Vademecum Nauczyciela

Wdrażanie podstawy programowej w szkole ponadpodstawowej

Scenariusze zajęć

dla doradców metodycznych

Informatyka

Moduł 1, 2, 3, 4

MINISTERSTWO EDUKACJI NARODOWEJ

ORE Ośrodek Rozwoju Edukacji

INFORMATYKA

SCENARIUSZ ZAJĘĆ
MODUŁ 1

Autor: dr Anna Beata Kwiatkowska

Temat: Założenia „nowej" podstawy programowej informatyki w szkole ponadpodstawowej

Czas: 2 godz. dydaktyczne (90 min)

Cele zajęć:

Uczestnik po zajęciach:

- wie, na czym polegają zmiany w kształceniu informatycznym w porównaniu z ustępującą podstawą, w kontekście nowego podejścia do nauczania tego przedmiotu, opartego na myśleniu komutacyjnym
- zna cele ogólne kształcenia w zakresie informatyki
- zna cele szczegółowe edukacji informatycznej w nauczaniu zintegrowanym i przedmiotu: informatyka na kolejnych etapach edukacyjnych
- umie zaprezentować spiralne podejście do realizacji poszczególnych zagadnień informatycznych, w szczególności do algorytmiki i programowania
- zna nowe podejście do kształcenia informatycznego ogólnego, w szczególności dotyczące rozwiązywania problemów z różnych dziedzin, i podaje przykłady takich problemów
- zna nowe podejście do realizacji zagadnień z algorytmiki i programowania na poziomie rozszerzonym

Metody pracy:

- elementy wykładu
- dyskusja
- wymiana doświadczeń

Formy pracy:

- indywidualna
- grupowa

Materiały dydaktyczne:

- prezentacja multimedialna
- podstawa programowa informatyki
- rozporządzenia MEN dotyczące „nowej" podstawy programowej

Przebieg zajęć:

1. Czynności organizacyjne – 2 min.
2. Zajęcia rozpoczynamy od wyjaśnienia, jak w edukacji należy rozumieć nowe podejście do przedmiotu: informatyka. Zwracamy uwagę na rozległość dziedziny: informatyka, rozumianej ogólnie **(slajdy 2–4)**, na obszary tej dziedziny rozważane w szkole **(slajd 5)** oraz objaśniamy pojęcie myślenia komputacyjnego **(slajd 6)** – 5 min.
3. Następnie przedstawiamy ogólne informacje o „nowej" podstawie programowej informatyki i filozofię zmian **(slajdy 7–9)**, wraz z siatkami godzin **(slajdy 10–15)**. Należy zwrócić szczególną uwagę na najważniejsze aspekty nauczania informatyki **(slajd 16)** oraz na to, że cele ogólne kształcenia informatycznego są takie same dla wszystkich uczniów przez wszystkie lata w szkole **(slajd 17)**. Podkreślamy, że każdy cel ogólny ma być realizowany spiralnie w każdej klasie. Porównujemy „ustępującą" i „nową" podstawę ogólnie i bardziej szczegółowo w zakresie najważniejszych obszarów zmian, takich jak: rozwiązywanie problemów/algorytmika, programowanie wizualne i tekstowe, robotyka, projektowanie 3D oraz bezpieczeństwo i aspekty prawne **(slajdy 18–21)** – 10 min.
4. Objaśniamy podejście informatyczne do rozwiązywania problemów z różnych dziedzin. Prosimy uczestników o podawanie przykładów problemów z różnych dziedzin oraz dyskusję nad poszczególnych etapami ich rozwiązywania zgodnie ze schematem **(slajdy 22–24)**. Należy zderzyć wyniki badań (Jean Piaget) ze współczesnymi cechami młodego pokolenia i przedstawić argumentację przemawiającą za możliwością wcześniejszego podejścia abstrakcyjnego przy rozwiązywaniu problemów **(slajd 25)** – 8 min.
5. Aby prawidłowo realizować podstawę programową informatyki w szkole ponadpodstawowej, należy dobrze znać jej zapisy dotyczące szkoły podstawowej. Omawiamy cele szczegółowe edukacji informatycznej w klasach I–III oraz informatyki w klasach IV–VI i VII–VIII (**slajdy 26–30)**. Należy podkreślić, że w klasach I–III uczniowie poznają pojęcia informatyczne intuicyjnie, często bez użycia komputera. W klasach IV–VI podejście jest bardziej formalne. Istotne jest podkreślenie znaczenia etapu przejścia na etap abstrakcyjnego podejścia do programowania: z wizualnego języka programowania do języka tekstowego, od klasy VII – 10 min.
6. Omawiamy nowe podejście do kształcenia informatycznego w szkole ponadpodstawowej na poziomie podstawowym. Trzeba podkreślić, że uczniowie z innych klas niż informatyczne wybrali inny przedmiot zainteresowań i powinni uczyć się informatyki przez rozwiązywanie problemów z tych dziedzin **(slajdy 31, 32)**. Należy poprosić uczestników o dokonanie analizy zapisów „nowej" podstawy programowej informatyki i wykonanie zestawienia w tabeli algorytmów omawianych na poszczególnych etapach kształcenia pod względem spiralności. Po zakończeniu dyskusji trzeba

zestawić wyniki uzyskane przez uczestników z informacjami przedstawionymi w prezentacji **(slajdy 33, 34)** – 10 min.

