm wykorzystujący metodę „dziel i rządź” | - umie stworzyć algorytm optymalny | - umie stworzyć algorytm jednoczesnego wyszukiwania najmniejszego i największego elementu wykorzystując metodę optymalną | - umie stworzyć program i go zmodyfikować wykorzystujący algorytm jednoczesnego wyszukiwania najmniejszego i największego elementu wykorzystując metodę optymalną |
| Dziel i zwyciężaj, czyli sortowanie metodą scalania | - wie co to są algorytmy proste  - wie co to jest sortowanie metodą scalania | - umie tworzyć i stosować algorytmy proste  - umie tworzyć algorytmy wykorzystujące sortowanie metodą scalania | - umie tworzyć i stosować algorytmy wykorzystujące sortowanie rekurencyjne | - wie co to jest scalanie zbiorów uporządkowanych  - umie stworzyć algorytm scalania dwóch zbiorów uporządkowanych | - umie stworzyć program i go zmodyfikować wykorzystujący algorytm scalania dwóch zbiorów uporządkowanych |
| Raz krócej, raz dłużej, czyli sortowanie kubełkowe | - wie czym jest sortowanie kubełkowe | - zna cechy algorytmu sortowania | - umie opracować algorytm sortowania kubełkowego | - umie opracować algorytm sortowania metodą scalania | - umie stworzyć program i go zmodyfikować wykorzystujący algorytm sortowania kubełkowego i sortowania metodą scalania |
| **IV. Grafika i druk** | | | | | |
| Modele w przestrzeni, czyli podstawy druku 3D | ‒ omawia zasadę powstawania „wydruku” 3D  ‒ wie, w jakich płaszczyznach powinny poruszać się mechanizmy w drukarce 3D  ‒ wie, czym jest filament | ‒ na podstawie rysunku z podręcznika omawia budowę drukarki 3D  ‒ wie, jak znaleźć edytory obiektów 3D i przykładowe, darmowe modele  ‒ wymienia nazwy podstawowych formatów plików z modelami 3D dla drukarek  ‒ umie zastosować program komputerowy do wyświetlenia obiektu 3D np. Paint3D | ‒ objaśnia zasadę powstawania druku 3D na podstawie ilustracji przedstawiających budowę drukarki  ‒ wymienia kilka edytorów obiektów 3D | ‒ rozróżnia cechy podstawowe cechy formatów graficznych dla modeli 3D  ‒ umie ocenić jakość drukarki 3D na podstawie specyfikacji technicznej  ‒ umie wstępnie dobrać rodzaj filamentu do konkretnego zadania | ‒ samodzielnie potrafi tworzyć obiekty 3D dla drukarek, korzystając z darmowych edytorów |
| Modelujemy, czyli jak projektować obiekty 3D | ‒ wymienia cechy edytorów 3D  ‒ wie, jak szukać edytorów w chmurze | ‒ korzysta z edytora 3D w chmurze, np. Tinkercad, w celu przeglądania gotowych projektów  ‒ wie, jak sprawdzić licencję danego projektu | ‒ modyfikuje modele w edytorze na podstawie opisu z podręcznika  ‒ tworzy prosty obiekt 3D na podstawie opisu z podręcznika | ‒ samodzielnie i według własnego pomysłu modyfikuje obiekt 3D z chmury  ‒ samodzielnie tworzy własny obiekt 3D dla drukarki, np. litery powiązane łącznikami | ‒ samodzielnie projektuje i wykonuje obiekty 3D przeznaczone dla drukarki 3D |
| Wizualizacja pomysłów, czyli projektujemy dom w edytorze 3D | ‒ umie przeglądać modele w chmurze SketchUp  ‒ kreśli podstawowe bryły w SketchUp | ‒ posługuje się chmurą SketchUp i mapą Google w celu zlokalizowania i przeglądania modeli 3D obiektów architektonicznych w swojej okolicy | ‒ tworzy proste projekty obiektów w edytorze SketchUp  ‒ wypełnia modele kolorem, deseniem lub grafiką z pliku | ‒ samodzielnie tworzy obiekty 3D na podstawie zdjęć lub obserwacji obiektów architektonicznych z okolicy swojej szkoły | ‒ używa zaawansowanych narzędzi projektowania 3D do edycji obiektów architektonicznych |
| Urządzamy mieszkanie, czyli wizualizacje 3D | - wie co to grafika 3G | - umie zaprojektować grafikę 2D z wykorzystaniem odpowiedniego oprogramowania | - umie zaprojektować grafikę 3D z wykorzystaniem odpowiedniego oprogramowania | - korzysta z oprogramowania 3D umożliwiającego tworzenie wizualizacji | - tworzy zaawansowane projekty w grafice 3D |
| Wypukłości nie tylko 3D, czyli tworzymy elementy graficzne publikacji | ‒ wie, na czym polega stosowanie warstw i co można dzięki nim osiągnąć  ‒ wymienia kilka nazw edytorów grafiki oferujących mechanizm warstw | ‒ zna przeznaczenie podstawowych narzędzi edycyjnych  ‒ posługuje się podstawowymi narzędziami edycyjnymi | ‒ na podstawie opisu z podręcznika umie utworzyć trójwymiarowy napis i wyeksportować go do pliku PNG | ‒ samodzielnie tworzy trójwymiarową grafikę z wykorzystaniem warstw i mechanizmów opisanych w podręczniku | ‒ biegle posługuje się edytorem grafiki rastrowej i tworzy grafikę wg własnego projektu |
| Dynamicznie, czyli jak tworzyć ruchome wizualizacje | - wie czym jest wizualizacja danych  - wie czym jest rysunek techniczny  - wie czym jest animacja komputerowa | - umie zastosować w praktyce animację danych w celu zaprezentowania wyników | - umie stworzyć prosty rysunek techniczny z wykorzystaniem programów CAD | - umie stworzyć prostą animację komputerową  - zna 4 etapy tworzenia wizualizacji | - samodzielnie tworzy zaawansowane projekty wykorzystujące oprogramowanie CAD  - samodzielnie tworzy zaawansowane animacje komputerowe |
| Z wydruku do komputera, czyli skanujemy i odczytujemy dokumenty | ‒ wie, co oznacza skrót OCR, i do czego służy program zaliczany do klasy programów OCR  ‒ wymienia niektóre przypadki, w których stosuje się OCR  ‒ wie, do czego służy skaner | ‒ obsługuje skaner  ‒ zna zasadę działania skanera i umie dobrać rodzaj skanera do określonego zadania  ‒ umie posłużyć się panelem obsługi skanera | ‒ zna pojęcie TWAIN i wie, gdzie stosuje się ten standard komunikacji  ‒ umie świadomie ustawić podstawowe parametry skanowania dokumentu tekstowego przeznaczonego do rozpoznania tekstu  ‒ uzasadnia dobór parametrów skanowania  ‒ na podstawie opisu używa programu OCR z chmury lub aplikacji | ‒ samodzielnie używa programu OCR i skanera do rozpoznawania pisma  ‒ opisuje różnice pomiędzy skanerami CIS a CCD | ‒ samodzielnie i sprawnie dobiera parametry programu OCR do rozpoznawania tabel i grafiki zawierającej litery, omawia cechy programu, które na to pozwalają |
| Dokumentujemy wydarzenia, czyli aktywna praca z aparatem fotograficznym | ‒ umie wykonywać zdjęcia aparatem fotograficznym lub telefonem komórkowym, korzystając z programu automatycznego | ‒ wie, jaka jest zależność pomiędzy czasem naświetlania a efektem na fotografii poruszającego się obiektu  ‒ wie, na co ma wpływ przysłona | ‒ zna pojęcie głębi ostrości i od czego ona zależy  ‒ zna i stosuje zasady kompozycji obrazu, uzasadniając sposób kadrowania  ‒ uzasadnia stosowanie kompozycji kadru fotograficznego w tworzeniu grafiki  ‒ sprawnie wykorzystuje programy tematyczne w aparacie | ‒ samodzielnie operuje ręcznymi nastawami aparatu fotograficznego w celu uzyskania odpowiedniego efektu na zdjęciu, np. głębi ostrości, rozmycia tła itp. | ‒ samodzielnie tworzy fotoreportaże i inne projekty fotograficzne |
| Szturmowiec w chmurze, czyli poprawiamy zdjęcia w edytorze grafiki rastrowej | ‒ umie wskazać zastosowanie warstw w procesie edycji zdjęcia  ‒ sprawnie loguje się do chmury z edytorem grafiki, np. pixlr.com | ‒ zna przeznaczenie podstawowych narzędzi i opcji edytorów grafiki rastrowej w tym pixlr.com i GIMP  ‒ z pomocą podręcznika posługuje się podstawowymi narzędziami edytora  ‒ umie poprawić kadrowanie zdjęcia przy pomocy edytora | ‒ sprawnie posługuje się edytorem w chmurze  ‒ sprawnie posługuje się podstawowymi narzędziami edycyjnymi, w tym stemplem  ‒ reguluje poziom jasności i kontrastu za pomocą narzędzi edytora  ‒ korzysta z automatycznych narzędzi poprawiających zdjęcia | ‒ precyzyjnie posługuje się narzędziami edycyjnymi  ‒ skutecznie dokonuje retuszu zdjęcia  ‒ świadomie i z rozwagą dobiera automatyczne narzędzia do korekty zdjęć | ‒ sprawnie operuje ustawieniami parametrów poszczególnych narzędzi, osiągając bardzo dobre efekty ich zastosowania |
| Tego tu nie było, czyli poprawiamy rzeczywistość na fotografiach | ‒ opisuje przykłady zastosowania warstw w edycji grafiki  ‒ wie, których narzędzi należy użyć do montażu zdjęć | ‒ omawia proces wklejania fragmentu zdjęcia za pośrednictwem warstw  ‒ stosuje warstwy do montażu zdjęcia  ‒ edycję przeprowadza z nienależytą starannością  ‒ wie, jak posługiwać się warstwami podczas montażu zdjęcia | ‒ dokonuje montażu zdjęcia lub grafiki na podstawie opisu z podręcznika  ‒ sprawnie operuje warstwami i narzędziami podczas montażu zdjęcia | ‒ prace nad korekta i montażem zdjęcia wykonuje starannie i precyzyjnie  ‒ precyzyjnie posługuje się narzędziami do zaznaczania fragmentów zdjęcia | ‒ samodzielnie i precyzyjnie odrestaurowuje zniszczone fotografie |
| Własny film, czyli jak twórczo wykorzystać kamerę | ‒ wymienia i rozpoznaje podstawowe plany filmowe  ‒ wie, czym różni się scenariusz od opowiadania lub powieści | ‒ zna pojęcie osi filmowej i uzasadnia jej stosowanie  ‒ na podstawie podręcznika tworzy krótki film z fotografii  ‒ wie, jaka jest różnica między plikiem projektu a plikiem wideo, np. mp4  ‒ wymienia nazwy popularnych edytorów wideo, odróżniając narzędzia darmowe od komercyjnych | ‒ umie zaplanować ustawienie kamer do rejestracji dialogu  ‒ dodaje do filmu dźwięk w prostym edytorze wideo  ‒ eksportuje film do pliku | ‒ samodzielnie tworzy tutorial na zadany temat zgodny z ułożonym przez siebie scenariuszem  ‒ samodzielnie dobiera materiały do filmu  ‒ prawidłowo dobiera formaty zapisu pliku wideo | ‒ realizuje własne projekty filmowe i umieszcza je w sieci |
| Ze stratą lub bez, czyli jak kompresować pliki | - wie co to jest kompresja  - wie czym jest kompresja stratna | - umie wymienić metody kompresji  - omawia kompresję obrazu  - omawia kompresję dźwięku | - omawia kodowanie arytmetyczne | - omawia kodowanie Huffmana | - omawia kodowanie LZW  - omawia kodowanie RLE |
| Klatki do komputera, czyli jak transmitować strumieniowo | ‒ zna funkcje podstawowych elementów kamery opisane w podręczniku  ‒ umie zainstalować program VNC i odtwarzać za jego pomocą pliki wideo | ‒ umie określić funkcję najważniejszych elementów kamery lub aparatu fotograficznego  ‒ opisuje sposoby kopiowania plików wideo z kamery lub aparatu do komputera | ‒ na podstawie opisu z podręcznika umie skonfigurować program do zapisu strumienia pochodzącego z kamery  ‒ korzystając z opisu i programu VLC, umie nałożyć na transmitowanym obrazie symbol graficzny | ‒ samodzielnie konfiguruje program komputerowy do zapisu strumienia wideo i audio na dysku komputera  ‒ samodzielnie nakłada symbole graficzne na obraz podczas transmisji strumieniowej  ‒ samodzielnie transmituje obraz i dźwięk za pośrednictwem telefonu komórkowego  ‒ przeprowadza transmisję strumieniową w sieci lokalnej | ‒ planuje i przeprowadza transmisje na żywo, np. z uroczystości szkolnych |
| **V. Komputer w sieci** | | | | | |
| Nie wszystko jest takie oczywiste, czyli jak działa internet | ‒ wie, jaką rolę odgrywają protokoły w sieciach komputerowych  ‒ opisuje rolę adresów w sieciach lokalnych i internecie | ‒ na podstawie podręcznika umie wyświetlić parametry połączenia sieciowego za pomocą polecenia tracert uruchomionego w Wierszu poleceń  ‒ na podstawie podręcznika uruchamia i stosuje program do śledzenia połączeń z serwerem wybranej strony WWW | ‒ wyjaśnia różnice pomiędzy adresem IP a adresem symbolicznym  ‒ na podstawie opisu z podręcznika sprawnie posługuje się programami do śledzenia połączeń w sieci  ‒ tłumaczy rolę DNS-ów w globalnej sieci  ‒ tłumaczy rolę adresów IP  ‒ wie, jakie instytucje są odpowiedzialne za przydzielanie adresów IP w sieci globalnej  ‒ wie, czym jest domena | ‒ samodzielnie korzysta z programów do śledzenia połączeń i znajdowania właściciela domen  ‒ tłumaczy zadania protokołu DHCP  ‒ interpretuje wyniki działania programów śledzących połączenia oraz polecenia systemowego tracert | ‒ biegle posługuje się różnymi programami do diagnozowania i testowania działania sieci komputerowych |
| Zabezpieczenia firewall | - wie czym jest firewall | - wie jak działa zapora sieciowa  - omawia typy zapór sieciowych | - omawia serwery NAT  - zna wady zapór sieciowych | - wyjaśnia, na czym polega tworzenie przez zaporę sieciową strefy ograniczonego zaufania nazywanej  również strefą zdemilitaryzowaną DMZ. | - samodzielnie konfiguruje zapory sieciowe |
| Sieć to nie tylko internet, czyli poznajemy topologię sieci komputerowych | ‒ wie, czym różni się sieć P2P od sieci w innych konfiguracjach  ‒ określa rolę przełącznika sieciowego | ‒ na podstawie podręcznika umie połączyć dwa urządzenia, np. telefony, w sieć P2P  ‒ na podstawie podręcznika omawia różnice pomiędzy różnymi topologiami sieci | ‒ omawia wady i zalety poszczególnych topologii sieci  ‒ umie przedstawić graficznie różne topologie sieci komputerowych  ‒ na podstawie podręcznika i algorytmu omawia proces zestawienia połączenia P2P urządzeń z interfejsem bluetooth | ‒ samodzielnie dokonuje połączeń komputerów z siecią w różnych topologiach  ‒ samodzielnie łączy urządzenia przez bluetooth i przesyła pliki, np. zdjęcia | ‒ projektuje proste sieci z uwzględnieniem ich przeznaczenia |
| Warstwowy model sieci komputerowej | - wie czym są modele sieci komputerowej | - omawia w jakim celu zostały stworzone modele sieci komputerowych | - omawia model TCP/IP | - omawia model ISO/OSI i jego warstwy | - omawia etapy komunikacji opisane poprzez modele warstwowe sieci ‒ ISO/OSI i TCP/IP |
| Funkcje urządzeń i protokoły | - wie co to są urządzenia sieciowe | - omawia poszczególne urządzenia sieciowe  - zna protokoły modeli warstwowych sieci | - omawia zastosowanie poszczególnych urządzeń sieciowych | - omawia protokoły warstw aplikacji  - omawia protokoły warstwy transportowej  - omawia protokoły warstwy sieciowej  - omawia protokoły wirtualnych sieci prywatnych: PPP, PPTP i L2TP | - omawia trasę po której pakiety są przesyłane do konkretnego serwera www  - omawia za co odpowiadają protokoły ICMP, ARP |
| Kto tam, czyli identyfikujemy komputery w sieci | ‒ na podstawie podręcznika konfiguruje sieć w komputerze z systemem Windows, korzystając z automatycznych opcji konfiguracyjnych | ‒ na podstawie podręcznika umie sprawdzić poleceniem ping poprawność połączenia sieciowego | ‒ na podstawie podręcznika omawia informacje wyświetlone za pomocą poleceń ipconfig oraz ping  ‒ na podstawie podręcznika omawia rolę poszczególnych protokołów sieciowych | ‒ samodzielnie posługuje się poleceniami ipconfig oraz ping w celu sprawdzenia parametrów połączeń sieciowych  ‒ samodzielnie opisuje poszczególne dane odczytane za pomocą ipconfig oraz ping  ‒ samodzielnie korzysta z automatycznych narzędzi konfiguracji sieci  ‒ samodzielnie omawia rolę poszczególnych protokołów sieciowych | ‒ samodzielnie konfiguruje połączenie sieciowe z pominięciem nastaw automatycznych |
| Kupujemy świadomie, czyli poznajemy parametry urządzeń peryferyjnych | ‒ umie oszacować koszty wydruku dla danego typu lub modelu drukarki  ‒ rozpoznaje i nazywa wejścia sygnałowe w monitorach  ‒ wie, do czego służy skaner | ‒ wymienia parametry drukarek  ‒ na podstawie podręcznika określa wpływ poszczególnych parametrów drukarek na jakość druku  ‒ na podstawie podręcznika omawia parametry monitorów  ‒ na podstawie podręcznika omawia parametry skanerów | ‒ na podstawie podręcznika omawia cechy i parametry poszczególnych typów drukarek i ich wpływ na wybór dokonywany ze względu na zastosowanie  ‒ określa parametry monitorów oraz wpływ formatu obrazu na zastosowanie na różnych stanowiskach | ‒ samodzielnie analizuje parametry urządzeń peryferyjnych i ocenia ich przydatność do konkretnego zastosowania  ‒ samodzielnie wyjaśnia zalety i wady różnych rodzajów ekranów monitorów | ‒ potrafi na podstawie danych katalogowych trafnie dobrać urządzenie peryferyjne, biorąc pod uwagę wymagania użytkownika |
| Projektuję rozbudowę i zakup nowego zestawu | - wie czym jest komputer stacjonarny  - wie czym jest komputer przenośny | - zna różnice między komputerem stacjonarnym i przenośnym | - potrafi przeprowadzić analizę potrzeb przed zakupem nowego komputera | - samodzielnie wyjaśnia wady i zalety komputera przenośnego i komputera stacjonarnego, omawiając jego poszczególne komponenty | - potrafi na podstawie danych dostępnych w sieci www wybrać urządzenia, które polepszą funkcjonowanie komputera przenośnego lub stacjonarnego |
| **VI. Opracowania rozwiązań problemów wybranymi aplikacjami** | | | | | |
| Host i serwer, czyli o działaniu sieci | – na podstawie podręcz- nika i z pomocą nauczyciela omawia podstawowe pojęcia związane z budową sieci komputerowych (np. *host*, *serwer*, *klient*, *LAN*, *WAN*, *WiFi*);  –wie, czym są media transmisyjne;  – wymienia nazwy najczęściej spotykanych topologii sieci lokalnej i na podstawie ilustracji (np. z podręcznika) wymienia ich podstawowe cechy. | – samodzielnie omawia znaczenie podstawowych pojęć związanych z budową i działaniem sieci komputerowych;  – na podstawie podręcz- nika omawia cechy poszczególnych mediów transmisyjnych;  – samodzielnie wymienia nazwy najczęściej spotykanych topologii sieci lokalnej i wymienia ich podstawowe cechy;  – wymienia jednostki przesyłu informacji i omawia zależności pomiędzy nimi;  – wie, że do różnych zastosowań można używać różnych rodzajów kabla sieciowego. | – samodzielnie omawia znaczenie wszystkich pojęć związanych z budową i działaniem sieci komputerowych;  – oblicza maksymalną ilość danych przesyłanych w sieci na podstawie danych o czasie i przepustowości łącza;  – z niewielką pomocą nauczyciela wymienia nazwy i cechy podstawowych kabli dla sieci lokalnych CAT 3, 5 i 5e, 6 i 7;  – wymienia podstawowe dla transmisji danych cechy światłowodu. | – wybiera odpowiednią topologię i medium dla różnych sieci lokalnych;  – identyfikuje topologię istniejącej sieci (np. w pracowni);  – identyfikuje rodzaje mediów transmisyjnych;  – prawidłowo stosuje poznane pojęcia dotyczące sieci lokalnych. | – omawia niewymienione w podstawie programowej zagadnienia związane z tematem. |
