

WYMAGANIA Z INFORMATYKI – KLASA VIII

opracowany na podstawie podręcznika:

Grażyna Koba, *Teraz bajty. Informatyka dla szkoły podstawowej. Klasa VIII. Nowe wydanie,*

Lp.	Temat lekcji	Wiedza i umiejętności	
		podstawowe	rozszerzające
		Uczeń:	Uczeń:
1.	Więcej o opracowywaniu tekstu – tabulatory i spacje nierozdzielające	stosuje różne typy tabulatorów, potrafi zmienić ich ustawienia w całym tekście; wie, kiedy wskazane jest zastosowanie spacji nierozdzielającej i odpowiednio ją stosuje	rozumie różnice w zastosowaniu wcięć i tabulatorów; dobiera, ustawia i zmienia tabulatory – odpowiednio do zawartości dokumentu; stosuje spację nierozdzielającą tam, gdzie jest to wskazane
2.	Więcej o opracowywaniu tekstu – listy numerowane i tabele	stosuje automatyczną numerację i wypunktowanie w prostych tekstach; wstawia do tekstu tabelę i wprowadza do niej dane; drukuje dokumenty tekstowe; korzysta z Pomocy do programu w celu znalezienia potrzebnych opcji	samodzielnie dobiera parametry drukowania (m.in. drukuje wybraną stronę dokumentu, tylko strony parzyste); rozumie i stosuje zasady automatycznego numerowania

3.	Praca z dokumentem wielostronicowym – stopka i nagłówek, wyszukiwanie słów i znaków	wstawia informacje do nagłówka i stopki w wielostronicowym dokumencie; wyszukuje słowa i znaki w tekście; korzysta z Pomocy do programu w celu znalezienia potrzebnych opcji	samodzielnie wyszukuje opcje menu potrzebne do rozwiązania dowolnego problemu związanego z redagowaniem i formatowaniem tekstu; przygotowuje profesjonalny wielostronicowy dokument, stosując poznane zasady redagowania tekstów
4.	Praca z dokumentem wielostronicowym – przypisy, kolumny	potrafi podzielić tekst na kolumny; tworzy przypisy; korzysta z Pomocy do programu w celu znalezienia potrzebnych opcji	przygotowuje profesjonalny wielostronicowy dokument
5.	Tworzenie e-gazetki – projekt	wie, czym jest chmura, jakie są możliwości pracy w chmurze; stosuje je w pracy zespołowej przy tworzeniu projektów; omawia etapy przygotowania projektu grupowego; współpracuje w grupie, wykonując polecenia koordynatora grupy	potrafi pełnić funkcję koordynatora grupy; wykorzystuje możliwości pracy w chmurze
6.	Pisanie poleceń w języku Python w trybie interaktywnym	zna pojęcia <i>interpretacja i kompilacja</i> ; wie, na czym polega tryb interaktywny; porównuje wykonywanie poleceń w trybie interaktywnym w środowisku Scratch i w środowisku Python; pisze proste polecenia w trybie interaktywnym (wyświetlanie napisów, wykonywanie obliczeń na konkretnych wartościach)	wyjaśnia na przykładach (w języku C++ i Python), czym się różni język interpretowany od kompilowanego; samodzielnie wykonuje ćwiczenia z podręcznika, pisząc polecenia w trybie interaktywnym; porównuje pracę w trybie interaktywnym w języku Logo z pisananiem poleceń w takim trybie w języku Python
7.	Pisanie programu w języku Python w trybie skryptowym, stosowanie zmiennych i wykonywanie obliczeń	zna etapy tworzenia programu w języku Python i ogólną budowę programu; pisze prosty program wyświetlający napis na ekranie; deklaruje zmienne typu liczbowego (całkowite, rzeczywiste); wie, jak nadać wartość zmiennej; pisze proste programy (częściowo z pomocą nauczyciela), w których są wykonywane obliczenia z użyciem zmiennych	rozumie, czym jest zmienna w programie, m.in. wie, że przypisanie zmiennej o tej samej nazwie innej wartości zastępuje poprzednią wartość; pisze samodzielnie programy z wykorzystaniem zmiennych; samodzielnie zapoznaje się z działaniem instrukcji warunkowej