7. Należy poprosić uczestników o przeanalizowanie przykładów – wybranych spośród przedstawionych problemów – oraz wskazanie algorytmów potrzebnych do ich rozwiązania, a także zakresu prac związanych z rozwiązywaniem problemu na poszczególnych etapach edukacyjnych **(slajdy 35, 46)** – 10 min.
8. Omawiamy analogicznie sposób realizacji zagadnień podstawy programowej dla szkoły ponadpodstawowej na poziomie rozszerzonym. Istotą edukacji na poziomie rozszerzonym jest formalne podejście do algorytmiki i programowania. Podczas omawiania zaznaczamy, że nawet praca z aplikacjami użytkowymi powinna być wzbogacona elementami programowania **(slajd 47, 48)**. Nowością jest podział omawianych algorytmów na trzy grupy. Należy podkreślić, że uczniowie nie muszą potrafić zaprogramować algorytmów z grupy trzeciej, lecz wystarczy, że umieją je omówić, zaprezentować i znają ich współczesne zastosowania **(slajd 49).** Trzeba przedyskutować z uczestnikami, jakie umiejętności programowania uczniowie nabywają w szkole podstawowej i jak je zwiększają na poziomie rozszerzonym w szkole ponadpodstawowej. Wyniki dyskusji konfrontować z informacjami zawartymi w prezentacji **(slajd 50)** – 10 min.
9. Przedstawiamy zakres tematów realizowanych w poszczególnych szkołach branżowych **(slajd 51)**. Zwracamy uwagę, że łącznie w szkole branżowej I i II stopnia należy zrealizować wszystkie zagadnienia z zakresu kształcenia na poziomie podstawowym w szkole ponadpodstawowej oraz że filozofia kształcenia informatycznego w szkole branżowej jest taka sama – 5 min.
10. Na koniec zaznaczamy, jak prawidłowa realizacja zagadnień podstawy programowej informatyki może pomóc kształceniu w zakresie wszystkich przedmiotów oraz całemu społeczeństwu w rozwoju kompetencji cyfrowych. Omawiamy poszczególne obszary informatyki rozważane w edukacji i szczegółowo wynikające z nich korzyści **(slajdy 52–57).** Odnosimy się do wyników matury z matematyki, wskazując obszary, które wymagają interwencji. Pokazujemy korelacje tych obszarów z najważniejszymi obszarami kształcenia informatycznego **(slajdy 58, 59)**. Omawiamy wkład nowego kształcenia informatycznego w przyszły rozwój sztucznej inteligencji **(slajd 60)** – 5 min.
11. Pytamy uczestników, czy założenia podstawy programowej są zrozumiałe. Przeznaczamy czas na ewentualne wyjaśnienia – 5 min.
12. Czas na pytania i wyjaśnienia wątpliwości uczestników. Podsumowanie modułu – 10 min.

INFORMATYKA

SCENARIUSZ ZAJĘĆ
MODUŁ 2

Autor: dr Anna Beata Kwiatkowska

Temat: Myślenie komputacyjne na przykładzie problemów z algorytmami tekstowymi w tle

Czas: 2 godz. dydaktyczne (90 min)

Cele zajęć:

Uczestnik po zajęciach:

- potrafi podać przykłady problemów z zakresu innych dziedzin korzystających z metod i technik informatyki
- potrafi poprzez rozwiązywanie problemów z zakresu innych dziedzin wprowadzić algorytmy na tekstach
- wie, jak korzystając z odpowiednich środków technicznych (animacja, pokaz), zachęcać do kreatywnego odkrywania algorytmów przez uczniów
- wie, jak powinny wyglądać prezentacje uczniowskie trudniejszych algorytmów
- zna większość algorytmów na tekstach, ujętych w podstawie programowej informatyki

Metody pracy:

- elementy wykładu
- dyskusja
- praca z wykorzystaniem możliwości komputera

Formy pracy:

- indywidualna
- grupowa

Materiały dydaktyczne:

- prezentacja multimedialna
- materiał do ćwiczeń

Przebieg zajęć:

1. Czynności organizacyjne – 2 min.
2. Zajęcia rozpoczynamy od przedstawienia, jak w edukacji należy rozumieć nowe podejście do przedmiotu: informatyka. Przypominamy pojęcie myślenia komputacyjnego **(slajdy 2, 3)**,

omawiamy pojęcie algorytmu, jego własności i sposoby zapisu **(slajd 4)**, przypominamy również, jak należy rozumieć programowanie oraz jego etapy **(slajdy 5, 6)** – 8 min.