| Włączamy do ruchu, czyli konfiguracja lokalnej sieci komputerowej | – omawia funkcję routera w sieci lokalnej;  – wie, że router należy odpowiednio skonfigurować;  – podaje, jak znaleźć opis konfiguracji routera (np. instrukcję, samouczek w sieci);  – identyfikuje nazwę i producenta routera. | – omawia funkcję DHCP w kontekście konfiguracji routera;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł wyjaśnia pojęcia: *DHCP*, *serwer*, *szyfrowanie* (np. WPA2-PSK) i ich znaczenie dla konfiguracji routera;  – na podstawie instrukcji i z pomocą nauczyciela omawia lub przeprowa- dza konfigurację przykładowego routera. | – na podstawie instrukcji omawia lub przeprowadza konfigurację przykładowego routera;  – na podstawie instrukcji lub opcji Pomocy omawia lub przeprowadza konfigurację przykładowego routera bez wykorzystania DHCP. | – samodzielnie omawia lub przeprowadza konfigurację przykładowego routera z wykorzystaniem DHCP i bez tego mechanizmu. | – dokonuje konfiguracji zaawansowanych funkcji routera (np. uprawnień poszczególnych użytkowników). |
| Nie tylko WiFi, czyli bezprzewodowy dostęp do internetu | – wie czym się różni WLAN od WiFi;  – omawia zastosowanie złączy Bluetooth do połączenia urządzeń komputerowych;  – z niewielką pomocą nauczyciela przeprowadza przesłanie pliku (np. z telefonu do laptopa) za pośrednictwem Bluetooth. | – wymienia nazwy i omawia przeznaczenie podstawowych mediów i urządzeń transmisyjnych w tym: WLAN, WiFi, Bluetooth;  – odróżnia i prawidłowo przypisuje do danych mediów ich symbole graficzne. | – wyjaśnia, na czym polega łączenie urządzeń za pomocą fal radiowych i podaje, jakie warunki muszą być przy tym spełnione;  – na podstawie opisów omawia różnice między bezprzewodowymi mediami transmisyjnymi;  – wyjaśnia znaczenie norm określających ramy działania poszczególnych mediów transmisyjnych. | – wymienia pasma radiowe przeznaczone dla mediów transmisyjnych stosowanych w sieciach bezprzewodowych;  – opisuje możliwości wykorzystania mediów bezprzewodowych i samodzielnie przepro- wadza konfigurację podłączanych do nich urządzeń (np. słuchawek, telefonu, laptopa). | – omawia zastosowania i działanie WiMAX;  – samodzielnie łączy klika urządzeń jednocześnie za pomocą Bluetooth. |
| Zespół, czyli realizujemy projekty w chmurze | – omawia budowę przykładowej chmury, wymieniając jej podstawowe składniki. | – przedstawia i omawia przykładowy schemat struktury chmury informatycznej;  – wymienia najczęściej spotykane w chmurach narzędzia (m.in. dysk, edytor, arkusz, kalendarz);  – wie, że chmurę można wykorzystać do pracy zespołowej. | – omawia przeznaczenie poszczególnych elementów chmury użytej na zajęciach;  – porównuje działanie programów z chmury z ich odpowiednikami z apli- kacji komputerowych;  – posługuje się chmurą w stopniu wystarczającym do wykonywania podstawowych czynności, takich jak edycja dokumentu, kopiowanie plików itp.;  – wie, które aplikacje można wykorzystać do organizacji pracy zespołu. | – biegle posługuje się programami i dyskami sieciowymi;  – samodzielnie przygotowuje chmurę do pracy zespołu, m.in. konfiguruje kalendarz, udostępnia foldery i pliki. | – samodzielnie planuje i organizuje pracę zespołu w chmurze;  – posługuje się kilkoma chmurami (np. Google i OneDrive). |
| Tutorial, czyli jak tworzyć pomoce i instrukcje obsługi | – wie czym są style i szablony;  – zmienia styl w trakcie edycji dokumentu. | – prawidłowo dobiera style do treści zawartej w dokumencie;  – wie, że można samodzielnie definiować style i szablony;  – uruchamia edycję, wykorzystując szablon oferowany przez edytor. | – dobiera styl i szablon zgodnie z planowaną zawartością i tematem dokumentu;  – na podstawie podręcz- nika definiuje szablon i styl dokumentu;  – odnajduje w sieci (np. w chmurze) szablony dla różnych dokumentów. | – samodzielnie definiuje szablon i styl dokumentu;  – projektuje szablon zgodnie z treścią przyszłego dokumentu;  – modyfikuje i projektuje nowe style. | – projektuje szablony dla różnych edytorów (np. LibreOffice Draw) i ich używa. |
| Rozbudowana struktura, czyli korzystamy z konspektu w edytorze tekstu | – rozumie i omawia pojęcie konspektu;  – podaje, czym jest akapit i jaką pełni funkcję w edycji i formatowaniu tekstu. | – na podstawie podręcz- nika tworzy konspekt dokumentu;  – na podstawie podręcz- nika dzieli dokument na sekcje i kolumny. | – samodzielnie tworzy konspekt dokumentu;  – samodzielnie dzieli dokument na sekcje i kolumny;  – tworzy spis treści na podstawie konspektu. | – uzasadnia stosowanie podziału dokumentu tekstowego na sekcje lub kolumny;  – stosuje różny podział na niektórych stronach (sekcjach) dokumentu. | – stosuje podziały i sekcje w różnych edytorach tekstu. |
| Broszura, czyli jak projektować duże dokumenty | – wstawia do dokumentów SmartArt i Kształty. | – uzasadnia wybór danego kształtu lub ilustracji SmartArt;  – na podstawie podręcz- nika tworzy spisy ilustracji i tabel. | – na podstawie podręcz- nika lub tutoriali zmienia domyślne opcje edytora;  – na podstawie podręcz- nika lub tutoriali wyko- nuje automatyczne spisy treści, tabel i ilustracji;  – na podstawie podręcz- nika lub tutoriali aktuali- zuje spisy po zmianach. | – samodzielnie zmienia domyślne opcje edytora;  – samodzielnie wykonuje automatyczne spisy treści, tabel i ilustracji;  – samodzielnie aktualizuje spisy po zmianach;  – decyduje, które z opcji domyślnych zmienić i uzasadnia swój wybór. | – spełnia kryteria oceny bardzo dobrej dla innych edytorów (np. LibreOffice). |
| Recenzja, czyli proponujemy poprawki w tekście | – odczytuje i interpretuje dokumenty, w których zastosowano tryb recenzji;  – odczytuje notatki i zaznaczenia w Adobe Acrobat Reader DC. | – na podstawie podręcz- nika używa opcji Recenzja w Word i wstawiania komentarzy w dokumencie PDF. | – na podstawie podręcz- nika lub tutoriali używa opcji Recenzja do wpisywania komentarzy w Word oraz opcji wstawiania komentarzy i zaznaczania fragmentów w dokumencie PDF;  – na podstawie podręcz- nika lub tutoriali uruchamia opcję śledzenia zmian. | – samodzielnie używa opcji Recenzja do wpisywania komentarzy w Word oraz opcji wstawiania komentarzy i zaznaczania fragmentów w dokumencie PDF;  – samodzielnie uruchamia opcję śledzenia zmian;  – świadomie używa opcji Recenzja do proponowa- nia zmian w dokumencie;  – analizuje wynik działa- nia opcji porównywania dokumentów. | – spełnia kryteria oceny bardzo dobrej dla innych edytorów (np. LibreOffice);  – przedstawia dokument z innego przedmiotu lub projektu, w którym użył opcji Recenzja dla dokumentu Word lub Notatka dla PDF. |
| OLE, czyli łączymy dokumenty w dokumentacji pracy zespołu | – na podstawie podręcz- nika wyjaśnia, czym jest i do czego służy technika OLE. | – wyjaśnia, jakie obiekty mogą być wstawiane za pomocą techniki OLE;  – na podstawie podręcz- nika lub opcji Pomoc umieszcza w dokumencie obiekty OLE;  – na podstawie podręcz- nika wyjaśnia, czym są OLE2 i ActiveX;  – wyjaśnia, dlaczego podczas używania ActiveX należy zachować szczególną ostrożność. | – na podstawie podręcz- nika i opcji Pomocy przetwarza obiekty OLE w edytorach tekstu itp. | – samodzielnie przetwarza obiekty OLE w edytorach tekstu itp. | – spełnia kryteria oceny bardzo dobrej dla innych edytorów (np. LibreOffice). |
| **VII. Rozwijanie kompetencji społecznych** | | | | | |
| Informatyka pokonuje schody, czyli nikt nie powinien być wykluczony | – wymienia cechy aplikacji ratujących życie lub ułatwiających funkcjonowanie osobom niepełnosprawnym. | – znajduje i instaluje wiarygodne aplikacje ratujące życie lub zdrowie;  – wymienia urządzenia peryferyjne do monitorowania parametrów organizmu;  – posługuje się translatorem języków. | – ocenia jakość aplikacji ratujące życie lub zdrowie;  – ocenia jakość aplikacji ułatwiających funkcjono- wanie osobom niepełno- sprawnym i starszym;  – wymienia cechy strony dostosowanej do potrzeb osób niepełnosprawnych;  – używa peryferiów monitorujących stan organizmu. | – dobiera peryferia i aplikacje pomagające w ratowaniu życia lub zdrowia;  – omawia cechy dobranych przez siebie aplikacji i porównuje ich jakość;  – pokazuje przykłady stron przystosowanych do potrzeb osób niepełnosprawnych. | – projektuje strony przystosowane do potrzeb osób niepełnosprawnych;  – korzysta z translatora w trybie offline. |
| Platformy uczą, czyli rola e-learningu w nauce i pracy | – wymienia podstawowe różnice między nauczaniem zdalnym a klasycznym. | – wymienia zalety i wady e-learningu;  – umie skorzystać z gotowych lekcji  e-learningowych. | – opisuje przykładową strukturę lekcji e-learningowej;  – opisuje przykładowy cykl nauczania e-learningowego. | – wymienia właściwości wybranego systemu do tworzenia i prowadzenia kursów e-learningowych;  – porównuje zalety i wady e-learningu z nauczaniem tradycyjnym;  – znajduje w sieci kursy e-learningowe na dany temat i odczytuje, na jakich warunkach można w nich uczestniczyć. | – układa scenariusz lekcji e-learningowej i wskazuje różnice w porównaniu z tradycyjną lekcją. |
| Możesz być administratorem, czyli jak zarządzać platformą e-learningową | – omawia warunki, jakie musi spełniać platforma e-learningowa. | – uruchamia aplikację Classroom;  – wymienia elementy, jakie powinny się znaleźć w strukturze kursu e-learningowego;  – na podstawie podręcz- nika tworzy lekcje w Classroom. | – projektuje przykładową strukturę kursu e-learningowego;  – sporadycznie korzystając z podręcznika, tworzy lekcję i dodaje temat zajęć w Classroom. | – projektuje strukturę kursu e-learningowego na zadany temat i tworzy ją w Classroom;  – testuje poprawność działania kursu. | – układa kurs e-learningowy w innej platformie niż Classroom. |
| Zasoby i testy, czyli wypełniamy kursy treścią | – wymienia rodzaje zasobów, które można umieszczać w kursach e-learningowych. | – objaśnia rolę poszczególnych rodzajów zasobów kursów e-learningowych;  – wymienia rodzaje pytań testowych, jakie mogą się znaleźć w testach online. | – dzieli zasoby kursów według różnych kryteriów;  – sporadycznie korzysta- jąc z podręcznika, umiesz- cza w Classroom gotowe zasoby edukacyjne i testy;  – zaprasza użytkowników do korzystania z kursu. | – samodzielnie tworzy część zasobów, które umieści w Classroom;  – tworzy krótkie testy w Classroom. | – umieszcza zasoby w innej platformie niż Classroom. |
| Dzielimy się wiedzą, czyli współtworzymy zasoby udostępniane na platformach | – uzasadnia potrzebę współdzielenia zasobów chmury lub platformy e-learningowej. | – z pomocą nauczyciela lub na podstawie podręcznika udostępnia zasoby umieszczone w chmurze lub platformie. | – samodzielnie udostępnia zasoby na platformie lub chmurze w wybrany przez siebie sposób. | – samodzielnie udostępnia zasoby, korzystając z różnych sposobów oferowanych przez oprogramowanie chmury lub platformy. | – wykonuje czynności opisane w podręczniku także w innych chmurach i platformach. |
| **VIII. Bazy danych** | | | | | |
| Jak gromadzić informacje, czyli komputerowe bazy danych | – podaje przykłady baz danych, z których korzysta np. w swoim telefonie;  – definiuje pojęcie bazy danych. | – wymienia wszystkie elementy bazy danych opisane w podręczniku. | – omawia przeznaczenie poszczególnych elementów bazy danych. | – samodzielnie opisuje czynności występujące w trakcie projektowania bazy danych. | – opisuje zależności między poszczególnymi elementami bazy danych. |
| Tabele i formularze, czyli jak utworzyć bazę danych | – wymienia przykładowe dane, jakie powinna zawierać baza uczestników projektu. | – uzasadnia wybór danych dla bazy uczestników projektu;  – na podstawie podręcz- nika tworzy tabele bazy danych zgodnie z danymi zdefiniowanymi dla uczestników projektu;  – rozumie znaczenie tabel w bazie danych. | – samodzielnie tworzy kolejne tabele projektu;  – korzystając z kreatora odnośników, tworzy je dla pól tabeli, wybierając je spośród wskazanych w podręczniku;  – umieszcza dane w tabelach;  – omawia znaczenie klucza;  – prawidłowo posługuje się pojęciami związanymi z bazami danych. | – tworzy klucze;  – tworzy odnośniki dla pól tabel;  – tworzy formularze. | – czynności opisane dla Access wykonuje także w innym systemie baz danych (np. LibreOffice Base). |
| Relacje i pytania, czyli jak uczynić bazę użyteczną | – wyjaśnia znaczenie relacji między tabelami bazy. | – wyjaśnia różnice między rodzajami relacji. | – tworzy klucz zewnętrzny;  – używa kreatora relacji między tabelami;  – wypełnia pola tabel powiązanych za pomocą kreatora;  – tworzy relacje za pomocą narzędzia Relacje na podstawie opisu w podręczniku. | – tworzy relacje za pośrednictwem kreatora i narzędzia Relacje;  – tłumaczy sens tworzenia relacji i efekt ich działania w przykładzie z ćwiczeń. | – czynności opisane dla Access wykonuje także w innym systemie baz danych (np. LibreOffice Base). |
| Kto pyta, nie błądzi, czyli jak korzystać z baz danych Access | – wyjaśnia znaczenie pojęcia *kwerenda*;  – wyjaśnia, na czym polega filtrowanie w informatyce. | – na podstawie podręcz- nika tworzy kwerendę, używając kreatora;  – na podstawie podręcz- nika lub tutoriali modyfikuje kwerendę, dodając filtrowanie;  – na podstawie podręcz- nika lub tutoriali sortuje dane i tworzy raport za pomocą kreatora. | – samodzielnie tworzy kwerendę, używając kreatora;  – samodzielnie modyfikuje kwerendę, dodając filtrowanie;  – samodzielnie sortuje dane i tworzy raport za pomocą kreatora. | – planuje kwerendę i ją opracowuje;  – prawidłowo decyduje o wyborze filtrowania, sposobie sortowania i raportowania;  – wykonuje eksport tabel do innych formatów (np. Excel). | – czynności opisane dla Access wykonuje także w innym systemie baz danych (np. LibreOffice Base). |
| Formularze, czyli ułatwiamy sobie wprowadzenie informacji do bazy danych | – z pomocą nauczyciela omawia przykład zamieszczony w podręczniku. | – wymienia nazwy róż- nych rodzajów formularzy i na podstawie podręcz- nika omawia ich cechy i zastosowanie;  – na podstawie opisów tworzy różne rodzaje formularzy za pomocą kreatora. | – przy tworzeniu formularzy korzysta z różnych rodzajów list. | – konstruuje formularze z widoku projektu, wykorzystując formaty;  – używa przycisków nawigacyjnych w formularzu. | – czynności opisane dla Access wykonuje także w innym systemie baz danych (np. LibreOffice Base). |
| Czy to możliwe, czyli baza danych w arkuszu Excel | – omawia analogie w tworzeniu tabel w Access i Excel. | – na podstawie podręcz- nika lub tutoriali tworzy w Excelu tabele  z nagłówkami;  – na podstawie podręcz- nika lub tutoriali używa formularzy do wprowa- dzania danych do tabel;  – na podstawie podręcz- nika lub tutoriali tworzy listę rozwijaną. | – samodzielnie tworzy w Excelu tabele  z nagłówkami;  – samodzielnie używa formularzy do wprowa- dzania danych do tabel;  – samodzielnie tworzy listę rozwijaną;  – stosuje sortowanie wegług wskazanych kryteriów;  – wstawia narzędzia do paska Szybki dostęp. | – planuje i wykonuje czynności tworzenia bazy w Excelu;  – dobiera kryteria sortowania;  – przenosi tabele z bazy danych do Excela. | – czynności opisane dla Excela wykonuje także w innym arkuszu (np. LibreOffice Calc). |
| Wiedza w sieci, czyli internet mądrych ludzi | – wyszukuje informacje w domyślnej wyszukiwarce przeglądarki internetowej;  – omawia znaczenie zachowania praw autorskich i podstawy licencji CC. | – na podstawie podręcz- nika zmienia domyślną wyszukiwarkę w przeglądarce Firefox;  – na podstawie podręcz- nika uszczegółowia dane w wyszukiwarce w celu zwiększenia prawdopo- dobieństwa dotarcia do szukanej informacji;  – na podstawie podręcz- nika wyszukuje za pomocą obrazu. | – uzasadnia celowość korzystania z różnych wyszukiwarek;  – pozyskuje informacje z baz danych, takich jak europeana.eu;  – tłumaczy podstawy i cel działania botów indeksujących;  – tłumaczy na przykładzie konieczność uszczegółowiania zapytań do wyszukiwarki;  – korzysta z zaawansowanych opcji wyszukiwarek;  – ocenia wiarygodność źródeł, wymieniając świadczące o niej cechy. | – planuje użycie odpowiedniej wyszukiwarki;  – wybiera bazę danych do wyszukiwania konkretnych informacji;  – trafnie dobiera słowa kluczowe, zarówno w opcjach głównych, jak i zaawansowanych wyszukiwarek;  – zmienia wyszukiwarki domyślne w różnych przeglądarkach internetowych. | – omawia cechy różnych wyszukiwarek internetowych i ocenia ich przydatność do konkretnego zadania. |
