Zakres treści i wymagania edukacyjne z informatyki dla klasy drugiej

Lp.	Temat	Liczba godzin	Osiągnięcia uczniów	
			Wymagania podstawowe. Uczeń:	**Wymagania ponadpodstawowe. Uczeń:**
1	Jak pobierać dane do arkusza kalkulacyjnego	3	– wyjaśnia, dlaczego warto stosować narzędzia wymiany danych – wymienia podstawowe zastosowania arkusza kalkulacyjnego – wyjaśnia podstawowe pojęcia związane z arkuszem kalkulacyjnym: skoroszyt, arkusz, adres komórki, formuła, funkcja, zakres adresów – pobierana dane z różnych źródeł i przetwarza je – modyfikuje dane w arkuszu – wykorzystuje adresy komórek w formułach obliczeniowych – wyjaśnia różnice między formułami i funkcjami – korzysta z wbudowanych funkcji arkusza kalkulacyjnego – stosuje różne sposoby zaznaczania zakresów komórek – kopiuje dane z komórek i wkleja je na różne sposoby, również między arkuszami – kopiuje formuły – stosuje funkcje: SUMA, ŚREDNIA, MAX, MIN, DŁ, JEŻELI – przedstawia dane w postaci wykresów	– wyszukuje samodzielnie w internecie dane potrzebne do realizacji określonych zadań – importuje do arkusza dane z różnych źródeł, w tym ze stron WWW – buduje złożone formuły pozwalające wykonywać obliczenia, rozwiązujące określone problemy – poprawnie stosuje adresowanie względne, bezwzględne i mieszane – modyfikuje dane podczas ich importowania – wyszukuje w internecie informacje na temat nowych funkcji i stosuje je w zadaniach – dobiera typ wykresu do rodzaju danych – interpretuje otrzymane wyniki zgodnie z ustalonymi założeniami
2	Wyciągamy wiedzę z danych	4	– omawia różnicę między filtrowaniem i sortowaniem danych – filtruje i sortuje dane – tworzy tabele i stosuje w nich sortowanie i filtrowanie danych – stosuje formuły arkusza kalkulacyjnego do losowego generowania zbiorów danych – stosuje funkcję INDEKS do zwracania wartości określonych komórek – opisuje możliwości tabel przestawnych	– generuje zestawy danych za pomocą narzędzi online – modyfikuje style tabel przestawnych – buduje tabele przestawne dla dużych zbiorów danych – tworzy fragmentatory i korzysta z osi czasu – interpretuje wyniki tabel i wykresów przestawnych – stosuje tabele przestawne do rozwiązywania złożonych zadań, w których wykorzystano duże zbiory danych

			– tworzy tabele przestawne – filtruje dane w tabeli przestawnej – aktualizuje tabelę przestawną po modyfikacji danych źródłowych – stosuje gotowe style tabel przestawnych – podsumowuje dane w tabeli przestawnej na różne sposoby – stosuje różne sposoby wyświetlania wartości w tabeli przestawnej – grupuje i rozgrupowuje daty w tabelach przestawnych – tworzy wykresy przestawne	
3	Korespondencja seryjna	2	– tworzy dokument główny korespondencji seryjnej – umieszcza pola korespondencji seryjnej w tworzonych dokumentach – tworzy bazę adresatów – stosuje reguły warunkowe do personalizacji listów seryjnych – poprawnie scala dokumenty seryjne	– zarządza danymi adresatów korespondencji seryjnej w arkuszu kalkulacyjnym – tworzy zestawy dokumentów seryjnych (listy, etykiety, koperty) – drukuje listy seryjne – wysyła korespondencję seryjną za pomocą poczty elektronicznej
4	Relacyjne bazy danych	3	– wyjaśnia podstawowe pojęcia związane z bazami danych: tabela, atrybut, rekord, pole, klucz główny, klucz obcy, relacja – wymienia różne zastosowania baz danych – projektuje nieduże bazy danych – zarządza danymi w bazie danych w programie MS Access – modyfikuje dane zawarte w bazie danych – tworzy tabele i definiuje relacje między nimi – tworzy raporty z użyciem kreatora	– tworzy kwerendy, formularze oraz raporty w programie MS Access – dostosowuje raport do aktualnych potrzeb – drukuje i eksportuje raporty do pliku
P1	Współdziałanie aplikacji – projekt zespołowy	3	– współpracuje w grupie, korzystając z narzędzi online – korzysta z programów graficznych podczas pracy nad zadaniem projektowym – stosuje funkcje arkusza kalkulacyjnego do przetwarzania danych – testuje rozwiązania wypracowane w grupie	– prezentuje efekty pracy grupowej na forum klasy – przyjmuje rolę lidera odpowiedzialnego za zespół i projekt