3. Omawiamy pierwszy, przykładowy, problem z innej dziedziny, przy którego rozwiązaniu pomocna jest znajomość typu łańcuchowego rozważanego na lekcjach informatyki oraz algorytmu dotyczącego porównywania tekstów. Rozważania prowadzimy na podstawie problemu porównywania łańcuchów DNA. Przedstawiamy problem z punktu widzenia biologii **(slajdy 7–10)** oraz możliwość jego rozwiązania przy pomocy dostępnych w internecie narzędzi **(slajdy 11–13)**. Zastanawiamy się jednocześnie, z jakiego algorytmu może to narzędzie korzystać – 10 min.
4. Przechodzimy do specyfikacji problemu, precyzując dane i wyniki **(slajd 14)**. Przedstawiamy animację prowadzącą do otrzymania wyniku **(slajdy 15–39)**. Animacja powinna doprowadzić do określenia przez uczestników szkolenia zastosowanego w animacji algorytmu. Następnie prosimy uczestników o podzielenie się na cztery grupy i zapisanie algorytmu: w języku naturalnym, języku schematów blokowych, pseudokodzie i wybranym języku programowania. Porównujemy zapisy uczestników z informacjami zawartymi w prezentacji **(slajdy 41–54)**. Przy omawianiu rozwiązań przypominamy sposób zapisu potrzebnych konstrukcji językowych oraz działań na typie łańcuchowym. Przy okazji przypominamy, że każdy znak w komputerze jest reprezentowany w postaci kodów ASCII **(slajd 55)**. Informacja ta przyda się przy samodzielnym rozwiązywaniu ćwiczeń do tego modułu – 15 min.
5. Przechodzimy do omówienia drugiego problemu, do którego rozwiązania wykorzystuje się algorytm badający, czy dany łańcuch jest palindromem. Problem dotyczy generowania symetrycznych względem osi wzorów tkanin **(slajd 56)**. Następnie przedstawiamy specyfikację algorytmu oraz animację pokazującą, w jaki sposób badać symetrię **(slajdy 57, 58)**. Zaznaczamy, że badanie zostanie przerwane, gdy natrafimy na brak zgodności we wzorze. Prosimy uczestników o samodzielne zapisanie algorytmu w każdej z postaci. Weryfikujemy poprawność na podstawie programu przedstawionego w prezentacji **(slajd 59)** – 15 min.
6. Omawiamy trzeci przykład problemu. Dotyczy on wyszukiwania wzorca, który ma bardzo rozległe zastosowanie w pracy z informacją **(slajdy 60–63)**. Problem ten doczekał się wielu algorytmów, z których wymieniamy kilka z nazwy, krótko charakteryzując techniki, z jakich korzystają, oraz na jakim poziomie kształcenia informatycznego powinny być omawiane. Dwa z nich mogą być omówione przez nauczyciela tylko na lekcjach lub zaprezentowane przez uczniów, przy czym uczniowie nie muszą umieć ich programować **(slajd 64)**. Prosimy uczestników o dyskusję, w jaki sposób rozwiązać problem szukania wzorca. Wzmacniamy przedstawione rozwiązanie animacją **(slajdy 65–72)**. Omawiamy zapis algorytmu w wybranym języku programowania **(slajdy 73, 74)**. Przeprowadzamy dyskusję dotyczącą złożoności obliczeniowej algorytmu naiwnego **(slajd 75)** – 15 min. Prezentujemy uczestnikom drugi, trudniejszy algorytm – Rabina-Karpa, ale rozwiązujący

problem szukania wzorca w innym czasie. Algorytm korzysta z innych, wcześniej poznanych algorytmów i działań: schematu Hornera, dzielenia modulo, algorytmu naiwnego. Przypominamy te zagadnienia i prowadzimy dyskusję dotyczącą złożoności obliczeniowej zaprezentowanego algorytmu. Algorytm korzysta z funkcji skrótu, która przydaje się w rozwiązywaniu różnorodnych problemów. Warto przytoczyć kilka takich sytuacji **(slajdy 76–80)** – 10 min.

7. Pytamy uczestników, czy omówione problemy są zrozumiałe i czy jasny jest dobór przykładów w kontekście kształcenia myślenia komputacyjnego. Zachęcamy do samodzielnego przestudiowania materiałów do ćwiczeń i ich wykonania. Przeznaczamy czas na ewentualne wyjaśnienia – 5 min.
8. Czas na pytania i wyjaśnienie wątpliwości uczestników. Podsumowanie modułu –10 min.

INFORMATYKA

SCENARIUSZ ZAJĘĆ
MODUŁ 3

Autor: dr Anna Beata Kwiatkowska

Temat: Kształcenie spiralne na podstawie problemu sortowania

Czas: 2 godz. dydaktyczne (90 min)

Cele zajęć:

Uczestnik po zajęciach:

- wie i potrafi przedstawić argumenty uzasadniające ważne miejsce problemu sortowania w edukacji informatycznej
- potrafi objaśnić spiralne rozwijanie się problematyki sortowania w podstawie programowej informatyki na kolejnych etapach edukacyjnych
- potrafi objaśnić, zaprezentować i zaimplementować algorytmy sortowań prostych i sortowania przez zliczanie
- zna szkic metody: dziel i zwyciężaj
- potrafi wykorzystać metodę: dziel i zwyciężaj w algorytmach min-max i sortowania przez scalanie
- potrafi obliczyć liczbę operacji dominujących w algorytmie

Metody pracy:

- elementy wykładu
- dyskusja
- praca z wykorzystaniem możliwości komputera

Formy pracy:

- indywidualna
- grupowa
- wspólne tworzenie programu

Materiały dydaktyczne:

- prezentacja multimedialna
- materiał do ćwiczeń
- komentarz do podstawy programowej informatyki

Przebieg zajęć:

1. Czynności organizacyjne – 2 min.
2. Zajęcia rozpoczynamy od przedstawienia argumentów uzasadniających zajmowanie przez algorytmy sortowania ważnego miejsca w kształceniu informatycznym:
   z jednej strony większość czynności wykonywanych np. na komputerze polega na sortowaniu,
   z drugiej strony każdy z omawianych algorytmów sortowania wnosi nową metodę lub technikę do kształcenia informatycznego **(slajdy 1, 2)**.
   Prezentujemy, jak należy rozumieć pojęcie sortowania oraz jaka jest jego formalna specyfikacja **(slajd 3)** – 5 min.
3. Przechodzimy do omówienia spiralnego rozszerzania wiadomości i umiejętności dotyczących sortowania na kolejnych etapach edukacyjnych. Wskazujemy, które zapisy podstawy programowej informatyki dla edukacji wczesnoszkolnej dotyczą sortowania **(slajd 4)**. Przedstawiamy przykłady problemów, które wymagają ułożenia elementów w logicznym porządku i tworzą sekwencję poleceń. Mogą to być np. zagadki lub łamigłówki z konkursów informatycznych **(slajdy 5, 6)**. Prosimy uczestników o dyskusję i wskazanie przykładów takich problemów – 5 min.
4. Omawiamy zapisy podstawy programowej informatyki dla klas IV–VI, nawiązujące do problemu sortowania **(slajd 7)**. Ponieważ w zapisach podstawy pojawia się algorytm szukania elementu najmniejszego lub największego, zaznaczamy, że jest to algorytm, którego znajomość jest potrzebna m.in. do sortowania przez proste wybieranie – dlatego szczegółowo go omawiamy **(slajdy 8–14)**. Prosimy uczestników o utworzenie programu realizującego ten algorytm w wybranym języku programowania. Wynik pracy porównujemy z informacją zawartą w prezentacji **(slajd 15)** – 10 min.
5. Przechodzimy do omówienia zagadnień związanych z sortowaniem, zapisanych w podstawie programowej informatyki dla klas VII–VIII **(slajd 16)**. Przedstawiamy animacje algorytmu przez proste wybieranie i pokazujemy, w jaki sposób wykorzystać w tym algorytmie wcześniej poznany algorytm szukania minimum. Zwracamy uwagę, że na kolejnych etapach sortowania przez proste wybieranie musimy znaleźć indeks miejsca, na którym stoi element minimalny **(slajdy 17–21).** Podkreślamy, że w tym sortowaniu operacją dominującą jest porównanie dwóch sortowanych elementów. Wspólnie obliczamy dokładną liczbę takich porównań w całym sortowaniu, liczba ta nie jest zależna od wstępnej kolejności elementów. Określamy złożoność obliczeniową algorytmu **(slajd 22)**. Wspólnie z uczestnikami zapisujemy algorytm na tablicy, począwszy od wewnętrznej pętli opisującej etap sortowania **(slajd 23)** – 15 min.

6. Przechodzimy do omówienia algorytmu sortowania przez zliczanie. Sortowanie to jest najbardziej naturalną metodą sortowania, wykorzystywaną od najmłodszych lat. Dziecko porządkuje klocki, zabawki, np. pod względem kolorów lub kształtów, stosuje przy tym zliczanie. Należy omówić ograniczenia w zastosowaniu tego sortowania. Omówienie sortowania powinno prowadzić do utrwalenia rozróżnienia indeksu od wartości elementu. Przedstawiamy jego poszczególne etapy **(slajdy 24–29)**. Należy zwrócić szczególną uwagę na zapis operacji zliczania na etapie drugim **(slajd 30)**. Wspólnie z uczestnikami implementujemy poszczególne etapy sortowania – zapraszamy kolejnych uczestników do zapisywania poleceń na głównym komputerze, w sposób widoczny dla wszystkich osób. Omawiamy cechy charakterystyczne i złożoność obliczeniowa algorytmu **(slajdy 31, 32)** – 15 min.
7. Odszukujemy zapisy podstawy programowej informatyki dla szkoły ponadpodstawowej na poziomie podstawowym, które dotyczą sortowań **(slajd 33)**. Omawiamy algorytm sortowania przez wstawianie. Zwracamy przy tym uwagę, że w algorytmie nie ma zamian elementów, lecz ma miejsce ich nadpisywanie **(slajdy 34, 35)**. Przedstawiamy przykładową implementację algorytmu **(slajd 36)** – 10 min.
8. W postaci animacji prezentujemy algorytm sortowania bąbelkowego **(slajdy 37–46)**. Prosimy uczestników o samodzielne zaimplementowanie algorytmu. Przedstawiamy jego przykładową implementację z modyfikacją, w której kolejny etap prowadzimy do miejsca ostatniej zamiany **(slajd 47)**. Zwracamy uwagę, że ta modyfikacja może znacznie przyspieszyć działanie algorytmu, dyskutujemy, w jakich przypadkach tak się dzieje – 10 min.
9. Przechodzimy do podstawy programowej informatyki dla poziomu rozszerzonego **(slajd 48)**. Omawiamy optymalny algorytm iteracyjny **(slajdy 49–55)**, obliczając liczbę wykonywanych przez niego porównań elementów. Następnie przedstawiamy animację algorytmu rekurencyjnego, wprowadzając w ten sposób metodę: dziel i zwyciężaj. Jest to algorytm optymalny również ze względu na liczbę porównań. Algorytm ten stanowi znakomite wprowadzenie do algorytmu sortowania przez scalanie **(slajdy 56–75)**. Analogicznie omawiamy algorytm sortowania przez scalanie **(slajdy 76–99)**. Zwracamy szczególną uwagę na prawidłowe wykonywanie scalania dwóch ciągów uporządkowanych **(slajdy 100–107)** oraz że sortowanie realizujemy, wykorzystując pomocniczą strukturę przechowującą scalone na danym etapie elementy. Przeznaczamy czas na ewentualne wyjaśnienia – 13 min.
10. Zachęcamy uczestników do samodzielnego przestudiowania materiałów do ćwiczeń i ich wykonanie. Czas na pytania i wyjaśnianie wątpliwości uczestników. Podsumowanie modułu – 5 min.