| Wirtualne serwery, czyli instalujemy Apache | – omawia pojęcie *serwer*;  – uruchamia środowisko XAMP. | – na podstawie podręcz- nika i doświadczeń z poprzednich lekcji instaluje i uruchamia środowisko XAMP. | – samodzielnie instaluje środowisko XAMP, wybierając wskazane komponenty;  – omawia zawartość XAMP i jego zastosowanie. | – decyduje o wyborze komponentów na podstawie informacji o późniejszym wykorzystaniu XAMP. | – instaluje i korzysta także z innych środowisk symulujących działanie serwera oraz omawia ich cechy. |
| Język zapytań, czyli poznajemy język SQL | – wyjaśnia, do czego służy język SQL. | – na podstawie podręcz- nika wymienia i krótko charakteryzuje podsta- wowe elementy składni SQL – DML, DDL, DCL, DQL;  – na podstawie podręcz- nika lub innych publikacji omawia typy danych języka SQL. | – na podstawie podręcz- nika omawia podstawowe elementy składni SQL;  – omawia przeznaczenie niektórych typów danych wykorzystywanych w MySQL. | – samodzielnie charakteryzuje podstawowe elementy składni SQL. | – omawia nieopisane w podręczniku elementy składni SQL. |
| Zadajemy pytania, czyli kwerendy w języku SQL | – na podstawie wiedzy z poprzednich lekcji omawia definicję i zastosowanie kwerendy;  – na podstawie podręcznika uruchamia elementy XAMP niezbędne do korzystania z bazy danych;  – przy pomocy nauczyciela i na podstawie opisu wykonuje ćwiczenia z podręcznika;  – na podstawie podręcznika lub innych źródeł omawia różnicę między SQL a MySQL. | – na podstawie podręcz- nika omawia definicję MySQL jako otwarto- źródłowego systemu zarządzania bazami danych;  – na podstawie podręcz- nika uruchamia phpMyAdmin i jego narzędzia z pakietu XAMP;  – na podstawie podręcz- nika i z niewielką pomocą nauczyciela tworzy nową bazę i tabele w MySQL;  – na podstawie podręcz- nika i z niewielką pomocą nauczyciela tworzy zapytania w SQL. | – na podstawie podręcz- nika wykonuje ćwiczenie i wyjaśnia poszczególne etapy tworzenia zapytań;  – omawia przykładowy kod tworzenia tabeli w SQL;  – na podstawie tabeli omawia operatory wykorzystywane podczas wyszukiwania informacji w bazie. | – samodzielnie tworzy bazę i zapytania w języku SQL;  – samodzielnie korzysta z MySQL. | – samodzielnie tworzy programy w SQL inne niż w przykładach podanych w podręczniku. |
| Więcej pytań, czyli jeszcze o kwerendach w języku SQL | – uruchamia w XAMP przykładowe programy SQL z podręcznika lub przedstawione przez nauczyciela. | – na podstawie podręcz- nika omawia składnię polecenia SELECT;  – na podstawie podręcz- nika omawia działanie klauzuli WHERE i operatorów logicznych;  – na podstawie podręcz- nika omawia efekt łączenia tabel w celu wyszukiwania informacji;  – na podstawie podręcz- nika analizuje umieszczone w nim przykłady w SQL. | – samodzielnie omawia składnię polecenia SELECT;  – samodzielnie omawia działanie klauzuli WHERE i operatorów logicznych;  – samodzielnie omawia efekt łączenia tabel w celu wyszukiwania informacji;  – samodzielnie analizuje umieszczone w podręcz- niku przykłady w SQL;  – sporadycznie korzystając ze źródeł wiedzy, wykonuje ćwiczenia z podręcznika. | – wykorzystuje zapytanie SELECT i operatory logiczne podczas tworzenia zapytań;  – łączy tabele w celu przeszukania większej ich liczby;  – wyjaśnia, jak szukać informacji w tabelach niepołączonych ze sobą. | – operuje MySQL i układa programy w SQL w przykładach innych niż poznane w trakcie lekcji. |
| Podzapytania, czyli instrukcje modyfikujące dane w języku SQL | – na podstawie podręcznika wyjaśnia pojęcie *podzapytania*. | – wyjaśnia pojęcie *podzapytania*;  – na podstawie podręcznika omawia znaczenie podzapytań. | – na podstawie podręcz- nika wykonuje opisane w nim ćwiczenie;  – wyjaśnia istotę podzapytań wierszowych i skorelowanych;  – wyjaśnia istotę podzapytań o charakterze tabeli tymczasowej;  – omawia przykłady programów SQL realizujących podzapytania. | – układa programy podzapytań w SQL do przykładów z podręcznika lub podanych przez nauczyciela. | – samodzielnie układa programy podzapytań w SQL. |
| Modyfikacje, czyli zarządzanie bazą danych | – na podstawie podręcznika wyjaśnia działanie poleceń: UPDATE, ALERT TABLE, DELETE, DELETE FROM. | – na podstawie podręcz- nika omawia składnię poleceń: UPDATE, ALERT TABLE, DELETE, DELETE FROM. | – z pomocą nauczyciela lub na podstawie podręcznika modyfikuje elementy bazy danych. | – samodzielnie, według zaleceń modyfikuje elementy (zmienia strukturę, usuwa rekordy, tabele) i strukturę bazy danych. | – samodzielnie określa zakres modyfikacji bazy ze względu na zaistniałe potrzeby i modyfikuje elementy i strukturę bazy danych. |
| Bezpieczeństwo bazy, czyli tworzymy kopie zapasowe | – omawia zasady dbania o bezpieczeństwo baz danych;  – podaje, czym jest kopia zapasowa. | – na podstawie podręcz- nika omawia istotę najczęściej spotykanych ataków na bazę (pasywne i aktywne);  – wymienia zasady, jakimi powinien się kierować administrator bazy w celu zabezpieczenia jej przed nieuprawnionym dostępem lub skasowaniem danych. | – omawia na przykładach definicje ataków pasywnych i aktywnych;  – omawia zasady, jakimi powinien się kierować administrator bazy w celu zabezpieczenia jej przed nieuprawnionym dostępem lub skasowaniem danych;  – omawia działanie poleceń: CHECK TABLE, REPAIR TABLE, RESET MASTER, PURGE MASTER, SHOW BINLOG EVENTS;  – omawia składnię poleceń do odzyskiwania danych z kopii bezpieczeństwa. | – tworzy kopię bezpieczeństwa, używając odpowiednich poleceń;  – wyjaśnia pojęcie przyrostowej kopii bezpieczeństwa;  – odzyskuje dane z kopii bezpieczeństwa. | – planuje zabezpieczenie bazy danych. |
| **IX. Programowanie stron internetowych** | | | | | |
| HTML, czyli przeglądarka interpretuje język programowania stron | – omawia przeznaczenie języka HTML;  – omawia rolę znaczników w języku HTML. | – omawia rolę przeglądar- ki internetowej w kontek- ście języka HTML;  – instaluje, uruchamia i korzysta ze wskazanego edytora (np. Notepad+) skonfigurowanego dla HTML5;  – korzysta z opisów naj- częściej używanych znacz- ników (np. z podręcznika). | – na podstawie podręcz- nika lub innych wiarygod- nych źródeł konfiguruje edytor do układania programów w HTML5;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł układa proste programy stron, np. wyświetlające tekst z zachowaniem prawid- łowej struktury programu;  – samodzielnie objaśnia rolę CSS w projektowaniu wyglądu strony. | – dobiera odpowiedni edytor do edycji programów w HTML5 i uzasadnia swój wybór;  – konfiguruje edytor;  – tworzy proste strony internetowe zawierające sformatowany tekst. | – tworzy rozbudowane strony internetowe w języku HTML5. |
| Budujemy stronę, czyli tabele, listy i inne elementy dobrej strony | – omawia znaczenie list w programie HTML5;  – uruchamia w przeglądarce gotowy program strony w HTML5 (np. z przykładu z podręcznika). | – na podstawie podręcz- nika uruchamia programy z przykładów;  – na podstawie przykła- dów omawia rolę list, definicji i tabel;  – tłumaczy istotę i przeznaczenie hipertekstu. | – na podstawie podręcz- nika lub innych wiary- godnych źródeł tworzy w kodzie HTML listy uporządkowane i nieuporządkowane oraz omawia ich znaczenie;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł tworzy listy definicji z dodanymi nagłówkami;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł tworzy tabele w HTML. | – samodzielnie tworzy w kodzie HTML listy uporządkowane i nieuporządkowane oraz omawia ich znaczenie;  – samodzielnie tworzy listy definicji z dodanymi nagłówkami;  – samodzielnie tworzy tabele w HTML;  – planuje użycie odpowiednich konstrukcji do projektowanej strony;  – używa atrybutów w konstrukcji hipertekstu. | – tworzy rozbudowane strony internetowe w języku HTML5. |
| Tabele i grafika, czyli kolejne składowe stron internetowych w HTML | – wymienia zasady przygotowania grafiki do publikacji na stronie z uwzględnieniem wymiarów i rozdzielczości. | – na podstawie podręcz- nika wstawia przygoto- waną przez nauczyciela grafikę do gotowego kodu strony;  – na podstawie podręcz- nika tłumaczy znaczenie wymiarów i skalowania dla szybkości wczytywania strony. | – określa parametry zdjęcia, biorąc pod uwagę jego miejsce na stronie;  – zmienia parametry zdjęcia przeznaczonego na stronę za pomocą prostych narzędzi systemu Windows. | – planuje miejsce na grafikę;  – przygotowuje grafikę do publikacji na stronie;  – umieszcza zdjęcia w planowanym miejscu i formacie na stronie w HTML;  – tworzy odnośnik z elementu graficznego umieszczonego na stronie w HTML. | – tworzy rozbudowane strony internetowe w języku HTML5 z zastosowaniem elementów graficznych. |
| Składnia stylów, czyli jak CSS pomaga w programowaniu wyglądu strony | – objaśnia rolę CSS w kreowaniu wyglądu strony. | – objaśnia istotę pliku stylów i jego połączenie z HTML. | – na podstawie podręcz- nika lub innych wiarygod- nych źródeł omawia zasady łączenia CSS z HTML;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł układa proste pliki stylów CSS, formatujące sposób wyświetlania tekstu;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł wyjaśnia znaczenie nagłówków i bloków w stosowaniu CSS;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł wyjaśnia znaczenie sekcji i selektora. | – samodzielnie omawia zasady łączenia CSS z HTML;  – samodzielnie układa proste pliki stylów CSS, formatujące sposób wyświetlania tekstu;  – samodzielnie wyjaśnia znaczenie nagłówków i bloków w stosowaniu CSS;  – samodzielnie wyjaśnia znaczenie sekcji i selektora;  – projektuje użycie CSS do formatowania stylu tekstu na stronie. | – tworzy rozbudowane strony internetowe w języku HTML5 z zastosowaniem CSS. |
| Pliki stylów, czyli CSS w akcji | – objaśnia działanie i znaczenie odsyłaczy na stronie w HTML;  – omawia rolę menu na stronie. | – na podstawie podręcz- nika lub innych wiarygod- nych źródeł omawia znaczenie selektorów i ich atrybutów w pliku CSS;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł omawia pojęcie *walidacja*. | – na podstawie podręcz- nika lub innych wiarygod- nych źródeł tworzy pliki HTML i CSS z menu z przyciskami, wykorzys- tując przykłady;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł używa pseudoklas z przykładu do określenia interakcji klawisza z kursorem;  – na podstawie podręcznika lub innych wiarygodnych źródeł rozmieszcza elementy strony, tworząc jej prosty layout. | – samodzielnie tworzy pliki HTML i CSS z menu z przyciskami, wykorzystując przykłady;  – samodzielnie używa pseudoklas z przykładu do określenia interakcji klawisza z kursorem;  – samodzielnie rozmiesz- cza elementy strony, tworząc jej prosty layout;  – zmienia wygląd i atrybu- ty menu z przykładu;  – zmienia położenie elementów strony;  – stosuje różną orientację elementów menu;  – przeprowadza walidację strony. | – tworzy rozbudowane strony internetowe w języku HTML5 z zastosowaniem CSS zawierające menu i efekty interakcji z kursorem. |
| Treści mogą się zmieniać, czyli elementy dynamiczne na stronie internetowej | – wskazuje elementy istniejących stron, które zmieniają się dynamicznie;  – na podstawie podręcz- nika wskazuje różnice między statyczną a dynamiczną strona internetową. | – omawia sposoby wklejania w kod strony gotowych odnośników do elementów dynamicznych;  – na podstawie podręcz- nika umieszcza w kodzie strony elementy dynamiczne. | – tworzy element dynamiczny zewnętrzny za pomocą darmowego edytora aplikacji (np. learningapps.com);  – pobiera i umieszcza na stronie kod urucha- miający widżety lub aplikacje internetowe z learningapps.com. | – planuje umieszczenie na stronie widżetów i innych elementów dynamicz- nych;  – dokonuje prawidłowego wyboru widżetów;  – projektuje elementy dy- namiczne (np. krzyżówki) w learningapps.com. | – tworzy strony z własnymi elementami dynamicznymi. |
| Widoczna w internecie, czyli jak opublikować stronę | – omawia funkcje domeny internetowej;  – wskazuje przynajmniej jedną firmę (stronę internetową) zajmującą się rejestracją domen. | – omawia proces wczytywania strony internetowej do przeglądarki i rolę domeny;  – omawia znaczenie niektórych domen (np. .com, .pl, .edu);  – podaje, jakie usługi musi uruchomić lub zamówić, by strona była widoczna w sieci. | – na podstawie podręcz- nika lub innych wiarygod- nych źródeł opisuje na przykładzie proces rezerwacji domeny;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł zakłada konta na darmowym serwerze z usługą hostingową (np. cba.pl);  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł wysyła pliki strony do serwera www. | – samodzielnie opisuje na przykładzie proces rezerwacji domeny;  – samodzielnie zakłada konta na darmowym serwerze z usługą hostingową (np. cba.pl);  – samodzielnie wysyła pliki strony do serwera www;  – sprawdza zajętość domen, którymi jest zainteresowany;  – korzysta z klienta FTP nie tylko do przesyłania plików strony. | – podaje, jak zarządzać domeną i usługą hostingową na przykładzie swojego konta. |
| CMS, czyli system zarządzania treścią strony internetowej | – omawia definicję CMS-u;  – podaje, do czego służy CMS i wymienia kilka najpopularniejszych (w tym Joomla i WordPress). | – omawia proces przy- gotowań do instalacji CMS-u;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł instaluje środowisko serwerowe (np. Laragon), a w nim CMS (np. WordPress);  – na podstawie podręcz- nika lub innych wiarygodnych źródeł loguje się do panelu sterowania CMS-em (np. WordPress). | – samodzielnie instaluje środowisko serwerowe (np. Laragon), a w nim CMS (np. WordPress);  – samodzielnie loguje się do panelu sterowania CMS-em (np. WordPress). | – dobiera odpowiedni do tematu strony CMS i uzasadnia swój wybór;  – wyjaśnia konieczność instalacji bazy danych dla CMS-a;  – używa wirtualnego środowiska serwera do testowania działania różnych stron internetowych. | – buduje ciekawe strony internetowe za pomocą różnych CMS-ów. |
| Panel i skórki, czyli tworzymy stronę w CMS | – wyjaśnia znaczenie szablonu (skórki) w systemie CMS;  – przegląda oferowane w sieci skórki dla WordPressa. | – odróżnia motywy od szablonów;  – omawia znaczenie i funkcję motywów w szablonach;  – na podstawie podręcznika wypełnia treścią stronę zbudowaną na bazie szablonu. | – na podstawie podręcz- nika lub innych wiarygod- nych źródeł instaluje w CMS-ie wskazany szablon;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł dobiera motyw i zmienia go w razie potrzeby;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł modyfikuje motyw skórki;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł wypełnia treścią stronę, tworząc nowe wpisy;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł umieszcza na stronie elementy graficzne. | – samodzielnie instaluje w CMS-ie wskazany szablon;  – samodzielnie dobiera motyw i zmienia go w razie potrzeby;  – samodzielnie modyfikuje motyw skórki;  – samodzielnie wypełnia treścią stronę, tworząc nowe wpisy;  – samodzielnie umieszcza na stronie elementy graficzne;  – trafnie i z uzasadnie- niem dobiera szablon i motyw do treści przyszłej strony;  – administruje stroną, wykorzystując panel sterowania;  – dodaje i usuwa użytkowników CMS-a i świadomie nadaje im uprawnienia;  – umieszcza na stronie filmy i inne elementy multimedialne. | – planuje i realizuje własne projekty stron na bazie CMS. |
| Szybkie i łatwe, czy programy do tworzenia stron internetowych | – podaje, jak zapisać tekstowy dokument Word w formacie HTML;  – wymienia kilka systemów darmowych blogów. | – na podstawie podręcznika lub innych wiarygodnych źródeł eksportuje do formatu HTML dokumenty Word i Excel;  – na podstawie podręcznika lub innych wiarygodnych źródeł zakłada konto w systemie darmowych blogów wskazanym przez nauczyciela. | – samodzielnie eksportuje do formatu HTML dokumenty Word i Excel;  – samodzielnie zakłada konto w systemie darmowych blogów wskazanym przez nauczyciela;  – formatuje dokument Word I Excel w taki sposób, by po eksporcie do HTML powstała estetyczna i funkcjonalna strona;  – wypełnia treścią blog utworzony w darmowym systemie blogów. | – prawidłowo i estetycznie rozmieszcza elementy graficzne, wzory itp. w eksportowanym dokumencie;  – testuje w środowisku serwerowym (np. Lara- gon) strony utworzone podczas eksportu dokumentów do HTML, zachowując układ folderów. | – prowadzi własnego bloga na ciekawy temat. |
| Pierwsze skrypty, czyli poznajemy PHP | – omawia specyfikę języka PHP (skrypt umieszczony w HTML i na odwrót);  – uruchamia środowisko Notepad++. | – na podstawie podręcz- nika omawia umieszcza- nie skryptów PHP w dokumencie HTML;  – uruchamia gotowe przykłady z podręcznika lub podane przez nauczyciela. | – omawia rolę funkcji w języku PHP, w tym: print(), echo(), printf();  – omawia działanie przykładowego skryptu, np. z podręcznika. | – układa prosty skrypt wyświetlający tekst;  – umieszcza skrypt na serwerze, uruchamia go w dokumencie HTML i testuje w przeglądarce. | – tworzy proste skrypty PHP. |