8.	Stosowanie instrukcji warunkowej do realizacji algorytmów z warunkami w języku Python	<p>wie, jak działa instrukcja warunkowa <code>if</code> w języku Python;</p> <p>wykazuje podobieństwo w działaniu instrukcji warunkowej w języku Python do działania tej instrukcji w środowisku programowania Scratch;</p> <p>analizuje przykładową sytuację warunkową i wyjaśnia konieczność zastosowania instrukcji warunkowej;</p> <p>potrafi napisać program realizujący prosty algorytm z warunkami</p>	<p>wykazuje podobieństwo w działaniu instrukcji warunkowej w języku Python do działania tej instrukcji w środowiskach programowania Scratch i Báltie;</p> <p>pisze trudniejsze programy wymagające zastosowania instrukcji warunkowej</p>
9.	Stosowanie instrukcji iteracyjnej do realizacji algorytmów iteracyjnych w języku Python	<p>wie, jak działa instrukcja iteracyjna <code>for</code> w języku Python;</p> <p>wie, do czego służy funkcja <code>range()</code>;</p> <p>korzystając z podręcznika, zapisuje w postaci programu proste algorytmy iteracyjne w języku Python i w języku Scratch;</p> <p>porównaj program z realizacją tego samego algorytmu w języku Python. Jakie znajdujesz podobieństwa, a jakie różnice?</p>	<p>omawia przykłady stosowania instrukcji <code>for</code> w języku Python z zapisaną w różnych wersjach <i>listą wartości</i> po słowie <code>in</code> i z wykorzystaniem funkcji <code>range()</code>;</p> <p>porównuje działanie instrukcji iteracyjnej <code>for</code> i w języku Python i powtarzaj w języku Scratch, wskazując różnice i podobieństwa;</p> <p>pisze programy realizujące trudniejsze algorytmy wymagające zastosowania instrukcji warunkowej i iteracyjnej;</p> <p>samodzielnie rozwiązuje zadania z podręcznika;</p> <p>bierze udział w konkursach informatycznych</p>

10.	Realizacja algorytmu Euklidesa w wersji z odejmowaniem	<p>przedstawia na przykładach działanie algorytmu Euklidesa w wersji z odejmowaniem oraz analizuje schemat blokowy tego algorytmu;</p> <p>zna działanie instrukcji <code>while</code> w wybranym języku programowania;</p> <p>wie, co to jest algorytm iteracyjny;</p> <p>korzystając z podręcznika, zapisuje w postaci programu algorytm Euklidesa w wersji z odejmowaniem w wybranym środowisku programowania (Scratch lub Baltie) i w wybranym języku programowania (C++ lub Python)</p>	<p>porównuje <i>działanie instrukcji iteracyjnej</i> (<code>while</code> lub powtarzaj aż) w wybranym środowisku programowania i języku programowania, wskazując różnice i podobieństwa;</p> <p>samodzielnie objaśnia działanie programu, m.in. uzasadnia użycie poszczególnych instrukcji zgodnie z listą kroków lub schematem blokowym algorytmu</p>
11.	Badanie podzielności liczb naturalnych	<p>wie, jak oblicza się resztę z dzielenia, stosując operator modulo (mod, <code>%</code>) w wybranym środowisku programowania;</p> <p>przedstawia na przykładach działanie algorytmu badania podzielności liczb naturalnych;</p> <p>analizuje listę kroków algorytmu badania podzielności liczb i rysuje schemat blokowy tego algorytmu;</p> <p>zapisuje w postaci programu algorytm badania podzielności liczb naturalnych w wybranym środowisku programowania (Scratch lub Baltie) i w wybranym języku programowania (C++ lub Python)</p>	<p>potrafi samodzielnie porównać programy utworzone w środowisku programowania Baltie i języku C++ lub w języku Scratch i języku Python – wyjaśnia różnice i podobieństwa;</p> <p>pisze trudniejszy program wykorzystując algorytm badania podzielności liczb;</p> <p>wie, jak zliczać wprowadzane liczby</p>
12.	Realizacja algorytmu Euklidesa w wersji z dzieleniem i algorytm wyodrębniania cyfr danej liczby	<p>przedstawia na przykładach działanie algorytmu Euklidesa w wersji z dzieleniem oraz zapisuje ten algorytm w postaci programu;</p> <p>przedstawia na przykładach działanie algorytmu wyodrębniania cyfr danej liczby oraz zapisuje w postaci programu ten algorytm; objaśnia działanie programu, m.in. użycie poszczególnych instrukcji;</p> <p>Porównuje programy utworzone w środowisku programowania Baltie i języku C++ lub w języku Scratch i języku Python;</p> <p>sprawdza działanie programów dla różnych danych</p>	<p>rysuje schematy blokowy algorytmu Euklidesa w wersji z dzieleniem na podstawie listy kroków;</p> <p>rysuje schemat blokowy algorytmu wyodrębniania cyfr danej liczby na podstawie utworzonej samodzielnie listy kroków;</p> <p>wie, jak sprawdzić podzielność jednej liczby przez inną bez konieczności obliczania reszty z dzielenia; zapisuje ten algorytm w postaci listy kroków, schematu blokowego lub programu;</p> <p>zapisuje w postaci programu algorytm wyodrębniania cyfr danej liczby od najbardziej znaczącej do najmniej znaczącej;</p> <p>wyszukuje w Internecie więcej informacji na temat Euklidesa i jego algorytmu</p>

