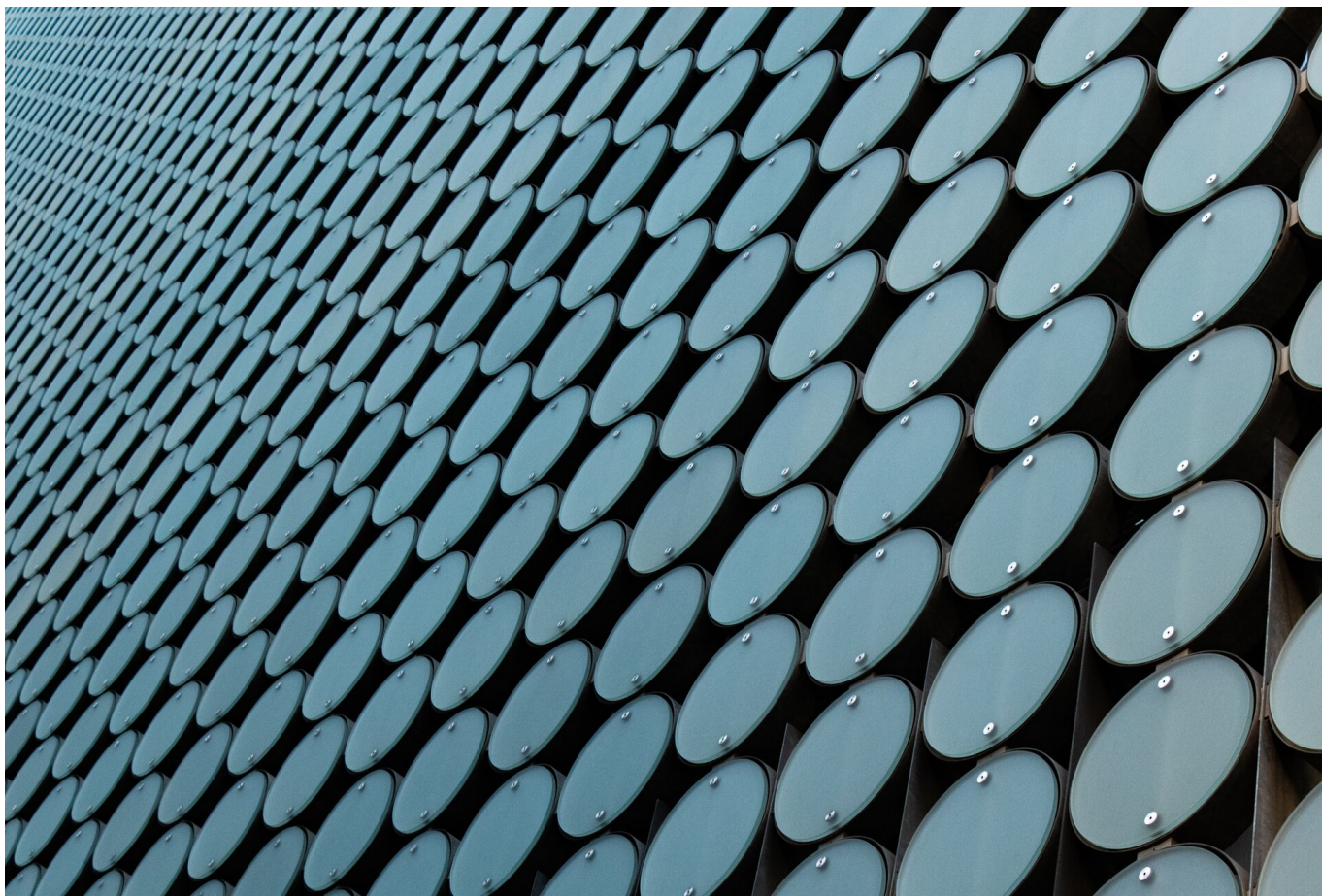




Analiza podejścia rekurencyjnego i iteracyjnego – zadania maturalne

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Prezentacja multimedialna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Analiza podejścia rekurencyjnego i iteracyjnego – zadania maturalne

Źródło: Mitchell Luo, domena publiczna.