| Stałe i zmienne, czyli typy danych w języku PHP | – na podstawie podręcz- nika wyjaśnia, czym są stałe i zmienne w języku PHP. | – wymienia znaki, jakie mogą być użyte w nazwach zmiennych i stałych w języku PHP;  – na podstawie podręcznika omawia typy zmiennych używanych w języku PHP;  – na podstawie podręcz- nika analizuje przykłado- wy skrypt PHP, w którym zastosowano zmienne i stałe;  – wymienia operatorów w języku PHP i omawia ich działanie. | – analizuje przykładowy skrypt ze wskazaniem funkcji, jakie pełnią zmienne i stałe;  – omawia sposób definiowania zmiennych i stałych w PHP;  – prawidłowo używa operatorów. | – układa nieskomplikowa- ne skrypty według specyfikacji, w których używa zmiennych i stałych. | – układa na podstawie własnych specyfikacji i pomysłów proste skrypty PHP. |
| Instrukcje sterujące, czyli praktyczne wykorzystanie języka PHP | – wyjaśnia ogólnie, na czym polega działanie instrukcji warunkowych i wyboru. | – omawia działanie instrukcji warunkowych i wyboru w PHP przez analogię do podobnych instrukcji języka C++;  – na podstawie opisu omawia działanie przyk- ładowych skryptów PHP zawierających instrukcje warunkowe i wyboru. | – samodzielnie omawia działanie przykładowych skryptów PHP zawiera- jących instrukcje warunkowe i wyboru, uruchamia je i testuje. | – układa według specyfikacji skrypty PHP zawierające instrukcje warunkowe i wyboru. | – układa na podstawie własnych specyfikacji i pomysłów skrypty PHP zawierające instrukcje warunkowe i wyboru. |
| Pętle, czyli jeszcze więcej możliwości wykorzystania języka PHP | – wyjaśnia ogólnie, na czym polega działanie pętli w algorytmie i w języku programowania. | – omawia działanie pętli: for, while, do while w PHP przez analogię do podob- nych instrukcji języka C++;  – wyjaśnia różnice między instrukcjami pętli. | – omawia działanie przykładowych skryptów PHP zawierających pętle;  – analizuje przykłady skryptów zawierających pętle. | – układa według specyfikacji skrypty PHP zawierające pętle. | – układa na podstawie własnych specyfikacji i pomysłów skrypty PHP zawierające pętle. |
| Funkcje w języku PHP, czyli jak wzbogacić naszą stronę | – na podstawie podręcznika objaśnia, na czym polega użycie funkcji w programie komputerowym. | – na podstawie podręcz- nika wyjaśnia różnice między funkcjami wbudo- wanymi a własnymi;  – na podstawie podręcznika wymienia nazwy i określa rolę jaką odgrywają niektóre funkcje wbudowane. | – samodzielnie wyjaśnia różnice między funkcjami wbudowanymi a własnymi;  – samodzielnie wymienia nazwy i określa rolę, jaką odgrywają niektóre funkcje wbudowane;  – analizuje przykłady skryptów ułożonych w PHP z użyciem funkcji;  – testuje przykłady skryptów PHP zawierających funkcje. | – omawia rolę niektórych funkcji wbudowanych;  – tworzy skrypt PHP według specyfikacji z użyciem funkcji wbudowanych i testuje ich działanie. | – układa skrypty PHP zawierające funkcje własne. |
| Tablice, czyli jak sortować dane w języku PHP | – omawia na realnym przykładzie istotę tablic. | – omawia różnice między zmiennymi tablicowymi w języku C++ a PHP;  – omawia różnice między tablicami jedno- a wielowymiarowymi;  – uruchamia przykładowe skrypty PHP zawierające tablice i na podstawie podręcznika omawia ich działanie. | – samodzielnie analizuje działanie przykładowych skryptów PHP zawierających tablice. | – układa według specyfikacji skrypty zawierające tablice. | – układa według własnych specyfikacji i pomysłów skrypty PHP zawierające tablice. |
| Pobieranie i przetwarzanie danych, czyli obsługa formularza w języku PHP | – omawia istotę i cele stosowania formularzy na stronach internetowych. | – na podstawie opisu omawia sedno metod POST i GET stosowanych w budowaniu formularzy;  – uruchamia i testuje przykładowe skrypty z formularzami. | – samodzielnie analizuje działanie przykładowych skryptów PHP zawiera- jących formularze;  – proponuje zmiany w przykładowych formularzach w związku ze zmianą specyfikacji skryptu. | – układa według specyfikacji skrypty tworzące na stronie formularze, testuje je i modyfikuje;  – używa w swoich skryptach obu metod i uzasadnia ich wybór. | – układa według własnych specyfikacji i pomysłów skrypty PHP zawierające zaprojektowane przez siebie formularze. |
| **X. Algorytmy i programowanie** | | | | | |
| Powtarzanie w pętli i wywołanie siebie, czyli iteracja i rekurencja w algorytmach | – omawia na realnych przykładach różnicę między rekurencją a iteracją. | – na podstawie podręcz- nika lub innych wiarygod- nych źródeł analizuje przykładowe algorytmy iteracyjne i rekurencyjne;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł rozpoznaje procesy rekurencyjne i iteracyjne. | – samodzielnie analizuje przykładowe algorytmy iteracyjne i rekurencyjne;  – samodzielnie rozpoz- naje procesy rekurencyjne i iteracyjne;  – wskazuje w przykłado- wych algorytmach miejsca, które decydują o iteracyjności lub rekurencyjności opisywanego procesu;  – analizuje przykładowy program. | – układa algorytmy z podejściem iteracyjnym i rekurencyjnym;  – układa programy na podstawie tych algorytmów;  – wskazuje instrukcje, które decydują o itera- cyjności lub rekuren- cyjności podejścia do realizacji algorytmu. | – rozwiązuje problemy obiema metodami i ocenia ich skuteczność. |
| Sortowanie bąbelkowe, czyli każda liczba jest mniejsza od maksymalnej lub jej równa | – na podstawie podręcz- nika lub innych wiarygod- nych źródeł omawia budowę tablicy jedno- wymiarowej;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł omawia istotę sortowania bąbelkowego. | – na podstawie podręcz- nika lub innych wiarygod- nych źródeł analizuje działanie algorytmu sortowania bąbelkowego w postaci listy kroków i schematu blokowego;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł analizuje przykład sprawdzający poprawność działania algorytmu. | – samodzielnie omawia istotę metody sortowania bąbelkowego;  – samodzielnie omawia działanie przykładowego algorytmu opartego o metodę sortowania bąbelkowego;  – samodzielnie sprawdza działanie algorytmu na przykładach. | – układa algorytm sortowania bąbelkowego;  – weryfikuje poprawność działania programu na przykładach. | – układa program sortujący metodą bąbelkową w innym języku niż C++ (np. Java). |
| Przez wstawianie, czyli jeszcze o porządkowaniu liczb | – na podstawie podręcz- nika lub innych wiarygod- nych źródeł omawia istotę sortowania przez wstawianie. | – na podstawie podręcz- nika lub innych wiarygod- nych źródeł analizuje działanie algorytmu sortowania bąbelkowego w postaci listy kroków i schematu przez wstawianie;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł analizuje przykład sprawdzający poprawność działania algorytmu. | – samodzielnie omawiana przykładzie istotę metody sortowania przez wstawianie;  – samodzielnie omawia działanie przykładowego algorytmu opartego o metodę sortowania przez wstawianie;  – samodzielnie sprawdza działanie algorytmu na przykładach. | – układa algorytm sortowania przez wstawianie w postaci listy kroków i schematu blokowego;  – weryfikuje poprawność działania programu na przykładach. | – układa program sortujący metodą przez wstawianie w innym języku niż C++ (np. Java). |
| Komputer porządkuje, czyli układamy programy sortujące | – na podstawie podręcz- nika lub innych wiarygod- nych źródeł weryfikuje działanie przykładowych programów (np. z pod- ręcznika). | – na podstawie podręcz- nika lub innych wiarygod- nych źródeł analizuje przykładowe programy i wskazuje miejsca, w których są wykonywane działania najważniejsze dla metody. | – na podstawie podręcz- nika lub innych wiarygod- nych źródeł układa programy sortujące;  – na podstawie podręcz- nika uzasadnia użycie danych instrukcji w przykładowych programach. | – samodzielnie układa programy sortujące według obu metod;  – samodzielnie weryfikuje poprawność działania tych programów;  – wskazuje instrukcje i rozkazy realizujące istotę metody. | – proponuje inne od przykładowych rozwiązania programowe;  – układa programy realizujące algorytmy sortowania bąbelkowego i przez wstawianie według własnego pomysłu. |
| Fibonacci i jego wzór, czyli generujemy kolejne liczby ciągu | – wie, kim był i kiedy żył Fibonacci;  – zna jego zasługi dla rozwoju cywilizacji. | – na podstawie podręcz- nika lub innych wiarygod- nych źródeł omawia na przykładzie istotę ciągu Fibonacciego;  – na podstawie podręcz- nika lub innych wiarygod- nych źródeł analizuje działanie przykładowego algorytmu obliczającego kolejne elementy ciągu, zapisanego w postaci schematu blokowego. | – samodzielnie omawia na przykładzie istotę ciągu Fibonacciego;  – samodzielnie analizuje działanie przykładowego algorytmu obliczającego kolejne elementy ciągu, zapisanego w postaci schematu blokowego;  – układa algorytm obliczający określoną liczbę liczb ciągu Fibonacciego;  – analizuje program w języku C++ ułożony według przykładowego algorytmu. | – układa program w języku C++ obliczający *n* kolejnych elementów ciągu Fibonacciego;  – testuje poprawność działania swojego programu na przykładzie. | – proponuje rozwiązanie problemu obliczania *n*-tego elementu ciągu Fibonacciego. |
| Szukamy lidera, czyli porządkowanie zbioru metodą binarnego wyszukiwania | – omawia, na czym polega metoda dziel i zwyciężaj;  – rozumie, na czym polega umieszczanie danych w tablicy. | – na podstawie podręcz- nika opisuje zastosowanie wyszukiwania binarnego w sortowaniu. | – omawia na przykładzie działanie algorytmu sortowania zbioru z wykorzystaniem wyszukiwania binarnego. | – testuje algorytm sortowania zbioru z wykorzystaniem wyszukiwania binarnego na dowolnych danych;  – układa pogram na podstawie algorytmu. | – stosuje algorytm dla tablicy wielowymiarowej. |
| Przecinając oś, czyli wyznaczenie miejsc zerowych funkcji metodą połowienia | – podaje definicję miejsca zerowego funkcji i wskazuje je na wykresie. | – na podstawie podręcz- nika omawia metodę połowienia. | – analizuje algorytm metody połowienia z podręcznika. | – na podstawie opisu metody opracowuje algorytm metody połowienia i na jego podstawie układa program w języku C++;  – testuje algorytm i program. | – omawia inne metody znajdowania miejsca zerowego, np. metodę analityczną. |
| Prawie robi różnicę, czyli obliczanie przybliżonej wartości pierwiastka kwadratowego | – podaje, czym jest przybliżenie wartości pierwiastka i zna konsekwencje jego stosowania. | – na podstawie podręcz- nika omawia metodę Newtona-Robsona. | – omawia podstawy matematyczne metody Newtona-Robsona;  – analizuje omówiony w podręczniku algorytm i program metody Newtona-Robsona. | – na podstawie opisu metody samodzielnie opracowuje algorytm metody Newtona- -Robsona;  – układa program znajdujący przybliżoną wartość pierwiastka kwadratowego i go testuje. | – omawia inne metody obliczania wartości pierwiastka kwadratowego. |
| Szybkie potęgowanie liczb w wersji iteracyjnej, czyli jak obliczyć potęgę o wykładniku naturalnym | – podaje właściwości potęgi liczby 0;  – własnymi słowami opisuje, czym jest potęgowanie liczb i jak się je wykonuje;  – uruchamia program z podręcznika i testuje jego działanie;  – podaje, na czym polega iteracyjne podejście do rozwiązywania problemów. | – na podstawie podręcz- nika omawia algorytm iteracyjny metody szybkiej potęgowania i porównuje ją z metodą kolejnego mnożenia przez podstawę potęgi. | – samodzielnie omawia metodę szybkiego potęgowania w wersji iteracyjnej;  – analizuje algorytm i program w języku C++ metody szybkiego potęgowania. | – omawia na przykładzie zalety metody szybkiego potęgowania;  – układa algorytm metody szybkiego potęgowania i na jego podstawie program w języku C++ w wersji iteracyjnej. | – porównuje podejście iteracyjne i rekurencyjne do obliczania potęgi metodą szybką. |
| Szybkie potęgowanie liczb w wersji rekurencyjnej, czyli jak obliczyć potęgę o wykładniku naturalnym | – uruchamia program z podręcznika i testuje jego działanie;  – podaje, na czym polega rekurencyjne podejście do rozwiązywania problemów. | – na podstawie podręcz- nika omawia rekurencyjne podejście do metody potęgowania szybkiego. | – samodzielnie omawia metodę szybkiego potęgowania w wersji rekurencyjnej;  – analizuje algorytm i program w języku C++ lub Java dla metody szybkiego potęgowania rekurencyjnego. | – układa program metody szybkiego potęgowania i na jego podstawie program w języku C++ lub Java w wersji rekurencyjnej. | – porównuje podejście iteracyjne i rekurencyjne do obliczania potęgi metodą szybką. |
| Suma binarna, czyli dodawanie w systemie dwójkowym | – omawia podstawy systemu dwójkowego;  – dodaje do siebie pisemnie niewielkie liczby binarne. | – zamienia reprezentacje liczb binarnych na dziesiętne i odwrotnie. | – układa algorytm dodawania binarnego i go testuje. | – układa program w języku C++ realizujący algorytm dodawania liczb binarnych i go testuje. | – układa program na podstawie algorytmu także w innym języku (np. Java). |
| Różnica binarna, czyli odejmowanie w systemie dwójkowym | – odejmuje pisemnie niewielkie liczby binarne. | – opisuje metodę odejmowania liczb binarnych i tłumaczy rolę pożyczki na przykładzie. | – układa algorytm odejmowania binarnego i go testuje. | – układa program w języku C++ realizujący algorytm odejmowania liczb binarnych i go testuje. | – układa program na podstawie algorytmu także w innym języku (np. Java). |
| Iloczyn binarny, czyli mnożenie w systemie dwójkowym | – mnoży pisemnie niewielkie liczby binarne. | – opisuje własnymi słowami na przykładzie metodę mnożenia liczb binarnych. | – układa algorytm mnożenia binarnego i go testuje. | – układa program w języku C++ realizujący algorytm mnożenia liczb binarnych i go testuje. | – układa program na podstawie algorytmu także w innym języku (np. Java). |
| Iloraz binarny, czyli dzielenie w systemie dwójkowym | – dzieli pisemnie niewielkie liczby binarne. | – opisuje własnymi słowami na przykładzie metodę dzielenia liczb binarnych. | – układa algorytm dzielenia binarnego i go testuje. | – układa program w języku C++ realizujący algorytm dzielenia liczb binarnych i go testuje. | – układa program na podstawie algorytmu także w innym języku (np. Java). |
| Metoda Monte Carlo, czyli jak obliczyć przybliżoną wartość liczby *pi* | – omawia znaczenie liczby *pi* na przykładach (np. obliczania obwodu koła). | – opisuje własnymi słowami na przykładzie, korzystając z rysunku pomocniczego, metodę Monte Carlo obliczania kolejnych pozycji liczby *pi*;  – zna wzór na liczbę *pi* z metody Monte Carlo. | – układa algorytm w dowolnym zapisie obliczający kolejne pozycje liczby *pi* metodą Monte Carlo. | – układa program w języku C++ realizujący algorytm Monet Carlo. | – układa program na podstawie algorytmu także w innym języku (np. Java). |
| **XI. Rozwiązywanie problemów z wykorzystaniem komputera w nauce i firmie** | | | | | |
| Nowoczesna firma, czyli jak programy komputerowe ułatwiają pracę | – wymienia typowe dla pracy biurowej zastosowania programów komputerowych  – wymienia podstawowe programy wykorzystywane w biurach: edytory tekstu, arkusze, pakiety Office i LibreOffice | – wymienia zawody, w których komputery i programy komputerowe wydatnie wpływają na poprawienie komfortu i wydajności pracy  – podaje przynajmniej jeden przykład zastosowania komputerów i programów w różnych zawodach | – samodzielnie opracowuje prosty arkusz, np. cennik i omawia jego wykorzystanie w działalności gospodarczej lub firmie  – podaje przykłady zastosowania programów komputerowych do prezentacji w przedsiębiorstwie i działalności gospodarczej  – podaje przykład zastosowania kalendarza i innych organizerów w pracy zawodowej | – samodzielnie tworzy inny niż w podręcznikowym przykładzie arkusz wspomagający pracę, np. kalkulator kosztów itp.  – samodzielnie omawia znaczenie chmur informatycznych w pracy zawodowej i nauce na podstawie przykładów | – samodzielnie, od dłuższego czasu, korzysta z niektórych programów prezentowanych na lekcji, np. kalendarza, Sway itp. |
| Kalkulujemy, czyli jak wykorzystać arkusz kalkulacyjny w zarządzaniu finansami | – umie posługiwać się prostym, gotowym arkuszem z listami rozwijanymi | – wykonuje w arkuszu listę rozwijaną na podstawie opisu z podręcznika  – omawia zastosowanie symulacji w arkuszu i uzasadnia ich stosowanie  – podaje przykłady zastosowania symulacji w arkuszu  – tworzy arkusz z listą rozwijaną na podstawie opisu z podręcznika  – samodzielnie omawia zastosowanie arkuszy z podręcznika | – uzasadnia stosowanie listy rozwijanej dla danej komórki w tabeli arkusza i modyfikuje go  – omawia, na przykładzie, działanie formuły warunkowej  – wie, na czym polega zagnieżdżanie formuł, np. warunkowej | – samodzielnie projektuje i tworzy arkusz z listami rozwijanymi inny niż w przykładzie w podręczniku  – samodzielnie układa formułę z zagnieżdżonymi formułami warunkowymi i uzasadnia ich zastosowanie | – samodzielnie opracowuje arkusze na zadany temat, służące np. symulacji zjawisk sugerowanych przez nauczyciela lub na podstawie własnego pomysłu |