13.	Wyszukiwanie największego elementu w zbiorze nieuporządkowanym	<p>analizuje listę kroków algorytmu wyboru większej z dwóch liczb; zapisuje ten algorytm w wybranym środowisku programowania (Baltie lub Scratch) i języku programowania (C++ lub Python);</p> <p>stosuje algorytm znajdowania największej liczby spośród n liczb do znajdowania najwyższego ucznia w klasie i zapisuje ten algorytm w wybranym środowisku programowania (Baltie lub Scratch) i języku programowania (C++ lub Python);</p> <p>pisze listy kroków i tworzy schematy blokowe prostych algorytmów</p>	<p>pisze listę kroków i buduje schemat blokowy algorytmu znajdowania najmniejszej liczby ze zbioru trzy-elementowego;</p> <p>potrafi samodzielnie wskazać różnice i podobieństwa realizacji wybranego algorytmu w programach utworzonych w środowisku Baltie i języku C++ lub w języku Scratch i języku Python</p>
14.	Wyszukiwanie danego elementu w zbiorze nieuporządkowanym i uporządkowanym	<p>opisuje algorytm wyszukiwania danego elementu w zbiorze nieuporządkowanym na konkretnym przykładzie;</p> <p>wie, na czym polega metoda połowienia;</p> <p>opisuje algorytm wyszukiwania danego elementu w zbiorze uporządkowanym na konkretnym przykładzie;</p> <p>stosuje algorytm poszukiwania przez połowienie, np. w grze w zgadywanie liczby i zapisuje ten algorytm w wybranym środowisku (Scratch lub Baltie) oraz w wybranym języku programowania (C++ lub Baltie);</p> <p>zna polecenia umożliwiające generowanie liczb losowych w wybranym środowisku programowania (Scratch lub Baltie) i w wybranym języku programowania (C++ lub Python)</p>	<p>potrafi podać inne od omówionych w podręczniku przykłady zastosowania algorytmu znajdowania wybranego elementu w zbiorze nieuporządkowanym i uporządkowanym;</p> <p>potrafi napisać listę kroków algorytmu wyszukiwania przez połowienie</p>
15.	Porządkowanie elementów zbioru metodą przez wybieranie i metodą przez zliczanie	<p>omawia algorytm sortowania metodą przez wybieranie, korzystając z rysunku w podręczniku i przygotowanych pomocy dydaktycznych</p> <p>analizuje listę kroków tego algorytmu;</p> <p>omawia algorytm sortowania metodą przez zliczanie, korzystając z rysunku w podręczniku i przygotowanych pomocy dydaktycznych</p>	<p>potrafi podać inne od omówionych w podręczniku przykłady zastosowania algorytmów porządkowania metodą przez wybieranie i metodą przez zliczanie;</p> <p>samodzielnie rozwiązuje zadania z podręcznika;</p> <p>bierze udział w konkursach informatycznych</p>