INFORMATYKA

SCENARIUSZ ZAJĘĆ
MODUŁ 4

Autor: dr Anna Beata Kwiatkowska

Temat: Abstrakcyjny model problemu z grafami w tle

Czas: 2 godz. dydaktyczne (90 min)

Cele zajęć:

Uczestnik po zajęciach:

- potrafi wskazać miejsce modelowania grafami w informatycznym rozwiązywaniu problemu
- zna przykłady prostych problemów, w których tle znajdują się grafy
- potrafi odnajdywać niektóre własności grafów na podstawie analizy ich reprezentacji
- zna sposoby implementacji rozbudowanych struktur danych
- potrafi dobrać reprezentację grafu do algorytmu
- potrafi objaśnić, zaprezentować i zaimplementować algorytm sortowania topologicznego
- sprawnie posługuje się różnorodnymi strukturami danych przy implementacji algorytmów: kolejka, lista/tablica, struktury dwuwymiarowe, czytanie danych z pliku

Metody pracy:

- elementy wykładu
- dyskusja
- praca z wykorzystaniem możliwości komputera

Formy pracy:

- indywidualna
- grupowa
- wspólne tworzenie programu

Materiały dydaktyczne:

- prezentacja multimedialna
- materiał do ćwiczeń
- komentarz do podstawy programowej informatyki

Przebieg zajęć:

1. Czynności organizacyjne – 2 min.
2. Zajęcia rozpoczynamy od przypomnienia etapów informatycznego rozwiązywania problemu. Zwracamy uwagę na fakt, że ważnym elementem rozwiązywania problemu jest umiejętność tworzenia modelu sytuacji problemowej, co ułatwia analizę i pozwala szybciej/łatwiej odkryć algorytm rozwiązujący problem. Należy zwrócić uwagę, że badania Piageta pochodzą z ubiegłego wieku i granica wieku dzieci, od osiągnięcia którego mogą myśleć abstrakcyjnie, prawdopodobnie jeszcze się obniżyła ze względu na postęp technologiczny **(slajdy 2, 3)** – 10 min.
3. Przechodzimy do omówienia prostych sytuacji problemowych, przedstawionych z użyciem grafów **(slajdy 4–12)**. Następnie omawiamy przykładowy problem, który może być przedstawiony za pomocą modelu grafu. Na takim modelu łatwo jest odkryć sposób postępowania przy ustalaniu czynności, który jest jednym z algorytmów sortowania topologicznego **(slajdy 13– 22)**. Pokazujemy jeszcze inne przykłady prostych problemów z grafami w tle **(slajdy 23–25)**. Prosimy uczestników o dyskusję i wskazanie przykładów takich problemów – 13 min.
4. Omawiamy ciekawą historię dotyczącą początków teorii grafów, związaną z jej twórcą **(slajdy 26 –29)**. Prezentujemy dwie metody pamiętania grafów w komputerze **(slajdy 30–33)**. Pokazujemy, jak analiza struktury może dać szybką odpowiedź na niektóre pytania **(slajd 34)**. Zwracamy uwagę, że dobór struktury ma wpływ na złożoność obliczeniową algorytmów dla grafów. Omawiamy struktury języka pozwalające na pamiętanie grafów w prezentacji przykładów **(slajdy 35 –38)**. Implementujemy grafy na poszczególnych strukturach **(slajdy 39–41)**, analizujemy dobór struktury w zależności od rozważanego problemu – 15 min.
5. Przechodzimy do pokazania, jak uczniowie mogą zaprezentować algorytm sortowania topologicznego. Należy zwrócić uwagę na struktury danych potrzebne dla prawidłowego działania algorytmu. Przypominamy, jak pracować z listami/tablicami, strukturą kolejki, pobieraniem danych z pliku **(slajdy 42–62)**. Jeśli starczy czasu, wspólnie implementujemy algorytm – 25 min.
6. Przeprowadzamy dyskusję prowadzącą do zidentyfikowania innych problemów, które można przedstawić za pomocą grafów. Próbujemy wskazać potrzebne algorytmy **(slajd 63)** – 15 min.
7. Zachęcamy do samodzielnego przestudiowania materiałów do ćwiczeń i ich wykonania – 5 min.
8. Czas na pytania i wyjaśnienie wątpliwości uczestników. Podsumowanie modułu – 5 min.

INFORMATYKA

MATERIAŁY DO ĆWICZEŃ
MODUŁ 1

Autor: dr Anna Beata Kwiatkowska

Temat: „Nowa" podstawa programowa – filozofia zmian

Ćwiczenie 1.1.