| Z sieci do tabeli, czyli jak interpretować dane za pomocą arkusza kalkulacyjnego | – podaje przykłady stron, na których publikowane są dane w postaci tabel  – umie pobrać ze strony internetowej plik z tabelą  – interpretuje wizualizację danych z tabeli | – wyszukuje w internecie tabele z danymi na dany temat, np. dotyczący rankingów szkół i uczelni  – na podstawie opisu z podręcznika pobiera dane z tabel ze stron internetowych i dokumentów tekstowych  – na podstawie opisu z podręcznika dobiera rodzaj wizualizacji danych w arkuszu | – omawia znaczenie przenoszenia danych z publikacji internetowych i plików tekstowych do arkusza  – samodzielnie prawidłowo dobiera rodzaj wizualizacji danych z tabel i uzasadnia swój wybór  – samodzielnie pobiera dane z tabel ze stron internetowych i dokumentów tekstowych  – na podstawie podręcznika stosuje sortowanie w tabelach arkusza  – na podstawie opisów np. z podręcznika wymienia i używa narzędzi arkusza do pobierania danych z różnych źródeł | – samodzielnie odnajduje tabele z danymi na zadany temat i ich zawartość wyświetla w arkuszu  – samodzielnie przenosi dane pomiędzy arkuszami  – formatuje wykresy danych, np. wyświetla dokładną wartość słupka lub plastra  – samodzielnie używa narzędzi do sortowania danych w tabelach  – samodzielnie stosuje narzędzia arkusza do importowania danych do tabel | – samodzielnie odnajduje, pobiera, sortuje i wizualizuje dane na zadany lub samodzielnie zaproponowany temat dotyczący przedmiotów szkolnych |
| Spośród wielu, czyli filtrowanie w arkuszu kalkulacyjnym | – na podstawie podręcznika prawidłowo uzasadnia przydatność sortowania i filtrowania danych w arkuszu  – wie, jakie można wybrać kryteria sortowania danych, np. tekstów i liczb  – wie, że można stosować filtrowanie przy użyciu wielu kryteriów jednocześnie  – z pomocą nauczyciela wymienia parametry, wg których można filtrować dane | – podaje przykłady, w których zastosowanie filtrowania ułatwia interpretację lub wyszukiwanie danych  – na podstawie opisu z podręcznika używa filtra liczb, np. Między  – na przykładzie z podręcznika uzasadnia przydatność filtrowania przy użyciu wielu kryteriów jednocześnie  – wie, czym jest Fragmentator  – czytając definicje koniunkcji i alternatywy, umie je prawidłowo zinterpretować | – na podstawie podręcznika dobiera filtry odpowiednie do rozwiązania problemu  – używa prawidłowo pojęć *koniunkcja* i *alternatywa*  – na podstawie podręcznika używa różnych filtrów, w tym także tekstów i kolorów, uzasadniając ich wybór  – używa filtrów zakresów danych  – używa Fragmentatora na podstawie opisu z podręcznika | – samodzielnie dobiera filtry do rozwiązania konkretnego problemu z wyświetlaniem danych z tabel z wieloma komórkami  – samodzielnie ocenia skuteczność zastosowanego filtra  – uzasadnia zastosowanie danego filtra  – samodzielnie korzysta z Fragmentatora | – samodzielnie i świadomie dobiera rodzaj filtrowania dla zadanej tabeli z danymi i prawidłowo uzasadnia wybór; używa przy tym pojęć informatycznych i matematycznych  – omawia przyczynę koniunkcji filtrów we Fragmentatorze |
| Z eksperymentu do arkusza, czyli analiza danych z doświadczenia | – uzasadnia zasadność stosowania symulacji komputerowych różnych zjawisk i procesów  – omawia przydatność symulacji dla przykładów z podręcznika  – uzasadnia przydatność umieszczania danych z wyników doświadczeń, np. pomiarów do arkusza kalkulacyjnego | – odtwarza w arkuszu przykłady z podręcznika i omawia ich przydatność, np. wizualizację wyników | – rozumie sens wykonania doświadczenia z rzucaniem kostką do gdy i tłumaczy wpływ liczby rzutów na wyniki  – wie, jakie znaczenie w symulacji może mieć generator liczb pseudolosowych  – na podstawie podręcznika interpretuje wyniki doświadczenia symulującego rzut kostką wykonanego w arkuszu  – porównuje wykres i wyniki doświadczenia ze wzorami prawa Ohma | – samodzielnie tworzy arkusz do symulacji rzutu kostką  – bada wpływ liczby rzutów kostką na wyniki symulacji  – samodzielnie omawia wyniki doświadczenia z obwodem elektrycznym i uzasadnia zastosowanie wykresu liniowego | – tworzy symulację zdarzeń rzutu kilkoma kostkami i interpretuje wyniki  – podaje przykład doświadczenia fizycznego i projektuje dla niego arkusz pomagający w interpretacji wyników  – doświadczalnie określa próg liczby rzutów kostką, powyżej którego wyniki dla poszczególnych oczek są zbliżone z zadaną dokładnością |
| Edytor grafiki w pracy zawodowej, czyli tworzymy reklamę | – wie, na czym polega stosowanie warstw i co można dzięki nim osiągnąć  – wymienia kilka nazw edytorów grafiki oferujących mechanizm warstw | – zna przeznaczenie podstawowych narzędzi edycyjnych  – posługuje się podstawowymi narzędziami edycyjnymi edytora grafiki, np. GIMP | – na podstawie opisu z podręcznika umie utworzyć ulotkę reklamową  – wykorzystuje warstwy do wklejania elementów graficznych i tekstu  – na podstawie podręcznika przeprowadza podstawową korektę zdjęcia | – samodzielnie tworzy estetyczną ulotkę reklamową z wykorzystaniem warstw i mechanizmów opisanych w podręczniku  – samodzielnie koryguje niektóre wady zdjęć | – biegle posługuje się edytorem grafiki rastrowej i tworzy grafikę wg własnego projektu |
| Reklama jest ważna, czyli jak wykonać atrakcyjną prezentację | – zna znaczenia dobrze zaplanowanej prezentacji  – umie uruchamiać prezentację  – zna znaczenie scenariusza prezentacji dla jej skuteczności | – na podstawie gotowego grafu, np. z podręcznika, omawia czynniki wpływające na jakość scenariusza prezentacji  – wie, że prezentacje można wykonać za pomocą różnych programów, w tym w chmurze, np. prezi.com  – wie, jak znaleźć i importować szablony prezentacji | – na podstawie opisu umie założyć darmowe konto w prezi.com i wie, do jakich zastosowań może je wykorzystać  – układa scenariusz prezentacji na zadany temat, np. dotyczący zawodu, w którym się kształci  – z niewielką pomocą, na podstawie scenariusza, tworzy prezentacje w programie LibreOffice Impress z wykorzystaniem różnych elementów medialnych  – na podstawie opisu tworzy nieskomplikowaną prezentacje w chmurze prezi.com | – samodzielnie tworzy scenariusz prezentacji na dany temat i na jego podstawie prezentacje w programie Impress lub prezi.com | – samodzielnie tworzy szablony w prezi.com i Impress |
| Prezentacja wideo, czyli jak przygotować prezentację filmową | – na podstawie opisów i ilustracji z podręcznika omawia podstawowe zasady filmowania, np. zachowanie osi filmowej i podaje przykłady  – umie opisać plany filmowe na podstawie ilustracji z podręcznika  – używa aplikacji ze swojego telefonu zapisującej zawartość wyświetlacza | – samodzielnie omawia znaczenie poszczególnych zasad obowiązujących w trakcie filmowania  – rejestruje filmy za pomocą telefonu i umie pobrać je na dysk komputera  – umie nazwać plany w oglądanej scenie filmowej  – na podstawie opisu rejestruje zawartość ekranu komputera i podaje przykłady zastosowania takich filmów | – rejestruje ujęcia krótkiej sceny filmowej z prawidłowym zastosowaniem planów filmowych i z zachowaniem osi  – na podstawie opisu ustawia parametry telefonu lub aplikacji w zależności od przeznaczenia rejestrowanego ujęcia | – samodzielnie dobiera parametry rejestrowanego ujęcia w zależności od przeznaczenia pliku  – samodzielnie, prawidłowo stosuje zasady filmowania  – uzasadnia wybór planu filmowego dla danego ujęcia  – samodzielnie rejestruje zawartość okna lub ekranu prawidłowo dobierając „filmowane” treści do tematu zadania | – podaje przykłady, np. z filmów, w których celowo złamano zasady filmowania i kadrowania, oraz interpretuje intencje operatora kamery  – samodzielnie opracowuje scenariusz filmu – tutoriala omawiającego wskazane przez nauczyciela lub obmyślane samodzielnie problemy informatyczne, np. montaż filmu |
| Multimedia w prezentacji, czyli dźwięk i film na slajdach | – podaje przykłady prezentacji lub ich tematy, w których zasadne jest użycie multimediów  – omawia zalety stosowania multimediów w prezentacjach | – na podstawie opisu z podręcznika rejestruje dźwięk i zapisuje go w postaci pliku  – wymienia podstawowe formaty plików zawierających dźwięk lub film  – na podstawie opisu umieszcza w prezentacji dźwięk lub film | – trafnie dobiera elementy multimedialne do tematyki prezentacji lub slajdu  – na podstawie opisu z podręcznika rejestruje i zapisuje dźwięk oraz umieszcza go w slajdach lub prezentacji  – wie, od czego zależy jakość dźwięku zapisanego w pliku | – sprawnie i trafnie dobiera ustawienia programu rejestrującego dźwięk w kontekście jakości i dopasowania do potrzeb prezentacji  – wyjaśnia, na czym polega wpływ na jakość zarejestrowanego dźwięku takich parametrów jak częstotliwość próbkowania i rozdzielczość bitowa (liczba bitów dla pojedynczego pomiaru)  – sprawnie i trafnie samodzielnie dobiera parametry zapisu  – trafnie dopasowuje dźwięk do prezentowanych slajdów | – samodzielnie opracowuje scenariusz prezentacji, z góry uwzględniający rolę dźwięku i video  – samodzielnie realizuje nagrania audio lub wideo |
| Skuteczne wsparcie, czyli jak przygotować pokaz prezentacji | – wie, że można zmieniać parametry odtwarzania multimediów w prezentacji PowerPoint  – umie drukować materiały informacyjne wspomagające prelegenta dla gotowej prezentacji  – wie, które opcje służą do zmiany parametrów multimediów na slajdach PowerPoint | – na odstawie opisu z podręcznika umie zmieniać niektóre parametry odtwarzania multimediów, np. miejsce wyświetlania filmu, m.in. wybór momentu jego startu  – wie, czym jest konspekt prezentacji i omawia jego znaczenie  – wie, jak odnaleźć opcje do formatowania multimediów na slajdach PowerPoint | – na podstawie opisu z podręcznika zmienia wskazane parametry odtwarzania multimediów  – przygotowuje materiały pomocnicze do prelekcji  – ustala sposób wyświetlania slajdów  – eksportuje prezentacje do różnych formatów | – samodzielnie i trafnie dobiera parametry odtwarzania multimediów na slajdach PowerPoint  – samodzielnie wybiera rodzaj materiałów promocyjnych projektu prezentacji  – pamięta o osadzaniu czcionek podczas zapisu prezentacji do użytku na różnych komputerach i objaśnia taką konieczność | – posługuje się biegle także innymi edytorami prezentacji |
| Atrakcyjnie i wygodnie, czyli jak upowszechniać prezentację | – wyjaśnia, w jakim celu prezentacje zapisuje się w różnych formatach  – wymienia urządzenia, na których można odtwarzać prezentacje zapisane w różnych formatach | – uruchamia prezentacje na telefonach i innych urządzeniach mobilnych  – objaśnia różnice pomiędzy różnymi formatami zapisu prezentacji  – wyjaśnia cel eksportowania prezentacji do innych formatów, np. graficznych (jpg) lub tekstowych (pdf)  – wie, które opcje pozwalają na ustawienia automatycznego pokazu, np. z chronometrażem | – samodzielnie ustala chronometraż pokazu slajdów  – eksportuje pokaz slajdów do formatów filmowych z zastosowaniem przejść między slajdami  – łączy komputer z projektorem multimedialnym  – wyświetla bezprzewodowo prezentacje z telefonu na ekranie telewizora lub projektora  – korzystając z podręcznika, przygotowuje pokaz wg opisanych przez nauczyciela warunków | – samodzielnie dobiera sposób prezentacji  – samodzielnie i trafnie ustala czas w chronometrażu  – samodzielnie i trafnie dobiera format zapisu prezentacji w zależności od sposobu i miejsca jej pokazu  – samodzielnie łączy urządzenia bezprzewodowe do prezentacji slajdów | – samodzielnie i trafnie określa wszystkie parametry i sposoby użycia prezentacji zgodnie z jej przeznaczeniem i miejscem wyświetlania |
| Utrzymujemy kontakt z klientami, czyli korespondencja seryjna | – omawia istotę korespondencji seryjnej i podaje przykłady jej wykorzystania | – wymienia źródła danych do korespondencji seryjnej  – umie posłużyć się gotowym dokumentem przygotowanym do korespondencji seryjnej, np. w celu drukowania kopert | – na podstawie opisów z podręcznika tworzy dokument tekstowy z polami korespondencji seryjnej i dołącza do niego dane  – omawia znaczenie reguł w korespondencji seryjnej | – samodzielnie tworzy dokumentu z polami korespondencji seryjnej i dołącza do nich dane  – samodzielnie używa reguł  – używa korespondencji seryjnej do adresowania kopert | – samodzielnie projektuje dokumentu z polami korespondencji seryjnej w różnych edytorach |
| **XII. Linux i inne systemy operacyjne** | | | | | |
| Różne sposoby instalacji, czyli przygotowujemy miejsce dla systemu Linux | – omawia funkcję wirtualnej maszyny i cel jej użycia  – wie, w jakim celu instaluje się Linux na nośniku zewnętrznym | – wymienia przykłady zastosowania systemu Linux  – definiuje jądro systemu Linux w kontekście jego wykorzystania | – samodzielnie instaluje wirtualną maszynę, np. VirtualBox  – wymienia i omawia sposoby instalacji Linux w komputerze PC  – omawia opcje programów do instalacji systemu na nośniku zewnętrznym | – wymienia i omawia znaczenie innych niż w komputerze PC zastosowań dystrybucji Linux  – wyjaśnia potrzebę uruchomienia opcji wirtualizacji w UEFI lub BIOS | – instaluje różne programy do wirtualizacji i omawia opcje oraz różnice pomiędzy nimi |
| Wiele wersji, czyli wybieramy dystrybucje Linux | – umie określić, czym jest dystrybucja systemu i odróżnia to pojęcie od rozpowszechniania i sprzedaży  – wie, czym jest GUI i omawia znaczenie takiego interfejsu | – pozyskuje wskazaną dystrybucję Linux  – na podstawie podręcznika instaluje system Linux w maszynie wirtualnej | – samodzielnie pobiera wybraną dystrybucję Linux  – instaluje Linux w maszynie wirtualnej  – na podstawie opisu instaluje Linux na nośniku zewnętrznym | – samodzielnie i prawidłowo dobiera ustawienia w VirtualBox dla danej dystrybucji Linux  – samodzielnie instaluje Linux na nośniku zewnętrznym, np. pendrivie | – samodzielnie charakteryzuje różne dystrybucje Linux i opisuje ich przeznaczenie |
| Bez kosztów, czyli programy w Linux | – wymienia podstawowe programy instalowane wraz z daną dystrybucją Linux (używaną na zajęciach) i dzieli je na grupy ze względu na ich przeznaczenie  – posługuje się menu w danym GUI w celu odnalezienia wskazanych programów | – samodzielnie odnajduje i uruchamia programy w danej dystrybucji Linux | – na podstawie opisu z podręcznika odnajduje w sieci programy dla Linux i instaluje je za pomocą Menadżera oprogramowania | – samodzielnie odnajduje i instaluje programy z wybranej grupy oprogramowania, np. edytor graficzny, w Linux za pomocą Menadżera oprogramowania | – samodzielnie instaluje programy w Linux bez korzystania z Menadżera oprogramowania |
| Tryb tekstowy jest ważny, czyli poznajemy konsolę Linux | – definiuje rolę konsoli i terminala w systemie Linux  – na podstawie opisu uruchamia Terminal | – wyjaśnia pojęcie powłoki systemowej  – samodzielnie uruchamia Terminal systemu | – na podstawie opisu z podręcznika świadomie ustawia preferencje Terminala  – na podstawie podręcznika dodaje nowe konto użytkownika, usuwa je, nadaje hasło | – samodzielnie wykonuje czynności opisane dla oceny dobrej | – biegle posługuje się Konsolą systemu Linux |
| Okienka nie są potrzebne, czyli używamy konsoli Linux | – wymienia podstawowe czynności, jakie można wykonać za pomocą Konsoli | – na podstawie podręcznika omawia znaczenie uprawnień do plików lub katalogów | – korzystając z Konsoli i na podstawie podręcznika, tworzy, usuwa i kopiuje katalogi  – na podstawie podręcznika wyświetla w konsoli zawartość wskazanego katalogu i określa uprawnienia dla nich danego użytkownika | – samodzielnie wykonuje czynności z oceny dobrej  – nadaje uprawnienia plikom i folderom  – samodzielnie porusza się po drzewie katalogów za pomocą poleceń w Konsoli | – biegle posługuje się Konsolą systemu Linux w czasie wykonywania ćwiczenia na ocenę bardzo dobrą |
| Komputer w kieszeni, czyli jak wykorzystać system Android w nauce i pracy | – wyjaśnia genezę systemu Android  – wymienia przykłady aplikacji pomocnych w nauce | – instaluje wskazane aplikacje w systemie Android  – rozumie wymagania aplikacji i postępuje ostrożnie z ich akceptacją | – instaluje aplikacje wskazane w podręczniku i samodzielnie uczy się ich obsługi  – wyjaśnia, na czym polega proces integracji urządzenia mobilnego pracującego pod kontrolą Androida z komputerem PC | – samodzielnie wyszukuje i instaluje programy służące do nauki i pomagające w życiu codziennym  – ocenia przydatność aplikacji  – konfiguruje ustawienia systemu Android | – samodzielnie ocenia jakość i przydatność aplikacji  – biegle korzysta z systemu Android |
| Był pierwszym z okienkami, czyli macOS i jego właściwości | – wyjaśnia przeznaczenie systemu macOS  – przedstawia genezę systemu macOS | – omawia warunki kompatybilności plików i formatów plików w macOS, Linux i Windows | – korzystając z podręcznika lub zasobów internetowych, wymienia i charakteryzuje aplikacje macOS będące odpowiednikami znanych z Windows i Linux  – wymienia cechy charakterystyczne dla macOS, w tym integrację z innymi urządzeniami Apple | – samodzielnie charakteryzuje macOS  – wymienia dziedziny, w których najlepiej sprawuje się macOS  – wymienia i charakteryzuje aplikacje, które tworzone są dla macOS i jednocześnie innych systemów, w tym Office.  – charakteryzuje złącza komputerów Apple, w tym Thunderbolt | – samodzielnie posługuje się systemem macOS |