16.	Stosowanie procedur w języku Scratch oraz funkcji w języku Python	<p>definiuje i stosuje procedury w języku Scratch i/lub Baitie;</p> <p>rozumie na czym polega wywołanie procedury/funkcji;</p> <p>rozdzielnia parametry formalne i aktualne;</p> <p>definiuje funkcje w języku C++, korzystając z opisu w podręczniku (na przykładzie funkcji wyszukującej maksimum z n liczb);</p> <p>stosuje funkcje w języku C++, korzystając z przykładowych programów opisanych w podręczniku</p>	<p>wyjaśnia znaczenie stosowania procedur/funkcji;</p> <p>wyjaśnia na przykładach, kiedy stosujemy dany rodzaj funkcji w języku C++ (zwracającą wartość i niezwracającą wartości)</p>
17.	Stosowanie list w języku Python do przechowywania danych	<p>wyjaśnia, korzystając z rysunku i opisu w podręczniku, czym są zmienne indeksowane;</p> <p>deklaruje tablice, wczytuje i wyprowadza elementy tablicy na ekran</p>	<p>potrafi samodzielnie przeanalizować gotowy program i na jego podstawie pisać trudniejsze programy, stosując funkcje i tablice;</p> <p>uczestniczy w konkursach i olimpiadach informatycznych</p>

18.	Zapisywanie w języku Python algorytmów porządkowania metodą przez wybieranie i metodą przez zliczanie	<p>posługując się opisem w podręczniku, wyjaśnia sposób zaprogramowania algorytmów porządkowania elementów w zbiorze metodą przez proste wybieranie i zliczanie w języku C++;</p> <p>programuje algorytmy porządkowania elementów w zbiorze metodą przez proste wybieranie i zliczanie, analizując opis programów z podręcznika; definiuje odpowiednie funkcje</p>	<p>potrafi samodzielnie przeanalizować gotowy program i na jego podstawie pisać trudniejsze programy, stosując funkcje i tablice;</p> <p>uczestniczy w konkursach i olimpiadach informatycznych</p>
19.	Zapisywanie w języku Python algorytmów wyszukiwania elementu w zbiorze uporządkowanym i nieuporządkowanym	<p>posługując się opisem w podręczniku, wyjaśnia sposób zaprogramowania wyszukiwania elementu w zbiorze uporządkowanym w języku C++;</p> <p>programuje algorytmy wyszukiwania elementu w zbiorze uporządkowanym, analizując opis programów z podręcznika; definiuje odpowiednie funkcje</p> <p>programuje algorytmy wyszukiwania elementu w zbiorze nieuporządkowanym; definiuje odpowiednie funkcje</p>	<p>potrafi samodzielnie przeanalizować gotowy program i na jego podstawie pisać trudniejsze programy, stosując funkcje i tablice;</p> <p>uczestniczy w konkursach i olimpiadach informatycznych</p>