Wiesz już, czym charakteryzuje się podejście rekurencyjne, a czym iteracyjne (zagadnienie to zostało szczegółowo omówione w e-materiale [Analiza podejścia rekurencyjnego i iteracyjnego](#)). W tym e-materiale rozwiążemy przykładowe zadania maturalne, w których w praktyce przećwiczymy dobieranie odpowiedniego algorytmu do analizowanego problemu.

O tym, jak zagadnienie rekurencji wyjaśnia matematyka, przeczytasz w e-materiałach:

- [Ciąg określony rekurencyjnie](#),
- [Ciąg geometryczny określony rekurencyjnie](#),
- [Wzór ogólny ciągu określonego rekurencyjnie](#),
- [Ciąg arytmetyczny określony wzorem rekurencyjnym](#).

Analizę algorytmów iteracyjnych i rekurencyjnych w wybranych językach programowania znajdziesz w e-materiałach:

- [Analiza podejścia rekurencyjnego i iteracyjnego w języku C++](#),
- [Analiza podejścia rekurencyjnego i iteracyjnego w języku Java](#),
- [Analiza podejścia rekurencyjnego i iteracyjnego w języku Python](#).

Twoje cele

- Scharakteryzujesz różnice między podejściem iteracyjnym a rekurencyjnym.
- Przeanalizujesz sposób rozwiązywania typowych zadań maturalnych wykorzystujących zarówno rekurencję, jak i iterację.
- Rozwiążesz samodzielnie kilka zadań maturalnych wymagających zastosowania technik rekurencyjnych i iteracyjnych.

Przeczytaj

Zadanie 1. Liczby Fibonacciego

Liczby Fibonacciego są definiowane w następujący sposób:

$$F_1 = 1, F_2 = 1,$$

$$F_n = F_{n-1} + F_{n-2} \text{ dla } n = 3, 4, \dots$$

Rekurencyjny algorytm, który służy do obliczania wartości F_n dla dowolnego $n \geq 1$, można zapisać następująco:

```
1 funkcja F(n)
2     jeśli n = 1 lub n = 2:
3         wynikiem jest 1
4     w przeciwnym razie:
5         wynikiem jest F(n - 1) + F(n - 2)
```

Zadanie zostało opracowane przez CKE i pojawiło się na I części egzaminu maturalnego z informatyki w czerwcu 2018 roku (poziom rozszerzony, egzamin w tzw. starej formule). Cały arkusz można znaleźć na stronie internetowej CKE.

Zadanie 1.1

Zapisz w wybranej przez siebie notacji (w języku programowania lub w pseudokodzie) algorytm iteracyjny, który służy do obliczania wartości liczby F_n dla dowolnego $n \geq 1$. Algorytm nie może używać tablic.

Rozwiązanie przedstawimy przy użyciu pseudokodu ze względu na fakt, że zadania maturalne dotyczące programowania można rozwiązać w wybranym języku programowania: C++, Java lub Python.

Praca domowa

Zaimplementuj przedstawione rozwiązanie w wybranym języku programowania: C++, Java, Python.

Przykładowe rozwiązanie:

Mamy przedstawić algorytm, który będzie obliczać wartość elementu ciągu Fibonacciego o podanym indeksie. Posłużymy się techniką iteracyjną (czyli wykonamy określoną liczbę powtórzeń pewnych czynności).