| **XIII. Programowanie i rozwiązywanie problemów za pomocą komputera** | | | | | |
| Przypomnij sobie, czyli wracamy do środowiska programistycznego | – definiuje środowisko programistyczne i jego najważniejsze elementy  – na podstawie podręcznika definiuje proces kompilacji kodu programu | – korzystając z podręcznika lub zasobów internetowych, instaluje środowisko programistyczne, np. Eclipse wraz z pakietem MinGW | – samodzielnie instaluje środowisko programistyczne, np. Eclipse wraz z pakietem MinGW  – korzystając z podręcznika lub zasobów internetowych, testuje poprawność działania środowiska programistycznego | – samodzielnie wykonuje czynności opisane dla oceny dobrej  – omawia działanie debuggera i proces kompilacji z uwzględnieniem modułów związanych ze środowiskiem systemu operacyjnego | – płynnie posługuje się środowiskiem programistycznym i konfiguruje je samodzielnie |
| Przypomnij sobie, czyli podstawy języka programowania | – definiuje kod źródłowy  – korzystając z podręcznika lub zasobów internetowych, nazywa i definiuje operatory w języku C++ | – omawia znaczenie zmiennych w programie  – korzystając z podręcznika lub zasobów internetowych, omawia działanie instrukcji warunkowej  – korzystając z podręcznika lub zasobów internetowych, omawia działanie pętli  – omawia znaczenie funkcji w programie komputerowym | – samodzielnie spełnia wymagania oceny dostatecznej  – korzystając z podręcznika lub zasobów internetowych, podaje przykłady wykorzystania zmiennych, funkcji, instrukcji warunkowych i pętli | – samodzielnie spełnia wymagania oceny dobrej  – wyjaśnia różnice pomiędzy typami danych w C++ oraz podaje przykłady ich stosowania | – wykorzystuje informacje z lekcji w prostych programach C++ |
| Czy litery to liczby, czyli kod ASCII i porównanie tekstów | – definiuje kod ASCII | – omawia znaczenie kodu ASCII  – definiuje plagiat i odnosi tę definicję także do rzeczywistości szkolnej, np. do kopiowania prac domowych | – wykorzystuje klawiaturę numeryczną do wprowadzania znaków za pomocą kodów ASCII | – interpretuje przepisy dotyczące plagiatów  – wie, czym jest JSA i jakie ma znaczenie w zwalczaniu kopiowania prac naukowych | Nie przewiduje się oceny celującej w tym temacie |
| Metoda naiwna, czyli szukamy wzorca w tekście | – na podstawie podręcznika opisuje istotę metody naiwnej | – na podstawie ilustracji z podręcznika omawia działanie algorytmu wyszukiwania wzorca w tekście | – samodzielnie omawia działanie algorytmu i metodę naiwną wyszukiwania wzorca w tekście  – wizualizuje metodę naiwną na przykładzie krótkiego tekstu  – na podstawie podręcznika testuje działanie algorytmu | – samodzielnie testuje działanie algorytmu | – samodzielnie tworzy lub modyfikuje algorytm wyszukiwania wzorca metodą naiwną |
| Realizacja algorytmu, czyli program szuka wzorca | – rozumie działanie algorytmu wyszukiwania wzorca metodą naiwną | – na podstawie podręcznika interpretuje i omawia działanie programu wyszukiwania wzorca | – samodzielnie omawia działanie programu z podręcznika i testuje go dla różnych danych | – omawia znaczenie pętli w programie z podręcznika | – samodzielnie układa program różny od programu z podręcznika |
| Przybliżona wartość, czyli komputer oblicza wartość pierwiastka kwadratowego | – rozumie, na czym polega proces szacowania wartości pierwiastka  – wie, do czego służy funkcja pow | – na podstawie podręcznika lub z pomocą nauczyciela szacuje wartość pierwiastka na osi liczbowej.  – biegle używa strumieni cin i cout oraz prawidłowo stosuje funkcję pow | – samodzielnie układa prosty program obliczający wartość pierwiastka, używając funkcji pow – program jest funkcjonalny i pozwala na wprowadzanie danych i wyświetlanie wyniku,  – na podstawie podręcznika omawia znaczenie liczby kolejnych przybliżeń na wynik  – rozumie działanie algorytmu obliczającego wartość pierwiastka kwadratowego i na podstawie tabeli z podręcznika testuje jego działanie | – samodzielnie układa algorytm obliczania pierwiastka kwadratowego  – wykazuje na przykładach wpływ liczby przybliżeń na dokładność wyniku działania programu | – samodzielnie układa program obliczający wartość pierwiastka trzeciego stopnia bez użycia funkcji pow |
| Zbiór Cantora, czyli najprostsze tworzenie fraktali | – wie, czym jest fraktal i podaje przykłady, np. z natury, zachowujące się jak fraktale, np. liście | – przedstawia w formie graficznej istotę zbioru Cantora | – samodzielnie układa algorytm tworzący zbiór Cantora  – testuje algorytm dla różnych stopni zbioru Cantora  – za pomocą podręcznika testuje i rozumie działanie programu kreślącego graficzne przedstawienie zbioru Cantora | – samodzielnie układa i modyfikuje program z wykorzystaniem biblioteki turtle ilustrujący zbiór Cantora | – samodzielnie układa program ilustrujący zbiór Cantora w języku C++ |
| Drzewo binarne, czyli hierarchiczna struktura danych | – na podstawie podręcznika kreśli przykładową strukturę drzewa binarnego | – na podstawie przykładów z podręcznika omawia zastosowanie binarnego drzewa poszukiwań  – zna i omawia istotę regularnego drzewa binarnego | – tworzy BST (binarne drzewo poszukiwań) i omawia mechanizm szukania danego elementu, np. liczby  – tłumaczy działanie programu wstawiającego element do kopca | – samodzielnie układa funkcję wstawiającą element do kopca  – samodzielnie omawia metodę szukania wartości w BST  – podejmuje próbę ułożenia algorytmu i programu wyszukującego element metodą drzewa poszukiwań | – samodzielnie układa algorytm i program wyszukujący element metodą drzewa poszukiwań |
| Trójkątny fraktal, czyli trójkąt Sierpińskiego | – wie, jak powstaje trójkąt i dywan Sierpińskiego i umie to zilustrować, np. na tablicy | – na podstawie przykładów z podręcznika omawia działanie algorytmu tworzącego fraktale Sierpińskiego | – tłumaczy istotę trójkąta i dywanu Sierpińskiego  – omawia znaczenie liczby kroków i możliwość nieograniczonej ich liczby | – samodzielnie układa program z użyciem biblioteki turtle kreślący trójkąt lub dywan Sierpińskiego | – samodzielnie układa program kreślący fraktale Sierpińskiego z możliwością wybrania liczby kolejnych podziałów |
| Krzywa fraktalna, czyli płatek Kocha | – na przykładzie z podręcznika tłumaczy, jak powstaje krzywa Kocha | – tłumaczy, na czym polega efekt samopodobności w płatku Kocha  – omawia algorytm powstawania płatka Kocha | – tłumaczy, na czym polega cecha płatka Kocha mówiąca tym, że brzeg płatka ma nieskończoną długość, a pole wartość skończoną  – na podstawie podręcznika omawia działanie programu używającego biblioteki turtle kreślącego płatek Kocha | – samodzielnie układa program kreślący płatek Kocha, np. z użyciem biblioteki turtle, i testuje jego działanie | – samodzielnie modyfikuje program kreślący płatek Kocha, np. dodaje możliwość wprowadzania liczby poziomów |
| **XIV. Wykorzystanie algorytmów w rozwiązywaniu problemów i programowaniu** | | | | | |
| Jak to zapisać, czyli zamiana systemów liczbowych z ósemkowego na szesnastkowy | – charakteryzuje liczby szesnastkowe i ósemkowe  – wie, jak są zbudowane pozycyjne systemy liczbowe | – na przykładzie z podręcznika omawia działanie algorytmów zamiany postaci liczby dziesiętnej na ósemkową, dziesiętnej na szesnastkową i ósemkowej na szesnastkową  – na przykładzie z podręcznika omawia działanie funkcji zamieniających postaci liczb | – na podstawie opisu i algorytmów z podręcznika układa funkcje zamieniające prezentację liczb z dziesiętnej na ósemkową i szesnastkową oraz z ósemkowej na szesnastkową | – samodzielnie układa programy (funkcje) zamieniające prezentacje liczb  – układa programy wykorzystujące funkcje zamieniające prezentacje liczb | – samodzielnie układa uniwersalne programy oparte o funkcje zamieniające prezentacje liczb na wiele postaci |
| Mniej lub więcej ósemek, czyli dodawanie i odejmowanie w systemie ósemkowym | – wskazuje podobieństwo pomiędzy pisemnym dodawaniem i odejmowaniem liczb dziesiętnych i liczb ósemkowych | – na podstawie podręcznika omawia algorytm dodawania i odejmowania liczb ósemkowych | – na podstawie algorytmu i przykładu z podręcznika układa program dodający lub odejmujący liczby w postaci ósemkowej | – samodzielnie układa funkcjonalny program dodający i odejmujący liczby podane w systemie ósemkowym | – samodzielnie opracowuje program typu kalkulator wykonujący dodawanie lub odejmowanie liczb w systemach ósemkowym, szesnastkowym i dziesiętnym, wyświetlając wyniki w każdym z nich |
| Iloczyn i iloraz oktalny, czyli mnożenie i dzielenie w systemie ósemkowym | – wskazuje podobieństwo pomiędzy pisemnym mnożeniem i dzieleniem liczb dziesiętnych i liczb ósemkowych | – na podstawie podręcznika omawia algorytm mnożenia i dzielenia liczb ósemkowych | – na podstawie algorytmu i przykładu z podręcznika układa program mnożący lub dzielący liczby w postaci ósemkowej | – samodzielnie układa funkcjonalny program mnożący i dzielący liczby podane w systemie ósemkowym | – samodzielnie opracowuje program typu kalkulator wykonujący cztery podstawowe działania w systemach ósemkowym, szesnastkowym i dziesiętnym, wyświetlając wyniki w każdym z nich |
| Nie tylko dwójkowy i dziesiętny, czyli suma i różnica w systemie innym niż dziesiętnym | – wie, jak są zbudowane pozycyjne systemy liczbowe  – umie przedstawić liczbę w dowolnym systemie pozycyjnym, np. siódemkowym | – formułuje ogólną zasadę budowy pozycyjnych systemów liczbowych | – omawia ogólne wzory podane w podręczniku dotyczące dodawania i odejmowania liczb w systemach pozycyjnych | – układa algorytm na podstawie wzorów z podręcznika dodający lub odejmujący liczby w dowolnym liczbowym systemie pozycyjnym | – układa program dodający lub odejmujący liczby w dowolnym liczbowym systemie pozycyjnym |
| Iloczyn i iloraz w systemie innym niż dziesiętnym, czyli wielokrotności w systemach pozycyjnych | – wie, jak są zbudowane pozycyjne systemy liczbowe  – na podstawie wiedzy z poprzedniej lekcji i podręcznika formułuje zasady mnożenia i dzielenia liczb w różnych systemach liczbowych | – umie utworzyć tabliczkę mnożenia w dowolnym systemie pozycyjnym, opierając się na przykładzie z podręcznika | – formułuje algorytm w postaci listy kroków, dzielący i mnożący liczby w dowolnym pozycyjnym systemie liczbowym | – układa program dzielący i mnożący liczby w dowolnym pozycyjnym systemie liczbowym | – modyfikuje wcześniej ułożony program typu kalkulator, dodając do niego inne liczbowe systemy pozycyjne |
| Wyciąganie elementów z listy, czyli znajdowanie najdłuższego spójnego podciągu niemalejącego | – na podstawie podręcznika objaśnia pojęcie najdłuższego podciągu rosnącego | – samodzielnie odnajduje najdłuższy podciąg rosnący w przykładowym ciągu | – samodzielnie bada, czy dany podciąg jest najdłuższym podciągiem rosnącym danego ciągu  – tworzy algorytm sprawdzania, czy dany podciąg jest rosnący  – omawia algorytm wyszukujący najdłuższy podciąg rosnący | – samodzielnie tworzy algorytm wyszukujący najdłuższy podciąg rosnący i omawia istotę jego działania | – układa program wyszukujący w zbiorze najdłuższy podciąg rosnący |
| Elementy występujące kolejno po sobie, czyli znajdowanie najdłuższego spójnego podciągu o największej sumie | – na podstawie podręcznika objaśnia pojęcie najdłuższego spójnego podciągu o największej sumie | – samodzielnie odnajduje najdłuższy spójny podciąg o największej sumie w podanym ciągu  – omawia istotę metody naiwnej  – na podstawie opisu z podręcznika objaśnia istotę sum prefiksowych | – na podstawie podręcznika omawia metodę naiwną lub z zastosowaniem sum prefiksowych wyszukiwania najdłuższego spójnego podciągu o największej sumie | – samodzielnie tworzy algorytm i program wyszukujący najdłuższy spójny podciąg o największej sumie na podstawie metody naiwnej lub stosując sumy prefiksowe | – samodzielnie układa programy dla obu metod |
| Ta sama kolejność, czyli szukanie najdłuższego wspólnego podciągu | – omawia istotę metod opartych na rekurencji  – umie znaleźć w przykładzie najdłuższy wspólny podciąg | – samodzielnie odnajduje najdłuższy wspólny podciąg dla przykładowych ciągów  – na podstawie podręcznika omawia algorytm dynamiczny i oparty na rekurencji rozwiązujący problem odnajdowania najdłuższego wspólnego podciągu | – samodzielnie bada, czy dany podciąg jest najdłuższym wspólnym podciągiem dwóch ciągów  – na podstawie podręcznika omawia zastosowanie metody rekurencyjnej i dynamicznej | – układa program wyszukujący NWP metodą rekurencyjną | – układa programy, stosując obie metody – rekurencyjną i dynamiczną |
| Z której strony, czyli położenia punktów względem prostej | – określa położenie punktu na płaszczyźnie dla przykładowych danych  – umie w układzie współrzędnych wykreślić prostą podaną wzorem | – samodzielnie określa definicje dotyczące punktów i prostych w układzie w spółrzędnych dotyczące ich wzajemnego położenia | – na podstawie podręcznika tłumaczy wzory opisujące wzajemne położenie punktów i prostych na płaszczyźnie | – samodzielnie omawia na przykładach zależności pomiędzy prostymi a punktami na płaszczyźnie | – określa metody badania wzajemnego położenia punktów i prostych w przestrzeni trójwymiarowej |
| Algorytmy badające własności geometryczne, czyli przynależności punktu do odcinka | – określa położenie punktu na płaszczyźnie dla przykładowych danych  – umie wykreślić odcinek w układzie współrzędnych na podstawie podanych danych | – samodzielnie określa definicje dotyczące punktów i odcinków w układzie w spółrzędnych dotyczące ich wzajemnego położenia | – na podstawie podręcznika tłumaczy wzory opisujące wzajemne położenie punktów i odcinków na płaszczyźnie  – na podstawie podręcznika tłumaczy działanie algorytmu i programu badającego przynależność punktu do odcinka na płaszczyźnie | – samodzielnie układa algorytm i program badający przynależność punktu do odcinka na płaszczyźnie | – układa program badający przynależność punktu do odcinka w przestrzeni trójwymiarowej |
| Skrzyżowanie dróg, czyli badanie przecinania się odcinków | – umie wskazać na płaszczyźnie punkt przecięcia odcinków i określić jego współrzędne  – objaśnia różne możliwe przypadki przecięcia się odcinków | – na podstawie podręcznika lub z pomocą nauczyciela objaśnia pojęcie iloczynu wektorowego | – objaśnia warunki, jakie muszą spełniać odcinki, by można było powiedzieć, że się przecinają na płaszczyźnie  – na podstawie podręcznika omawia algorytm badający przecinanie się odcinków | – samodzielnie układa algorytm badający przecinanie się odcinków w postaci pseudokodu | – układa program badający przecinanie się odcinków |
| Wewnątrz czy na zewnątrz, czyli badanie przynależności punktu do trójkąta | – definiuje trójkąt jako figurę geometryczną umieszczoną na płaszczyźnie  – definiuje figurę geometryczną wypukłą | – na podstawie podręcznika omawia metodę polegającą na podziale trójkąta na trzy trójkąty | – na podstawie podręcznika objaśnia działanie algorytmu badającego przynależność punktu do wnętrza trójkąta na płaszczyźnie | – samodzielnie układa algorytm w postaci pseudokodu badający przynależność punku do wnętrza trójkąta | – samodzielnie układa program badający przynależność punku do wnętrza trójkąta |
| Wyznaczanie pola ograniczonego wykresem funkcji, czyli całkowanie numeryczne metodą prostokątów | – na podstawie podręcznika objaśnia definicję całki oznaczonej i nieoznaczonej w kontekście obliczania pola ograniczonego wykresem funkcji w danym przedziale | – na podstawie podręcznika omawia istotę metody prostokątów  – omawia różnice pomiędzy metodą prostokątów z nadmiarem i z niedomiarem | – objaśnia algorytm opisujący metodę obliczania pola obszaru ograniczonego funkcją metodą prostokątów | – samodzielnie układa opisujący metodę obliczania pola obszaru ograniczonego funkcją metodą prostokątów | – samodzielnie układa program obliczający pole obszaru ograniczonego funkcją metodą prostokątów |
| Wyznaczanie pola ograniczonego wykresem funkcji, czyli całkowanie numeryczne metodą trapezów | – na podstawie podręcznika objaśnia definicję całki oznaczonej i nieoznaczonej w kontekście obliczania pola ograniczonego wykresem funkcji w danym przedziale i wskazuje na ilustracji w podręczniku różnice pomiędzy metodą prostokątów a trapezów | – na podstawie podręcznika omawia istotę metody prostokątów  – omawia różnice pomiędzy metodą prostokątów a trapezów | – objaśnia algorytm opisujący metodę obliczania pola obszaru ograniczonego funkcją metodą trapezów | – samodzielnie układa opisujący metodę obliczania pola obszaru ograniczonego funkcją metodą trapezów | – samodzielnie układa program obliczający pole obszaru ograniczonego funkcją metodą trapezów |
| Znajdowanie elementów w zbiorze, czyli wyszukiwanie liniowe | – omawia istotę metody naiwnej w zastosowaniu do wyszukiwania elementów w zbiorze | – na podstawie podręcznika omawia istotę metody wyszukiwania liniowego | – omawia algorytm w postaci pseudokodu wyszukiwania liniowego  – tworzy algorytm dla metody naiwnej | – samodzielnie tworzy algorytm w postaci pseudokodu wyszukiwania liniowego | – układa program dla metody liniowej |