			– korzysta z zasobów internetowych, wyszukując potrzebne informacje – stosuje zasady netykiety i korzysta z niej w komunikacji zdalnej	
10	Od problemu do programu	4	– wyjaśnia pojęcie algorytmu – podaje przykłady algorytmów niekomputerowych – wymienia cechy poprawnego algorytmu – wyjaśnia na przykładzie pojęcie specyfikacji problemu – tworzy algorytm wyznaczania pierwiastka kwadratowego – zapisuje algorytm Herona w postaci listy kroków – wyjaśnia pojęcia związane z algorytmiką i programowaniem: schemat blokowy, lista kroków, kod źródłowy, kod wynikowy, kompilator, interpreter, słowa kluczowe, funkcje, plik wykonywalny – zapisuje algorytm w postaci kodu źródłowego – kompiluje zapisany kod źródłowy – znajduje i poprawia błędy w kodzie źródłowym programu – wyjaśnia pojęcie zmiennej i typu zmiennej – wymienia zasady tworzenia kodu źródłowego w wybranym języku programowania – stosuje podstawowe konstrukcje wybranego języka programowania: instrukcje wejścia i wyjścia, operatory arytmetyczne i logiczne oraz instrukcję warunkową – tworzy program sprawdzający warunek trójkąta	– tworzy samodzielnie programy, wykorzystując poznane instrukcje wybranego języka programowania – stosuje w swoich programach zagnieżdżone instrukcje warunkowe – pisze programy rozwiązujące zadania matematyczne i fizyczne oraz problemy z napisami
11	Wyszukujemy i sumujemy	2	– wyjaśnia, na czym polega iteracyjne rozwiązywanie problemu – stosuje w swoich programach podstawowe rodzaje pętli: while i for – opisuje zasady użycia pętli w programach – analizuje w tabeli działanie algorytmu krok po kroku – opisuje, jak komputer porównuje dwie wartości	– stosuje różne rodzaje pętli, dostosowując wybór do rozwiązywanego problemu – stosuje w kodzie operatory inkrementacji i dekrementacji – używa w algorytmach rozwiązania z wartownikiem, czyli specjalną wartością wskazującą koniec ciągu wczytywanych do programu wartości – analizuje kod źródłowy i poprawia ewentualne błędy

			– tworzy program wyszukiwania największej liczby z zadanego ciągu liczb	
12	Binarny system liczbowy	3	– zapisuje liczby w binarnym systemie liczbowym oraz w systemie szesnastkowym – wyjaśnia pojęcia związane z systemami liczbowymi: system pozycyjny, podstawa systemu liczbowego – omawia algorytm konwersji liczb między systemami dziesiętnym i binarnym – zapisuje algorytm konwersji między systemami liczbowymi w postaci programu komputerowego	– stosuje operację dzielenia całkowitego w rozwiązywaniu problemów – tworzy algorytmy konwersji między różnymi systemami liczbowymi – programuje algorytmy konwersji między różnymi systemami liczbowymi, stosując strukturę tablicy lub listy
13	Czy ta liczba jest pierwsza?	3	– definiuje liczby złożone i liczby pierwsze – podaje przykłady użycia liczb pierwszych – omawia algorytmy sprawdzające podzielność liczb – bada podzielność wybranych liczb, programując poznane algorytmy w wybranym języku – grupuje instrukcje w funkcje i wyjaśnia cel stosowania funkcji	– tworzy samodzielnie programy dla poznanych algorytmów – wyjaśnia pojęcia liczb doskonałych, bliźniaczych, zaprzyjaźnionych – pisze programy wykorzystujące poznane rodzaje liczb pierwszych – wyjaśnia praktyczne znaczenie liczb pierwszych w informatyce
14	Algorytm Euklidesa i działania na ułamkach	1	– wyjaśnia pojęcia: NWD, NWW – podaje przykłady zastosowania algorytmu Euklidesa – zapisuje algorytm Euklidesa w postaci listy kroków – tworzy program pozwalający na dodawanie ułamków – stosuje odpowiednie konstrukcje wybranego języka programowania do implementacji omawianych zagadnień (w tym: funkcję, która nie zwraca wartości)	– tworzy programy realizujące działania na ułamkach – opisuje algorytm Euklidesa i tworzy realizujący go program w wybranym języku programowania – opisuje różnicę w sprawności dwóch wersji algorytmu Euklidesa: z odejmowaniem i z dzieleniem – poznaje inne zastosowania algorytmu Euklidesa, wykorzystując informacje zawarte w internecie lub innych źródłach