1. Definiujemy zmienną `indeks`, która będzie przechowywać indeks wyliczanego elementu ciągu:

```
1 indeks ← n
```

2. Piszemy instrukcję warunkową umożliwiającą obliczenie wartości pierwszego lub drugiego elementu ciągu Fibonacciego. W obu przypadkach podajemy wówczas wartość 1:

```
1 indeks ← n
2
3 jeżeli indeks = 1 lub indeks = 2:
4     wypisz(1)
5 w przeciwnym razie:
```

3. Aby obliczyć wartość elementu o indeksie numer 3 lub wyższym, stosujemy [iterację](#).
Tworzymy najpierw trzy zmienne:

- `aktualna` – przechowuje wartość wyliczoną dla obecnego indeksu,
- `poprzednia` – przechowuje wartość elementu o indeksie niższym o 1,
- `bPoprzednia` – przechowuje wartość elementu o indeksie niższym o 2.

```
1 indeks ← n
2
3 jeżeli indeks = 1 lub indeks = 2:
4     wypisz(1)
5 w przeciwnym razie:
6     aktualna ← 0
7     poprzednia ← 1
8     bPoprzednia ← 1
```

4. Tworzymy pętlę dla, w której będą obliczane wartości kolejnych elementów ciągu Fibonacciego (licznik pętli inicjujemy wartością 3, ponieważ dwa pierwsze elementy ciągu już znamy):

```
1 indeks ← n
2
3 jeżeli indeks = 1 LUB indeks = 2:
4     wypisz(1)
```

```

5 w przeciwnym razie:
6     aktualna ← 0
7     poprzednia ← 1
8     bPoprzednia ← 1
9
10    dla i = 3, 4, ..., indeks wykonuj:

```

5. Wewnątrz pętli wykonane zostaną trzy działania:

1. obliczenie wartości elementu ciągu o aktualnym indeksie (suma dwóch poprzednich elementów);
2. wartość o indeksie niższym o 1 na potrzeby następnego cyklu pętli stanie się wartością o indeksie niższym o 2;
3. wartość aktualna na potrzeby następnego cyklu pętli stanie się wartością o indeksie niższym o 1.

```

1 indeks ← n
2
3 jeżeli indeks = 1 LUB indeks = 2:
4     wypisz(1)
5 w przeciwnym razie:
6     aktualna ← 0
7     poprzednia ← 1
8     bPoprzednia ← 1
9
10    dla i = 3, 4, ..., indeks wykonuj:
11        aktualna ← poprzednia + bPoprzednia
12        bPoprzednia ← poprzednia
13        poprzednia ← aktualna
14
15    wypisz(aktualna)

```

Schemat oceniania:

- **2 p.** – za poprawny algorytm (w tym za prawidłowe zapisanie warunków początkowych i iteracji 1 p.).
- **1 p.** – za zapamiętanie tylko dwóch poprzednich wyrazów ciągu.
- **0 p.** – za podanie odpowiedzi błędnej albo brak odpowiedzi.

Za przedstawione rozwiązanie zadania otrzymalibyśmy 2 punkty, ponieważ prawidłowo zapisane zostały warunki początkowe (dla indeksów 1 i 2) oraz pętla iteracyjna.

Zadanie 1.2

Aby obliczyć F_{45} , wywołano najpierw funkcję iteracyjną, a potem – rekurencyjną. Okazało się, że czas trwania obliczeń realizowanych przez funkcję rekurencyjną był długi, podczas gdy funkcja iteracyjna prawie natychmiast podała wynik. Uzasadnij długi czas działania funkcji **rekurencyjnej**.

Przykładowe rozwiązanie:

Jedną z lepszych wizualizacji rekurencji jest zapisanie jej w postaci drzewa, w którym poszczególne wywołania funkcji uruchamiają kolejne jej wywołania (aż do napotkania przypadku podstawowego). Wielokrotne wywoływanie powoduje wielokrotne obliczanie tych samych wartości. Poszczególne elementy ciągu Fibonacciego wyliczane są zatem bez brania pod uwagę wyrazów już znanych. Za każdym razem funkcja rekurencyjna dochodzi do przypadku podstawowego. Ponadto im dalszy wyraz ciągu Fibonacciego ma zostać obliczony, tym więcej powtórzeń jest wykonywanych. Z tego powodu taka technika okazuje się wolniejsza od metody iteracyjnej i daje złożoność wykładniczą od n .