| Szukanie połówek, czyli wyszukiwanie elementów poprzez połowienie | – na podstawie podręcznika omawia istotę metody wyszukiwania przez połowienie i podaje przykłady | – omawia istotę sortowania i jej znaczenie w metodzie szukania elementów metodą binarną (połowienie) | – omawia algorytm w postaci pseudokodu wyszukiwania binarnego | – samodzielnie tworzy algorytm w postaci pseudokodu wyszukiwania binarnego | – układa program dla metody wyszukiwania binarnego |
| Ustawianie kart, czyli sortowanie przez wstawianie | – na podstawie podręcznika lub innych wiarygodnych źródeł omawia istotę sortowania „przez wstawianie” | – na podstawie podręcznika lub innych wiarygodnych źródeł analizuje działanie algorytmu sortowania bąbelkowego w postaci listy kroków i schematu „przez wstawianie”  – analizuje przykład sprawdzający poprawność działania algorytmu | – samodzielnie na przykładzie omawia istotę metody sortowania „przez wstawianie”  – omawia działanie przykładowego algorytmu opartego na metodzie sortowania „przez wstawianie”  – sprawdza działanie algorytmu na przykładach | – samodzielnie układa algorytm sortowania „przez wstawianie” w postaci listy kroków i schematu blokowego  – samodzielnie weryfikuje poprawność działania programu na przykładach | – samodzielnie układa program sortujący metoda „przez wstawianie” w innym języku niż C++, np. Java |
| Reguła falsi, czyli przybliżone rozwiązywanie równań | – omawia zagadnienie określoności i ciągłości funkcji | – na podstawie podręcznika omawia istotę metody falsi w kontekście szukania miejsca zerowego funkcji  – omawia znaczenie twierdzenie Talesa w obliczaniu miejsca zerowego funkcji | – samodzielnie na przykładzie omawia istotę metody falsi do obliczania miejsca zerowego funkcji  – omawia działanie przykładowego algorytmu opartego ona metodzie falsi  – sprawdza działanie algorytmu na przykładach | – samodzielnie układa algorytm znajdowania miejsca zerowego funkcji metoda falsi | – układa program znajdowania miejsca zerowego funkcji metoda falsi |
| Geometria obliczeniowa, czyli sprawdzanie przynależności punktu do wielokąta wypukłego | – omawia różnice między figurami wypukłymi a wklęsłymi | – omawia na przykładzie pojęcie przynależności punktu do figury wypukłej i wklęsłej | – na podstawie podręcznika wyjaśnia działanie algorytmu badania przynależności punktu do wielokąta wypukłego i wklęsłego | – samodzielnie układa i omawia algorytm badania przynależności punktu do wielokąta wypukłego i wklęsłego | – układa program badający przynależność punktu do wielokąta wypukłego i wklęsłego |
| Algorytm wyszukujący, czyli liniowe przeszukiwanie ciągu w poszukiwaniu żądanego elementu z wykorzystaniem wartownika | – omawia zagadnienie i specyfikę algorytmu naiwnego  – na podstawie podręcznika omawia pojęcie przestrzeni poszukiwań | – na podstawie podręcznika omawia istotę wyszukiwania liniowego z wartownikiem | – na podstawie podręcznika wyjaśnia działanie algorytmu wyszukiwania liniowego z wartownikiem | – samodzielnie układa i omawia algorytm wyszukiwania liniowego z wartownikiem | – układa program wyszukiwania liniowego z wartownikiem |
| Podejście zachłanne, czyli problem plecakowy | – wyjaśnia istotę metody zachłannej  – na podstawie podręcznika wyjaśnia, na czym polega problem plecakowy | – samodzielnie omawia zastosowanie metody zachłannej w kontekście problemu plecakowego | – na podstawie podręcznika wyjaśnia działanie algorytmu rozwiązującego problem plecakowy | – samodzielnie układa algorytm rozwiązujący problem plecakowy metodą zachłanną | – układa program rozwiązujący problem plecakowy metodą zachłanną |
| Z miasta A do miasta B, czyli szukanie najkrótszej drogi metodą zachłanną | – omawia istotę metody zachłannej i podejścia iteracyjnego w rozwiązywaniu problemów  – wyjaśnia, na czym polega problem znalezienia najkrótszej drogi | – na podstawie podręcznika wyjaśnia podstawy i założenia algorytmu Dijkstry | – samodzielnie na przykładzie wyjaśnia działanie algorytmu Dijkstry | – samodzielnie układa algorytm Dijkstry i wyjaśnia jego działanie | – układa program rozwiązujący problem poszukiwania najkrótszej drogi pomiędzy punktami |
| Pakowanie plecaka, czyli programowanie dynamiczne | – przypomina założenia problemu pakowania plecaka  – na podstawie podręcznika wyjaśnia założenia programowania dynamicznego  – opisuje zastosowanie zmiennych tablicowych | – wyjaśnia działanie metody zstępującej i wstępującej  – objaśnia zastosowanie metodą programowania dynamicznego w rozwiązywaniu problemu plecakowego | – samodzielnie, na przykładzie wyjaśnia działanie algorytmu rozwiązującego problem plecakowy z zastosowaniem programowania dynamicznego | – samodzielnie układa algorytm rozwiązujący problem plecakowy z zastosowaniem programowania dynamicznego | – układa program rozwiązujący problem plecakowy z zastosowaniem programowania dynamicznego |
| Metoda haszowania, czyli wyszukiwanie wzorca w tekście | – wyjaśnia i definiuje pojęcie haszowania  – omawia znaczenie wyszukiwania wzorca w tekście | – na podstawie podręcznika opisuje metodę haszowania w kontekście wyszukiwania wzorca w tekście  – omawia znaczenie zmiennych tablicowych w programowaniu metodą haszowania | – omawia istotę tablicy haszującej na podstawie funkcji przedstawionej w podręczniku  – na podstawie podręcznika omawia zastosowanie metody haszowania do wyszukiwania wzorca w tekście | – samodzielnie omawia istotę tablicy haszującej i przedstawia przykładową implementację programową  – omawia zastosowanie metody haszowania do wyszukiwania wzorca w tekście | – układa program rozwiązujący problem wyszukiwania wzorca w tekście z zastosowaniem metody haszowania |
| Metoda Monte Carlo, czyli symulacja ruchów Browna | – wie, czym jest modelowanie matematyczne  – omawia istotę metody Monte Carlo | – na podstawie podręcznika omawia przykład stosowania metody Monte Carlo w wyznaczaniu wartości liczby pi  – na podstawie podręcznika omawia zjawisko ruchów Browna | – omawia algorytm i program przedstawiony w podręczniku wyznaczający wartość liczby pi metodą Monte Carlo  – omawia przedstawiony w podręczniku problem symulacji ruchów Browna | – samodzielnie omawia metodę Monte Carlo w kontekście zastosowania do wyznaczania wartości liczby pi  – samodzielnie omawia zastosowanie metody Monte Carlo do symulacji ruchów Browna | – układa program wizualizujący symulacje ruchów Browna |
| **XV. Komputer pomaga w nauce** | | | | | |
| Korzystamy z office.com, czyli jak wykorzystać aplikacje chmury w nauce | – opisuje proces tworzenia darmowego konta OneDrive lub prezentuje własne konto  – wie, do czego służą aplikacje ToDo i Sway | – na podstawie podręcznika korzysta z ToDo w PC lub telefonie  – na podstawie podręcznika tworzy niewielkie prezentacje w Sway | – samodzielnie wykonuje czynności przewidziane dla oceny dostatecznej | – korzysta z ToDo i Sway w sposób twórczy i samodzielny  – tworzy samodzielnie ciekawe prezentacje w Sway | – posługuje się wieloma aplikacjami chmury Microsoft |
| Nie tylko w firmie, czyli wykorzystanie aplikacji komunikacyjnych w nauczaniu | – definiuje i charakteryzuje e-pracę | – na podstawie ilustracji z podręcznika omawia organizację e-pracy w firmie  – wymienia zalety i wady Teams  – omawia cechy firmy prowadzącej pracę zdalną | – charakteryzuje nauczanie zdalne za pośrednictwem różnych narzędzi, w tym komunikatorów, np. Teams | – samodzielnie organizuje spotkania zespołu, np. w celu wspólnego odrabiania pracy domowej | – tworzy biznesplan firmy informatycznej opartej na e-pracy |
| Każdy ma notatki, czyli jak wykorzystać chmurę do wspólnej nauki | – omawia cechy chmury, która mogłaby być wykorzystana do wspólnej nauki do egzaminów lub pracy klasowej | – udostępnia pliki w chmurze | – omawia strukturę chmury wykorzystywanej do wspólnej nauki | – organizuje wsparcie informatyczne w chmurze dla zespołu przygotowującego się do egzaminu | – zarządza zespołami w chmurze |
| Walidacja, czyli sprawdzamy wyniki swojej pracy | – na podstawie podręcznika definiuje pojęcie walidacji | – na podstawie podręcznika omawia zasadę 1-10-100 | – omawia rolę World Wide Web Consortium w tworzeniu standardów | – na podstawie podręcznika umie skorzystać z internetowych narzędzi walidacyjnych | – samodzielnie korzysta z usług walidacyjnych |
| Matura, czyli jak komputery wspomagają przygotowanie do egzaminu | – korzysta z gotowych opracować w programie GeoGebra  – umie znaleźć aplikacje pomagające w przygotowaniach do matury | – na podstawie podręcznika wykonuje wykresy podstawowych funkcji, np. kwadratowej  – odnajduje w sieci przykłady ciekawych projektów, np. kreślenia fraktali | – samodzielnie wykonuje wykresy funkcji w GeoGebrze  – samodzielnie odnajduje, instaluje i korzysta z aplikacji pomagających w przygotowaniach do matury, np. z tablic lub lektur | – samodzielnie przeprowadza doświadczenia z programem GeoGebra | – wykonuje ciekawe symulacje w programie GeoGebra |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **XVI. Algorytmika i programowanie** | | | | | | | | | | |
| Więcej o liczbach, czyli logarytm, sito Eratostenesa i schemat Hornera. | * zna definicję logarytmu * zna ogólną ideę algorytmu sita Eratostenesa (wykreślanie wielokrotności kolejnych liczb pierwszych) * potrafi zapisać naiwny   (“pierwiastkowy”) algorytm sprawdzania  pierwszości | | * zna przykłady sytuacji, w   których używa się logarytmu   * potrafi   prześledzić działanie algorytmu sita Eratostenesa na konkretnych danych | | * potrafi rozpoznać złożoność logarytmiczną w algorytmach * potrafi zapisać w pseudojęzyku algorytm sita Eratostenesa * potrafi zapisać w pseudojęzyku schemat Hornera | | * potrafi   zaimplementować sito Eratostenesa i schemat Hornera w wybranym  języku programowania   * zna złożoność algorytmu sita Eratostenesa i porównuje ją z   “pierwiastkowym”  algorytmem | | – | |
| Rekursja raz jeszcze,  czyli dziel i zwyciężaj. | – potrafi zasymulować (prześledzić działanie)  prostej funkcji rekurencyjnej | | – zna warunki działania rekursji, potrafi  przekształcać proste algorytmy (z jednym | | – prawidłowo analizuje algorytmy rekurencyjne typu „dziel i  zwyciężaj”, zna i umie | | – potrafi samodzielnie zaprojektować  algorytm opierający się na rekursji z dwoma | | – potrafi zanalizować nietypowe wywołania  rekurencyjne (np. na więcej niż dwóch | |
|  |  | | wywołaniem rekurencyjnym) z iteracyjnych na rekurencyjne i  odwrotnie | | stosować warunki działania rekursji w projektowaniu algorytmów | | wywołaniami typu  „dziel i zwyciężaj” | | podzadaniach albo na różniących się od siebie podzadaniach) | |
| Sortowanie przez scalanie, czyli pierwszy efektywny algorytm sortowania. | – zna ogólną ideę algorytmu sortowania przez scalanie (podzielić na dwie połowy, rekursja, scalenie) | | – zna ogólną ideę  scalania, potrafi podać złożoność algorytmu (n log n) | | – potrafi zapisać cały algorytm sortowania przez scalanie w pseudojęzyku | | – potrafi zapisać algorytm sortowania przez scalanie w wybranym języku programowania bez błędów i pilnując przypadków  brzegowych | | – potrafi uargumentowć złożoność algorytmu sortowania przez scalanie | |
| Sortowanie szybkie, czyli jak się sortuje w praktyce. | – zna ogólną ideę algorytmu szybkiego sortowania (przestawić elementy, a potem wywołanie rekurencyjne) | | – wie, czym jest klucz i zna ideę operacji przestawiania, zna  złożoność algorytmu w średnim wypadku | | – potrafi zapisać cały algorytm sortowania przez scalanie w pseudojęzyku | | – potrafi zapisać algorytm sortowania przez scalanie w wybranym języku programowania, rozumie korzyść z losowania klucza | | – potrafi  przedyskutować i uargumentować  złożoność algorytmu przy różnych wyborach klucza  (najlepszym/najgorszy m) | |
| Specjalne elementy w tablicy, czyli  największy, najmniejszy i lider. | * potrafi znaleźć oddzielnie element najmniejszy i największy * wie, czym jest lider i dominanta, potrafi podać przynajmniej jeden algorytm   znajdowania dominanty w zbiorze | | – potrafi prześledzić działanie  (przeanalizować) algorytm jednoczesnego znajdowania najmniejszego i  największego elementu oraz algorytm znajdowania lidera | | – potrafi zapisać w pseudojęzyku algorytm jednoczesnego znajdowania najmniejszego i  największego elementu oraz algorytm znajdowania lidera i podać ich złożoność | | – potrafi  uargumentować liczbę porównań w algorytmie jednoczesnego znajdowania  największego/najmniejs zego elementu oraz poprawność algorytmu znajdowania lidera | | – | |
| Tablice dynamiczne i listy wiązane, czyli kiedy tablica nie wystarcza | – zna ogólną ideę tablicy dynamicznej (tablica + zmienny rozmiar) oraz listy wiązanej | | – opisuje operacje dostępne na tablicy dynamicznej i liście wiązanej, potrafi  przedstawić na rysunku budowę i działanie listy wiązanej | | – potrafi  przeanalizować prosty algorytm, w którym występują operacje na tablicach dynamicznych / listach wiązanych, opisuje rozwiązanie problemu Flawiusza za pomocą listy wiązanej | | – swobodnie analizuje i projektuje algorytmy  zawierające tablice dynamiczne i listy wiązane | | – stosuje tablice dynamiczne i listy wiązane (np. std::vector, std::list, listy w Pythonie) w swoich programach | |
| Stos i kolejka, czyli struktury proste i bardzo użyteczne. | – zna ogólną ideę stosu i kolejki | | – opisuje operacje na stosie i kolejce | | – analizuje proste  algorytmy, w których występują struktury stosu i kolejki, aplikuje strukturę stosu do  rozwiązania problemu ONP | | – swobodnie projektuje analizuje algorytmy, w których występują struktury stosu i kolejki | | – posługuje się typem stos/kolejka w  wybranym języku programowania w swoich programach | |
| Grafy, czyli co mają wspólnego sieć społecznościowa i paryskie metro | – potrafi rozpoznać graf w sytuacjach praktycznych | | – podaje przykłady konkretnych sytuacji modelowanych za pomocą grafu, zna definicję grafu | | – zna implementację grafu przez macierze i listy sąsiedztwa | | * potrafi analizować i projektować algorytmy, w których występują grafy * potrafi porównać implementację   macierzową i listową,  podając ich zalety i wady | | – potrafi samodzielnie zaimplementować graf w wybranym języku programowania | |
| Wyszukiwanie idola, czyli o tym, jak spóźniliśmy się na turniej tenisowy | – wie, czym jest element-idol, potrafi znaleźć element-idol  dowolnym algorytmem (np. pytając o wszystkie  pary) | | – zna ogólną ideę algorytmu wyszukiwania idola („pytaj jeden element do momentu  przegranej”) | | – potrafi zapisać w pseudojęzyku algorytm wyszukiwania idola | | – potrafi uargumentować  złożoność algorytmu wyszukiwania idola | | – | |
| Najkrótsze ścieżki w grafie, czyli jak komputery znajdują drogę | – potrafi podać, czym jest ścieżka w grafie i pokazać jej długość na przykładzie | | – zna ogólną ideę algorytmu wyszukiwania ścieżki  („najpierw elementy o  odległości 1, potem 2, i tak dalej”) | | – potrafi zapisać w pseudojęzyku szkic algorytmu BFS | | – potrafi zapisać w pseudojęzyku pełny algorytm BFS z zastosowaniem kolejki i prosto uzasadnić jego  złożoność | | – implementuje działający algorytm BFS w wybranym  języku programowania | |
| Programowanie strukturalne kontra obiektowe, czyli dane w centrum uwagi | – rozumie ideę programowania obiektowgo („funkcje przypisane do danych”) | | – rozumie pojęcie klasy/obiektu oraz pola i metody | | – wyjaśnia różnicę między programowaniem strukturalnym i obiektowym, rozumie sytuacje, w których programowanie obiektowe jest  koniecznością | | – wyjaśnia bardziej  zaawansowane pojęcia: *dziedziczenie*, *polimorfizm* | | – potrafi stosować w praktyce paradygmat obiektowy (czyli samodzielnie pisze obiektowo programy) | |
| Jak rozwiązywać zadania programistyczne, czyli podstawowe porady na proste kody | – rozumie treść zadań programistycznych i umie przełożyć proste zadania na  implementację w wybranym języku programowania | | – świadomie stosuje zasady czytelnego pisania kodu (wcięcia, nazwy zmiennych etc.) | | * stosuje dobre praktyki programistyczne (unikanie kopiuj-wklej, wydzielanie do podprocedur) * umie przetestować swój program, w tym napisać dla niego proste   dane testowe | | * pisze programy w sposób czytelny i   zrozumiały dla innych   * potrafi swobodnie testować program, wyszukiwać i poprawiać błędy | | – rozwiązuje zadania na poziomie olimpijskim np. na serwerach typu  „online judge” | |
| **XVII. Arkusz kalkulacyjny** | | | | | | | | | | |
| Podstawowe typy danych, czyli co | – uzasadnia wybranie odpowiedniego typu | | – potrafi używać podstawowych funkcji  arkusza i stosować w | | – potrafi samodzielnie używać różne typy  adresowania | | – samodzielnie potrafi dobrać rodzaje  adresowania, tak aby | | – | |
| możemy wpisać w komórkę arkusza | formatowania do podanych danych  – rozróżnia sposoby adresowania | | nich różne typy adresowania  – potrafi dostosować  typ formatowania do podanych danych | |  | | formuła była jak najbardziej uniwersalna i używać  zagnieżdżonych formuł | |  | |