Zapoznaj się z treścią Rozporządzenia Ministra Edukacji Narodowej z dnia 14 lutego 2017 r. w sprawie podstawy programowej wychowania przedszkolnego oraz podstawy programowej kształcenia ogólnego dla szkoły podstawowej, w tym dla uczniów z niepełnosprawnością intelektualną w stopniu umiarkowanym lub znacznym, kształcenia ogólnego dla branżowej szkoły I stopnia, kształcenia ogólnego dla szkoły specjalnej przysposabiającej do pracy oraz kształcenia ogólnego dla szkoły policealnej (http://prawo.sejm.gov.pl/isap.nsf/download.xsp/WDU20170000356/O/D20170356.pdf)

a) Znajdź i przeanalizuj wszystkie zapisy w ogólnym opisie wychowania przedszkolnego i edukacji wczesnoszkolnej oraz pozostałych klas szkoły podstawowej dotyczące kształcenia informatycznego (oprócz treści samej podstawy informatyki).
b) Zapoznaj się z zapisami podstawy programowej edukacji informatycznej i informatyki.
c) Zapoznaj się z warunkami i sposobami realizacji zapisów podstawy edukacji informatycznej i informatyki opisanymi w rozporządzeniu.
d) W zapisach podstawy dla innych przedmiotów znajdź miejsca, gdzie wykorzystywane są umiejętności informatyczne.

Ćwiczenie 1.2.

Zapoznaj się z treścią Rozporządzenia Ministra Edukacji Narodowej z dnia 30 stycznia 2018 r. w sprawie podstawy programowej kształcenia ogólnego dla liceum ogólnokształcącego, technikum oraz branżowej szkoły II stopnia (http://prawo.sejm.gov.pl/isap.nsf/download.xsp/WDU20180000467/O/D20180467.pdf) i przeprowadź analizę analogiczną do analizy z ćwiczenia 1.1.

Ćwiczenie 1.3.

Porównaj siatki godzin dla przedmiotów informatycznych w „starej" i „nowej" podstawie programowej.

Ćwiczenie 1.4.

Przeprowadź analizę zapisów „nowej" podstawy programowej informatyki dla wszystkich etapów kształcenia i wykonaj zestawienie w postaci tabeli algorytmów omawianych na poszczególnych etapach kształcenia, wskazując podejście spiralne.

Ćwiczenie 1.5.

Zapoznaj się z przykładami problemów z różnych dziedzin, omówionymi w prezentacji. Wskaż algorytmy potrzebne do ich rozwiązania lub do których dany problem nawiązuje.

Ćwiczenie 1.6.

Przeanalizuj spiralność kształcenia w zakresie programowania na poszczególnych etapach edukacyjnych. Wyniki porównaj z przedstawionymi podczas wykładu w prezentacji.

MATERIAŁY DO ĆWICZEŃ
MODUŁ 2

Autor: dr Anna Beata Kwiatkowska

Temat: Myślenie komputacyjne na przykładzie problemów z algorytmami tekstowymi w tle

Skorzystaj z materiału metodycznego zawartego w prezentacji do modułu 2. i wykonaj następujące ćwiczenia:

Ćwiczenie 2.1.

Polskie imiona żeńskie kończą się na literę a. Napisz program, który uzupełni początek listu do ciebie, sformułowany na podstawie twojego imienia: Szanowna Pani, Szanowny Panie.

Ćwiczenie 2.2.

Dane są dwa łańcuchy reprezentujące sekwencje nukleotydowe dwóch organizmów A i B, każdy o długości n. Zapisz w różnych postaciach (język naturalny, schemat blokowy, pseudokod, język programowania) rozwiązanie dla problemów:

a) określenie liczby niezgodnych nukleotydów miedzy organizmami A i B,
b) obliczenia procentu niezgodności organizmów A i B.

Ćwiczenie 2.3.

Dany jest łańcuch znaków A długości n. W wybranym języku programowania utwórz program z funkcją realizującą algorytm sprawdzania, czy łańcuch A jest palindromem. Funkcja powinna zwracać odpowiedź TAK w przypadku, gdy A jest palindromem lub NIE w przypadku, gdy A nie jest palindromem.

Ćwiczenie 2.4.

Masz do dyspozycji dwa znaki: @, +. Dla tkaniny o rozmiarach n x n znaków zaprojektuj losowo wzór taki, że po złożeniu tkaniny na pół wzory nałożą się na siebie (będą takie same).
Wskazówka: Projektuj wzór z wierszy znaków utworzonych jako palindromy.

Ćwiczenie 2.5.

Dane są dwa łańcuchy znaków: tekst T o długości n oraz wzorzec W o długości k – obydwa utworzone z liter alfabetu angielskiego. Utwórz program realizujący algorytm naiwnego wyszukiwania liczby wystąpień wzorca W w tekście T.

Ćwiczenie 2.6.

Polskie imiona żeńskie zawsze kończą się na literę a, natomiast imiona męskie na literę różną od litery a.

Dany jest plik tekstowy imiona.txt, w każdym wierszu tego pliku znajduje się polskie imię żeńskie lub męskie. Napisz funkcję sprawdzająca, czy dane imię jest imieniem żeńskim. Wykorzystaj tę funkcję w programie, który wszystkie imiona żeńskie z tego pliku umieści w pliku panie.txt, a wszystkie imiona męskie umieści w pliku panowie.txt. Każde imię powinno być zapisane w odrębnym wierszu, zgodnie z kolejnością w pliku imiona.txt. Dodatkowo w ostatnim wierszu każdego pliku wynikowego zapisz liczbę umieszczonych w nim imion.