Schemat oceniania:

- **1 p.** – za poprawną odpowiedź.
- **0 p.** – za podanie odpowiedzi błędnej albo brak odpowiedzi.

Nasza odpowiedź jest trochę obszerniejsza niż ta przedstawiona w arkuszu sporządzonym przez CKE, ale wyjaśnia powód wolniejszego działania algorytmu rekurencyjnego. W rezultacie na egzaminie otrzymalibyśmy 1 punkt.

Słownik

iteracja

technika programowania polegająca na wielokrotnym powtarzaniu zapisanego ciągu instrukcji aż do spełnienia określonego warunku

rekurencja

technika programowania, w której funkcja wywołuje samą siebie aż do napotkania przypadku podstawowego, czyli takiego, którego nie da się rozłożyć na mniejsze problemy; przykład przypadku podstawowego dla obliczania piątego wyrazu ciągu Fibonacciego:

$$\begin{aligned} fib(4) &= fib(3) + fib(2) = (fib(2) + fib(1)) + (fib(1) + fib(0)) = \\ &= ((fib(1) + fib(0)) + fib(1)) + fib(1) = (fib(1) + fib(0)) + fib(1) + fib(1) = \\ &= ((1 + 0) + 1) + (1 + 0) = 3 \end{aligned}$$

przypadek podstawowy

wartość, dla której funkcja rekurencyjna zwraca określony wynik

Prezentacja multimedialna

Zadanie 1.3

Aby przyspieszyć rekurencyjne obliczanie wartości n -tego wyrazu ciągu Fibonacciego, można skorzystać z następujących wzorów, prawdziwych dla dowolnego całkowitego $k \geq 2$:

$$F_{2k} = (F_{k+1})^2 - (F_{k-1})^2$$

$$F_{2k-1} = (F_k)^2 + (F_{k-1})^2$$

Zapisz w wybranej przez siebie notacji (w postaci listy kroków, w języku programowania lub w pseudokodzie) algorytm **rekurencyjny**, który służy do obliczania wartości liczby F_n dla dowolnego $n \geq 1$ i korzysta z tych wzorów.

Zadanie zostało opracowane przez CKE i pojawiło się na I części egzaminu maturalnego z informatyki w czerwcu 2018 roku (poziom rozszerzony, egzamin w tzw. starej formule). Cały arkusz można znaleźć na stronie internetowej CKE.

Polecenie 1

Przedstaw rozwiązanie zadania w postaci programu w języku C++, Java lub Python.

Polecenie 2

Porównaj swoje rozwiązanie z przedstawionym w prezentacji.

Rozwiązanie



Więcej zadań maturalnych znajdziesz w zbiorze dostępnym na stronie internetowej Okręgowej Komisji Egzaminacyjnej w Warszawie.
Źródło: OKE, oke.waw.pl, tylko do użytku edukacyjnego.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PQ0NR5wh>

To zadanie jest nieznacznie trudniejsze od wcześniejszych, ponieważ wymaga użycia przedstawionych w arkuszu wzorów, z których wiele osób zapewne jeszcze nie korzystało.

2

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PQ0NR5wh>

Najpierw definiujemy własną funkcję obliczającą kwadrat pewnej liczby. Będzie ona potrzebna do wyznaczania poszczególnych elementów ciągu:

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PQ0NR5wh>

Następnie zdefiniujemy funkcję rekurencyjną `fibonacci()`, która będzie obliczać wartość wybranego elementu ciągu:

4

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PQ0NR5wh>

Pierwszymi składnikami funkcji `fibonacci()` będą zmienne `i` oraz `k`. Są one potrzebne do modyfikowania indeksu tak, aby zgadzał się ze wzorami:

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PQ0NR5wh>

Podajemy dwa przypadki, które powodują zwrócenie przez funkcję wartości 1. Chodzi oczywiście o obliczenie dwóch pierwszych elementów ciągu Fibonacciego:

6

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PQ0NR5wh>

Następnie ustalimy warunek, który pozwoli wybrać wersję wzoru używanego do wyliczania elementu ciągu (jeden wariant jest wykorzystywany w przypadku indeksów parzystych, a drugi nieparzystych). Jeśli indeks jest parzysty, używamy pierwszego wzoru:

Operator mod oznacza resztę z dzielenia, natomiast $\div$ dzielenie całkowitoliczbowe.