20.	Prezentacja multimedialna w dziesięciu krokach	<p>tworzy prezentację na wybrany temat: przygotowuje plan prezentacji; wstawia do slajdów tekst, obrazy, hiperłącza, przyciski akcji, dodaje animacje i wykonuje przejścia między slajdami;</p> <p>dba o poprawność redakcyjną tekstów;</p> <p>korzysta z szablonów slajdów;</p> <p>dodaje do prezentacji efekty dźwiękowe;</p> <p>przygotowuje i uruchamia pokaz</p>	<p>potrafi samodzielnie zaprojektować i przygotować multimedialną prezentację na wybrany temat, cechującą się ciekawym ujęciem zagadnienia, interesującym układem slajdów</p>
21.	Nagranie zawartości ekranu i podstawowy montaż filmu oraz dodawanie multimedii do prezentacji	<p>zna podstawowe możliwości programu do obróbki filmu;</p> <p>nagrywa filmy;</p> <p>poddaje nagrany film podstawowej obróbce;</p> <p>wstawia film do prezentacji;</p> <p>współpracuje w grupie przy tworzeniu projektu, wykonując samodzielnie zadania szczegółowe;</p> <p>przygotowuje prezentację w postaci albumu fotograficznego;</p> <p>prezentuje efekty pracy grupowej</p>	<p>potrafi samodzielnie zaprojektować i przygotować multimedialną prezentację na wybrany temat, cechującą się ciekawym ujęciem zagadnienia, interesującym układem slajdów</p>
22.	Historia i rozwój informatyki – projekt	<p>podaje i omawia przykłady dawnych przyrządów i maszyn do wykonywania obliczeń;</p> <p>omawia główne etapy historycznego rozwoju informatyki i technologii; uwzględnia m.in. pierwsze komputery, pierwszy język programowania;</p> <p>w kontekście rozwoju informatyki dyskutuje (wspólnie z nauczycielem) na temat różnych zawodów wymagających kompetencji informatycznych;</p> <p>tworzy prezentację na temat związany z historią i rozwojem informatyki;</p> <p>współpracuje w grupie przy tworzeniu projektu, wykonując samodzielnie zadania szczegółowe;</p> <p>prezentuje efekty pracy grupowej</p>	<p>wyszukuje dodatkowe informacje na temat historycznego rozwoju informatyki i technologii;</p> <p>w kontekście rozwoju informatyki dyskutuje (wspólnie z nauczycielem) na temat dalszego kształcenia informatycznego w zakresie rozszerzonym;</p> <p>potrafi samodzielnie zaprojektować i przygotować multimedialną prezentację na wybrany temat, cechującą się ciekawym ujęciem zagadnienia, interesującym układem slajdów;</p> <p>pełni funkcję koordynatora grupy</p>

23.	Tworzenie strony internetowej – podstawowa struktura	<p>zna ogólne zasady projektowania stron WWW i wie, jakie narzędzia umożliwiają ich tworzenie;</p> <p>potrafi, korzystając z podstawowych znaczników HTML, utworzyć prostą strukturę strony;</p> <p>formatuje tekst na stronie</p>	<p>zna większość znaczników HTML;</p> <p>posługuje się wybranym programem przeznaczonym do tworzenia stron WWW;</p> <p>potrafi tworzyć proste witryny składające się z kilku połączonych ze sobą stron;</p> <p>dba o poprawność merytoryczną i redakcyjną tekstów</p>
24.	Tworzenie strony internetowej – poprawianie wyglądu strony	<p>stosuje na stronie wypunktowania;</p> <p>tworzy tabele i umieszcza w nich informacje;</p> <p>wstawia na stronę obrazy i odsyłacze do innych stron;</p> <p>wie, jak opublikować stronę w Internecie</p>	<p>samodzielnie rozbudowuje tworzoną stronę internetową, dodając m.in. tło, linki i strony;</p> <p>publikuje stronę WWW w Internecie</p>
25.	Systemy zarządzania treścią – projekt	<p>wie, jak tworzy się proste blogi;</p> <p>wie, czym są systemy zarządzania treścią;</p> <p>korzystając z przykładowego systemu zarządzania treścią (np. systemu WordPress), tworzy prostego bloga, m.in.: umieszcza nowe wpisy, dodaje kategorie, tagi;</p> <p>wie, jak dodać nową stronę, zastosować motyw</p>	<p>potrafi samodzielnie tworzyć bloga z wykorzystaniem systemu zarządzania treścią, odszukując potrzebne opcje</p>
26.	Systemy zarządzania treścią – zadania projektowe	<p>wie, jakie są możliwości pracy w chmurze; stosuje je w pracy zespołowej przy tworzeniu projektów;</p> <p>współpracuje w grupie przy tworzeniu projektu, wykonując samodzielnie zadania szczegółowe</p>	<p>potrafi pełnić funkcję koordynatora grupy;</p> <p>wykorzystuje możliwości pracy w chmurze</p>