Ćwiczenie 2.7.

Dane są dwa słowa złożone, w których występują tylko litery a i b. Masz funkcję obliczającą skrót (hasz) danego słowa. Wykorzystaj ją do porównania, czy dwa wyrazy są równe. W przypadku takich samych wartości skrótów słów skorzystaj z algorytmu naiwnego.

MATERIAŁY DO ĆWICZEŃ
MODUŁ 3

Autor: dr Anna Beata Kwiatkowska

Temat: Kształcenie spiralne na podstawie problemu sortowania

Skorzystaj z materiału metodycznego zawartego w prezentacji do modułu 3. i wykonaj następujące ćwiczenia:

Ćwiczenie 3.1.

Dane są liczby n, k ∈ N, a[i]∈N dla $0 \leq i \leq n-1$.

a) Odpowiedź na pytanie, czy istnieje takie i, że a[i] = k?
b) Ile wyrazów a[i] = k?
c) Oblicz średnią arytmetyczna liczb a[i].

Ćwiczenie 3.2.

Dane są cztery liczby naturalne n, x, a, b oraz n losowych liczb naturalnych z przedziału [a ,b] zapamiętanych w liście. Zastanów się, jak znaleźć miejsce wystąpienia najmniejszej z wylosowanych liczb? Utwórz program, który wypisze numery wszystkich takich miejsc.

Ćwiczenie 3.3.

Dane są: n∈N, po n losowych liczb w listach A i B posortowanych rosnąco oraz liczba x**.**
Czy istnieją takie a ∈ A i b ∈ B takie, że x = a+b?

Ćwiczenie 3.4.

Danych jest n cyfr wylosowanych z przedziału [a, b], gdzie n jest liczbą **naturalną** mniejszą od 20, a i b są cyframi. Napisz program o złożoności liniowej, który wypisze wszystkie cyfry występujące najczęściej wśród wylosowanych.

Przykład:

Dane: n=10 , a=2, b=5, wylosowane cyfry: 2 3 4 5 4 3 4 3 4 3

Wynik: 3, 4

Ćwiczenie 3.5.

Aby zagrać w totolotka, musimy wybrać losowo 6 liczb z 49. Chcielibyśmy wybrać te losowe liczby w dokładnie sześciu losowaniach, mając pewność, że kolejno losowana liczba nie powtórzy się. Napisz program, który tak postąpi.

Wskazówka: Jednym ze sposobów jest utworzenie 49-elementowej tablicy z kolejnymi liczbami od 1 do 49, zamiana pierwszy raz wylosowanej liczby z liczbą z ostatnią w tablicy, drugiej z przedostatnią, itd. i za każdym razem skracanie zakresu losowania o jeden. Napisz program, który wylosuje w ten sposób sześć różnych liczb.

Ćwiczenie 3.6.

Danych jest n liczb wylosowanych z przedziału [a, b], gdzie n jest liczbą naturalną mniejszą od 20, a i b są liczbami naturalnymi nie większymi niż 100. Dana jest również liczba naturalna k <= n. Napisz program, który wypisze k w kolejności od końca największą liczbę wśród wylosowanych liczb.

Przykład:

Dane: n=10 , a=2, b=25, wylosowane liczby: 21, 10, 14, 15, 14, 13, 4, 23, 2, 3, k=4

Wynik: 14

Ćwiczenie 3.7.

W każdym z dziesięciu wierszy pliku slowa.txt znajduje się jedno słowo zapisane małymi literami alfabetu angielskiego.

a) Posortuj te słowa alfabetycznie i w takiej kolejności zapisz je do pliku posortowane.txt po jednym słowie w każdym wierszu.
b) Przepisz słowa z pliku posortowane.txt do pliku unikatowe.txt bez powtórzeń.

Ćwiczenie 3.8.

W pliku tekst.txt znajduje się tekst utworzony z małych liter alfabetu angielskiego.

a) Napisz funkcję, która zliczy wystąpienia poszczególnych liter w tym tekście.
b) Wykorzystaj tę funkcję do znalezienia wszystkich liter, które wystąpiły najczęściej w tekście. Wypisz wszystkie te litery na ekranie, oddzielając je spacją.

Ćwiczenie 3.9.

Dane są dwa pliki tekstowe: dane1.txt i dane2.txt. Każdy z nich zawiera ciąg liczb naturalnych z przedziału [0,100] uporządkowanych niemalejąco. Każda liczba zapisana jest w oddzielnym wierszu pliku. Utwórz plik wynik.txt zawierający scalone ciągi z obydwu plików. Uporządkowane liczby zapisz w jednym wierszu po spacji.

Ćwiczenie 3.10.

Dana jest liczba naturalna n, ciąg n liczb naturalnych A=(a_0, a_1,… ,a_{n-1}) oraz liczba naturalna x. Zastanów się, jak sprawdzić, czy w A istnieje podciąg złożony ze stojących obok siebie liczb o sumie x. Utwórz program, który jeśli odpowiedź jest twierdząca, poda indeks początkowy i końcowy tego pociągu, w przeciwnym razie wypisze komunikat: BRAK.

Ćwiczenie 3.11.

Napisz program z funkcją rekurencyjną generującą n-te słowo Fibonacciego

$$\text{Fibs(m)}=\begin{cases} \text{"b"} & \text{gdy m = 0} \\ \text{"a"} & \text{gdy } m = 1 \\ Fibs(m-1)+Fibs(m-2) & \text{gdy } m > 1 \end{cases}$$

Wypisz liczbę wystąpień każdej z liter w tym słowie.