| Trójkąt Sierpińskiego, czyli do czego przydaje się adresowanie  względne i adresowanie bezwzględne oraz wypełnienie serią  danych | – wie, jak wykorzystać podane przez nauczyciela odwzorowania do wygenerowania trójkąta Sierpińskiego | | – potrafi samodzielnie wypełnić komórki podobnymi  wartościami (liczby, daty) | | – potrafi samodzielnie otworzyć pokazane  ćwiczenie, rozumie, jak użyć podanego układu iterowanych rozwiązań | | – potrafi samodzielnie generować inne fraktale niż te pokazane na lekcji (po podaniu mu wzoru na iterowane  przekształcenia) | | – rozumie i potrafi wyjaśnić, na czym polega generowanie fraktali za pomocą układu iterowanych rozwiązań | |
| Jak korzystać z gotowych funkcji, podstawowe funkcje tekstowe, czyli jak łatwo modyfikować dane tekstowe | – zna i potrafi użyć funkcji pokazanych na lekcji | | – potrafi przeczytać opis nowej funkcji i użyć jej w zadaniu | | – potrafi samodzielnie używać podanych funkcji i używać ich w formułach | | – potrafi samodzielnie używać podanych funkcji i używać ich w formułach  zagnieżdżonych | | – potrafi samodzielnie używać podanych funkcji i używać ich w formułach  zagnieżdżonych, tak  aby te formuły były uniwersalne | |
| Jak korzystać z gotowych funkcji, podstawowe funkcje dotyczące daty i czasu, czyli jak łatwo  pracować z datą. | – zna i potrafi użyć funkcji pokazanych na lekcji | | – potrafi przeczytać opis nowej funkcji i użyć jej w zadaniu | | – potrafi samodzielnie używać podanych funkcji i używać ich w formułach | | – potrafi samodzielnie używać podanych funkcji i używać ich w formułach  zagnieżdżonych | | – potrafi samodzielnie używać podanych funkcji i używać ich w formułach  zagnieżdżonych, tak  aby te formuły były uniwersalne | |
| Jak szukać w Excelu, czyli gdzie to jest | – potrafi podać, jakie funkcje w arkuszu pozwalają wyszukać, potrafi używać funkcji WYSZUKAJ na  prostych przykładach | | – odtwarza przykłady użycia funkcji WYSZUKAJ podane w podręczniku | | – potrafi użyć funkcji WYSZUKAJ.PIONOW  O dla przykładów podanych w podręczniku | | – potrafi samodzielnie stosować funkcję WYSZUKAJ dla dowolnych przykładów podanych przez nauczyciela i potrafi sam wskazać takie  przykłady | | – potrafi samodzielnie stosować funkcję WYSZUKAJ.PIONOW O dla dowolnych  przykładów i potrafi sam wskazać takie przykłady | |
| Co ukrywa numer PESEL, czyli do czego przydaje się adres mieszany | – potrafi z podanego numeru PESEL odczytać datę urodzin i płeć | | – potrafi samodzielnie napisać formułę, która policzy z podanego numeru PESEL datę urodzin | | – samodzielnie potrafi napisać formułę, która wyznaczy płeć z numeru PESEL | | – potrafi napisać formułę, która sprawdzi czy numer PESEL jest prawidłowy | | – samodzielnie potrafi napisać uniwersalne formuły wyznaczające datę urodzenia, płeć i sprawdzające  poprawność dla wielu danych | |
| Tabela przestawna, czyli wygodne grupowanie danych | – wie, kiedy stosować tabele przestawne, potrafi odczytywać informacje z tabeli przestawnej | | – potrafi odtworzyć przykłady użycia tabeli przestawnej pokazane na lekcji | | – samodzielnie potrafi utworzyć tabelę  przestawną i policzyć, sumę, średnią, minimum. maksimum  dla wyznaczonych grup | | – potrafi samodzielnie utworzyć tabelę  przestawną, zastosować w niej filtr i posortować dane według podanego  klucza | | – potrafi samodzielnie utworzyć tabele  przestawną i dla grup tworzyć podgrupy oraz użyć funkcji liczących  dla grup i podgrup | |
| Wykresy, czyli jak atrakcyjnie  przedstawiać dane | * potrafi odczytywać dane z wykresu * wie, jakie są podstawowe typy wykresu * potrafi dobrać odpowiedni rodzaj   wykresu do podanych danych | | – potrafi odtworzyć przykłady użycia tabeli przestawnej pokazane na lekcji | | – samodzielnie potrafi utworzyć wykres i sformatować go według podanych wytycznych | | * potrafi samodzielnie dobrać wykres do podanych danych i sformatować go tak,   aby był czytelny   * potrafi przedstawić kilka serii danych na wykresie | | – potrafi utworzyć wykres z tabeli przestawnej i sformatować go, aby dane były czytelne | |
| Pobieranie danych z pliku, czyli z notatnika do Excela. | – potrafi pobrać dane z pliku tekstowego  zawierające tekst i liczby całkowite | | – potrafi pobrać dane z pliku zawierające długie liczby i numer  pesel | | – potrafi pobrać dane zawierające liczby rzeczywiste | | – potrafi pobrać dane z pliku zawierające datę i czas | | – potrafi pobrać dane z dowolnego pliku  zawierającego tekst ze znakami polskimi | |
| Więcej o liczbach, czyli logarytm, sito Eratostenesa i schemat Hornera. | | * zna definicję logarytmu * zna ogólną ideę algorytmu sita Eratostenesa (wykreślanie wielokrotności kolejnych liczb pierwszych) * potrafi zapisać naiwny   (“pierwiastkowy”) algorytm sprawdzania  pierwszości | | * zna przykłady sytuacji, w   których używa się logarytmu   * potrafi   prześledzić działanie algorytmu sita Eratostenesa na konkretnych danych | | * potrafi rozpoznać złożoność logarytmiczną w algorytmach * potrafi zapisać w pseudojęzyku algorytm sita Eratostenesa * potrafi zapisać w pseudojęzyku schemat Hornera | | * potrafi   zaimplementować sito Eratostenesa i schemat Hornera w wybranym  języku programowania   * zna złożoność algorytmu sita Eratostenesa i porównuje ją z   “pierwiastkowym”  algorytmem | | – | |
| Rekursja raz jeszcze,  czyli dziel i zwyciężaj. | | – potrafi zasymulować (prześledzić działanie)  prostej funkcji rekurencyjnej | | – zna warunki działania rekursji, potrafi  przekształcać proste algorytmy (z jednym | | – prawidłowo analizuje algorytmy rekurencyjne typu „dziel i  zwyciężaj”, zna i umie | | – potrafi samodzielnie zaprojektować  algorytm opierający się na rekursji z dwoma | | – potrafi zanalizować nietypowe wywołania  rekurencyjne (np. na więcej niż dwóch | |
|  | |  | | wywołaniem rekurencyjnym) z iteracyjnych na rekurencyjne i  odwrotnie | | stosować warunki działania rekursji w projektowaniu algorytmów | | wywołaniami typu  „dziel i zwyciężaj” | | podzadaniach albo na różniących się od siebie podzadaniach) | |
| Sortowanie przez scalanie, czyli pierwszy efektywny algorytm sortowania. | | – zna ogólną ideę algorytmu sortowania przez scalanie (podzielić na dwie połowy, rekursja, scalenie) | | – zna ogólną ideę  scalania, potrafi podać złożoność algorytmu (n log n) | | – potrafi zapisać cały algorytm sortowania przez scalanie w pseudojęzyku | | – potrafi zapisać algorytm sortowania przez scalanie w wybranym języku programowania bez błędów i pilnując przypadków  brzegowych | | – potrafi uargumentowć złożoność algorytmu sortowania przez scalanie | |
| Sortowanie szybkie, czyli jak się sortuje w praktyce. | | – zna ogólną ideę algorytmu szybkiego sortowania (przestawić elementy, a potem wywołanie rekurencyjne) | | – wie, czym jest klucz i zna ideę operacji przestawiania, zna  złożoność algorytmu w średnim wypadku | | – potrafi zapisać cały algorytm sortowania przez scalanie w pseudojęzyku | | – potrafi zapisać algorytm sortowania przez scalanie w wybranym języku programowania, rozumie korzyść z losowania klucza | | – potrafi  przedyskutować i uargumentować  złożoność algorytmu przy różnych wyborach klucza  (najlepszym/najgorszy m) | |
| Specjalne elementy w tablicy, czyli  największy, najmniejszy i lider. | | * potrafi znaleźć oddzielnie element najmniejszy i największy * wie, czym jest lider i dominanta, potrafi podać przynajmniej jeden algorytm   znajdowania dominanty w zbiorze | | – potrafi prześledzić działanie  (przeanalizować) algorytm jednoczesnego znajdowania najmniejszego i  największego elementu oraz algorytm znajdowania lidera | | – potrafi zapisać w pseudojęzyku algorytm jednoczesnego znajdowania najmniejszego i  największego elementu oraz algorytm znajdowania lidera i podać ich złożoność | | – potrafi  uargumentować liczbę porównań w algorytmie jednoczesnego znajdowania  największego/najmniejs zego elementu oraz poprawność algorytmu znajdowania lidera | | – | |
| Tablice dynamiczne i listy wiązane, czyli kiedy tablica nie wystarcza | | – zna ogólną ideę tablicy dynamicznej (tablica + zmienny rozmiar) oraz listy wiązanej | | – opisuje operacje dostępne na tablicy dynamicznej i liście wiązanej, potrafi  przedstawić na rysunku budowę i działanie listy wiązanej | | – potrafi  przeanalizować prosty algorytm, w którym występują operacje na tablicach dynamicznych / listach wiązanych, opisuje rozwiązanie problemu Flawiusza za pomocą listy wiązanej | | – swobodnie analizuje i projektuje algorytmy  zawierające tablice dynamiczne i listy wiązane | | – stosuje tablice dynamiczne i listy wiązane (np. std::vector, std::list, listy w Pythonie) w swoich programach | |
| Stos i kolejka, czyli struktury proste i bardzo użyteczne. | | – zna ogólną ideę stosu i kolejki | | – opisuje operacje na stosie i kolejce | | – analizuje proste  algorytmy, w których występują struktury stosu i kolejki, aplikuje strukturę stosu do  rozwiązania problemu ONP | | – swobodnie projektuje analizuje algorytmy, w których występują struktury stosu i kolejki | | – posługuje się typem stos/kolejka w  wybranym języku programowania w swoich programach | |
| Grafy, czyli co mają wspólnego sieć społecznościowa i paryskie metro | | – potrafi rozpoznać graf w sytuacjach praktycznych | | – podaje przykłady konkretnych sytuacji modelowanych za pomocą grafu, zna definicję grafu | | – zna implementację grafu przez macierze i listy sąsiedztwa | | * potrafi analizować i projektować algorytmy, w których występują grafy * potrafi porównać implementację   macierzową i listową,  podając ich zalety i wady | | – potrafi samodzielnie zaimplementować graf w wybranym języku programowania | |
| Wyszukiwanie idola, czyli o tym, jak spóźniliśmy się na turniej tenisowy | | – wie, czym jest element-idol, potrafi znaleźć element-idol  dowolnym algorytmem (np. pytając o wszystkie  pary) | | – zna ogólną ideę algorytmu wyszukiwania idola („pytaj jeden element do momentu  przegranej”) | | – potrafi zapisać w pseudojęzyku algorytm wyszukiwania idola | | – potrafi uargumentować  złożoność algorytmu wyszukiwania idola | | – | |
| Najkrótsze ścieżki w grafie, czyli jak komputery znajdują drogę | | – potrafi podać, czym jest ścieżka w grafie i pokazać jej długość na przykładzie | | – zna ogólną ideę algorytmu wyszukiwania ścieżki  („najpierw elementy o  odległości 1, potem 2, i tak dalej”) | | – potrafi zapisać w pseudojęzyku szkic algorytmu BFS | | – potrafi zapisać w pseudojęzyku pełny algorytm BFS z zastosowaniem kolejki i prosto uzasadnić jego  złożoność | | – implementuje działający algorytm BFS w wybranym  języku programowania | |
| Programowanie strukturalne kontra obiektowe, czyli dane w centrum uwagi | | – rozumie ideę programowania obiektowgo („funkcje przypisane do danych”) | | – rozumie pojęcie klasy/obiektu oraz pola i metody | | – wyjaśnia różnicę między programowaniem strukturalnym i obiektowym, rozumie sytuacje, w których programowanie obiektowe jest  koniecznością | | – wyjaśnia bardziej  zaawansowane pojęcia: *dziedziczenie*, *polimorfizm* | | – potrafi stosować w praktyce paradygmat obiektowy (czyli samodzielnie pisze obiektowo programy) | |
| Jak rozwiązywać zadania programistyczne, czyli podstawowe porady na proste kody | | – rozumie treść zadań programistycznych i umie przełożyć proste zadania na  implementację w wybranym języku programowania | | – świadomie stosuje zasady czytelnego pisania kodu (wcięcia, nazwy zmiennych etc.) | | * stosuje dobre praktyki programistyczne (unikanie kopiuj-wklej, wydzielanie do podprocedur) * umie przetestować swój program, w tym napisać dla niego proste   dane testowe | | * pisze programy w sposób czytelny i   zrozumiały dla innych   * potrafi swobodnie testować program, wyszukiwać i poprawiać błędy | | – rozwiązuje zadania na poziomie olimpijskim np. na serwerach typu  „online judge” | |
| **XVIII. Arkusz kalkulacyjny** | | | | | | | | | | | |
| Podstawowe typy danych, czyli co | | – uzasadnia wybranie odpowiedniego typu | | – potrafi używać podstawowych funkcji  arkusza i stosować w | | – potrafi samodzielnie używać różne typy  adresowania | | – samodzielnie potrafi dobrać rodzaje  adresowania, tak aby | | – | |
| możemy wpisać w komórkę arkusza | | formatowania do podanych danych  – rozróżnia sposoby adresowania | | nich różne typy adresowania  – potrafi dostosować  typ formatowania do podanych danych | |  | | formuła była jak najbardziej uniwersalna i używać  zagnieżdżonych formuł | |  | |
| Trójkąt Sierpińskiego, czyli do czego przydaje się adresowanie  względne i adresowanie bezwzględne oraz wypełnienie serią  danych | | – wie, jak wykorzystać podane przez nauczyciela odwzorowania do wygenerowania trójkąta Sierpińskiego | | – potrafi samodzielnie wypełnić komórki podobnymi  wartościami (liczby, daty) | | – potrafi samodzielnie otworzyć pokazane  ćwiczenie, rozumie, jak użyć podanego układu iterowanych rozwiązań | | – potrafi samodzielnie generować inne fraktale niż te pokazane na lekcji (po podaniu mu wzoru na iterowane  przekształcenia) | | – rozumie i potrafi wyjaśnić, na czym polega generowanie fraktali za pomocą układu iterowanych rozwiązań | |
| Jak korzystać z gotowych funkcji, podstawowe funkcje tekstowe, czyli jak łatwo modyfikować dane tekstowe | | – zna i potrafi użyć funkcji pokazanych na lekcji | | – potrafi przeczytać opis nowej funkcji i użyć jej w zadaniu | | – potrafi samodzielnie używać podanych funkcji i używać ich w formułach | | – potrafi samodzielnie używać podanych funkcji i używać ich w formułach  zagnieżdżonych | | – potrafi samodzielnie używać podanych funkcji i używać ich w formułach  zagnieżdżonych, tak  aby te formuły były uniwersalne | |
| Jak korzystać z gotowych funkcji, podstawowe funkcje dotyczące daty i czasu, czyli jak łatwo  pracować z datą. | | – zna i potrafi użyć funkcji pokazanych na lekcji | | – potrafi przeczytać opis nowej funkcji i użyć jej w zadaniu | | – potrafi samodzielnie używać podanych funkcji i używać ich w formułach | | – potrafi samodzielnie używać podanych funkcji i używać ich w formułach  zagnieżdżonych | | – potrafi samodzielnie używać podanych funkcji i używać ich w formułach  zagnieżdżonych, tak  aby te formuły były uniwersalne | |
| Jak szukać w Excelu, czyli gdzie to jest | | – potrafi podać, jakie funkcje w arkuszu pozwalają wyszukać, potrafi używać funkcji WYSZUKAJ na  prostych przykładach | | – odtwarza przykłady użycia funkcji WYSZUKAJ podane w podręczniku | | – potrafi użyć funkcji WYSZUKAJ.PIONOW  O dla przykładów podanych w podręczniku | | – potrafi samodzielnie stosować funkcję WYSZUKAJ dla dowolnych przykładów podanych przez nauczyciela i potrafi sam wskazać takie  przykłady | | – potrafi samodzielnie stosować funkcję WYSZUKAJ.PIONOW O dla dowolnych  przykładów i potrafi sam wskazać takie przykłady | |
| Co ukrywa numer PESEL, czyli do czego przydaje się adres mieszany | | – potrafi z podanego numeru PESEL odczytać datę urodzin i płeć | | – potrafi samodzielnie napisać formułę, która policzy z podanego numeru PESEL datę urodzin | | – samodzielnie potrafi napisać formułę, która wyznaczy płeć z numeru PESEL | | – potrafi napisać formułę, która sprawdzi czy numer PESEL jest prawidłowy | | – samodzielnie potrafi napisać uniwersalne formuły wyznaczające datę urodzenia, płeć i sprawdzające  poprawność dla wielu danych | |
| Tabela przestawna, czyli wygodne grupowanie danych | | – wie, kiedy stosować tabele przestawne, potrafi odczytywać informacje z tabeli przestawnej | | – potrafi odtworzyć przykłady użycia tabeli przestawnej pokazane na lekcji | | – samodzielnie potrafi utworzyć tabelę  przestawną i policzyć, sumę, średnią, minimum. maksimum  dla wyznaczonych grup | | – potrafi samodzielnie utworzyć tabelę  przestawną, zastosować w niej filtr i posortować dane według podanego  klucza | | – potrafi samodzielnie utworzyć tabele  przestawną i dla grup tworzyć podgrupy oraz użyć funkcji liczących  dla grup i podgrup | |
| Wykresy, czyli jak atrakcyjnie  przedstawiać dane | | * potrafi odczytywać dane z wykresu * wie, jakie są podstawowe typy wykresu * potrafi dobrać odpowiedni rodzaj   wykresu do podanych danych | | – potrafi odtworzyć przykłady użycia tabeli przestawnej pokazane na lekcji | | – samodzielnie potrafi utworzyć wykres i sformatować go według podanych wytycznych | | * potrafi samodzielnie dobrać wykres do podanych danych i sformatować go tak,   aby był czytelny   * potrafi przedstawić kilka serii danych na wykresie | | – potrafi utworzyć wykres z tabeli przestawnej i sformatować go, aby dane były czytelne | |
| Pobieranie danych z pliku, czyli z notatnika do Excela. | | – potrafi pobrać dane z pliku tekstowego  zawierające tekst i liczby całkowite | | – potrafi pobrać dane z pliku zawierające długie liczby i numer  pesel | | – potrafi pobrać dane zawierające liczby rzeczywiste | | – potrafi pobrać dane z pliku zawierające datę i czas | | – potrafi pobrać dane z dowolnego pliku  zawierającego tekst ze znakami polskimi | |