Ćwiczenie 3.12.

Szukanie minimum wśród n liczb wymaga n-1 porównań liczb. Jeśli chcielibyśmy tą metodą znaleźć jednocześnie minimum i maksimum, to wykonamy 2*(n-1)=2*n-2 porównań.

Danych jest n losowych liczb naturalnych. Napisz program, który znajdzie minimalną i maksymalną wśród tych liczb, wykonując tylko 3/2*n-2 porównań liczb.

MATERIAŁY DO ĆWICZEŃ
MODUŁ 4

Autor: dr Anna Beata Kwiatkowska

Temat: Abstrakcyjny model problemu z grafami w tle

Skorzystaj z materiału metodycznego zawartego w prezentacji do modułu 4. i wykonaj następujące ćwiczenia:

Ćwiczenie 4.1.

Utwórz słownik zawierający zdania o trzech ważnych rzekach oraz krajach, przez które one przepływają. Para-klucz, wartość: Nil – Egipt

a) Wykorzystaj pętlę do wyświetlenia zdania o każdej rzece, na przykład: Nil przepływa przez Egipt.
b) Wykorzystaj pętlę do wyświetlenia nazw wszystkich rzek przechowywanych w słowniku.
c) Wykorzystaj pętlę do wyświetlenia nazw wszystkich państw przechowywanych w słowniku.
d) Poproś użytkownika, aby podał nazwę rzeki. Jeśli nazwy tej rzeki nie ma w słowniku, poproś o podanie nazwy kraju, a potem parę tę dopisz do słownika. Wyświetl cały słownik.

Ćwiczenie 4.2.

Wykorzystaj słownik do przechowania liczb ulubionych przez twoich znajomych z grupy. Użyj imion osób jako kluczy. Dla każdej osoby przechowaj w słowniku listę ulubionych liczb.

a) Wyświetl imiona wszystkich osób i ich ulubione liczby.
b) Znajdź liczbę/y najczęściej wybierane jako ulubione.

Ćwiczenie 4.3.

Wykorzystaj słownik do przechowywania informacji o uczniu. W słowniku powinny znaleźć się informacje takie jak: imię, nazwisko, wiek, miasto. Utwórz listę n uczniów.

a) Wyświetl wszystkie informacje przechowywane w słowniku (klucze i wartości).
b) Wyświetl nazwiska uczniów o wybranym imieniu.
c) Oblicz średni wiek ucznia.

Ćwiczenie 4.4.

d) Dany jest ciąg n losowych cyfr, n<50. Napisz program liniowy, który wypisze długość (liczbę wyrazów) najdłuższego podciągu spójnego, złożonego ze stojących obok siebie w tym ciągu takich samych cyfr (5 punktów).
e) Przykład: n=14 i ciąg liczb: 2 4 4 4 4 4 7 2 3 3 5 2 2 2. Wynikiem jest 5.

Ćwiczenie 4.5.

W pliku slowa.txt znajdują się słowa napisane w języku angielskim, po jednym w każdym wierszu. Utwórz słownik list, w którym na każdej liście będą znajdowały się słowa zaczynające się od takiej samej litery. Wypisz zawartość tego słownika na ekranie. Znajdź i wypisz na ekranie literę, na którą zaczyna się najwięcej słów. Jeśli takich jest liter więcej, wypisz wszystkie.

Ćwiczenie 4.6.

W pliku liczby.txt znajdują się liczby całkowite, po jednej w każdym wierszu. Utwórz słownik list, w którym na każdej liście będą znajdowały się liczby kończące się na tę samą cyfrę. Wypisz zawartość tego słownika na ekranie. Znajdź i wypisz na ekranie wszystkie cyfry, na które nie kończy się żadna liczba.

Ćwiczenie 4.7.

Napisz program do nauki nazw cyfr w języku angielskim (lub innym). Utwórz listę słowników (po jednym słowniku dla każdej cyfry. W każdym słowniku przyjmij następujące klucze: „cyfra”, „angielski”, „czy umiem” i nadaj im odpowiednie wartości. Początkowa wartość klucza „czy umiem” dla wszystkich cyfr powinna być równa „nie”. Program powinien uczyć cię tak długo, aż podasz wszystkie prawidłowe nazwy cyfr w języku angielskim, czyli wartość klucza dla wszystkich cyfr będzie równa „tak”.

Ćwiczenie 4.8.

Dany jest graf o n wierzchołkach i m krawędziach (n, m<10).

a) W trakcie wczytywania danych oblicz stopnie wierzchołów tego grafu.
b) Znajdź wierzchołek o najwyższym stopniu, jeśli jest takich więcej, wypisz wszystkie.
c) Wyświetl na ekranie macierz sąsiedztwa oraz listę sąsiadów tego grafu. Skorzystaj z przykładu podanego w prezentacji i porównaj wyniki.

Ćwiczenie 4.9.

Napisz dwie funkcje dla sprawdzenia, czy graf o n wierzchołkach i m krawędziach:

a) ma cykl Eulera,
b) ma drogę Eulera.

Ćwiczenie 4.10.

Utwórz program realizujący algorytm sortowania topologicznego opisany w prezentacji.

vademecum.ore.edu.pl