7

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PQ0NR5wh>

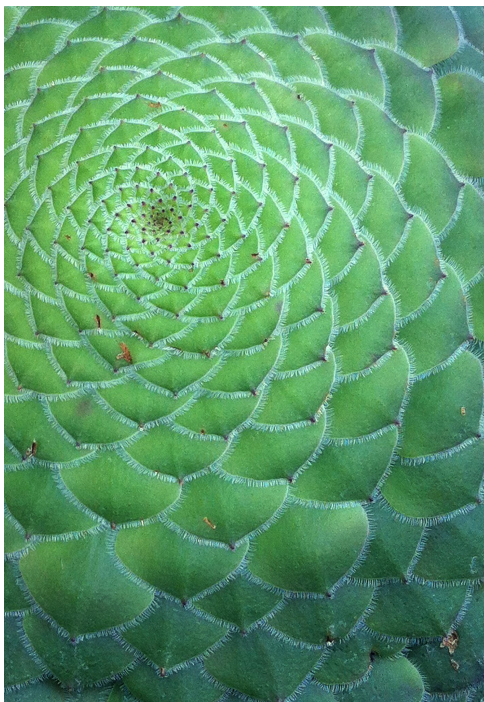
Dla elementów o indeksach nieparzystych stosujemy drugi wzór:

8

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PQ0NR5wh>

Skoro algorytm jest gotowy, zapoznaj się z zasadami przyznawania punktów na egzaminie i sprawdź, czy przedstawione rozwiązanie spełniło wszystkie kryteria.



Ciąg Fibonacciego można odnaleźć w wielu otaczających nas elementach świata przyrody.
Źródło: Max Ronnersjö, p.wikipedia.org, CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PQ0NR5wh>

Schemat punktowania

- **2 pkt** – za poprawną odpowiedź, w tym:
 - **1 pkt** – za prawidłowe zdefiniowanie warunków;
 - **1 pkt** – za prawidłowe wywołania rekurencyjne;
 - **0 pkt** – za podanie odpowiedzi błędnej albo brak odpowiedzi.

Nasze rozwiązanie jest bardzo podobne do przykładowego, możemy więc stwierdzić, że otrzymalibyśmy maksymalną liczbę punktów. Prawidłowo zdefiniowaliśmy warunki użycia poszczególnych wzorów oraz wywołania funkcji rekurencyjnej.

Sprawdź się

Zadanie 2

Ciąg Fibonacciego to ciąg liczb naturalnych określony rekurencyjnie w sposób następujący: $F_1 = 1$, $F_2 = 1$, a każdy następny element ciągu jest sumą dwóch poprzednich, czyli:

$$F_n = \begin{cases} 0 & \text{dla } n = 0 \\ 1 & \text{dla } n = 1 \\ F_{n-1} + F_{n-2} & \text{dla } n > 1 \end{cases}$$

Binarny fraktal Fibonacciego to dwuwymiarowa tablica zawierająca w kolejnych wierszach binarne zapisy kolejnych liczb Fibonacciego, gdzie każde zero w zapisie zastąpiono białym kwadratem, a każdą jedynkę czarnym kwadratem (p. rysunek). Wszystkie binarne zapisy powinny składać się z jednakowej liczby cyfr, czyli do zapisów krótszych niż najdłuższy należy dodać zera wiodące.

n	F_n	zapis binarny F_n	binarny fraktal Fibonacciego
1	1	000001	
2	1	000001	
3	2	000010	
4	3	000011	
5	5	000101	
6	8	001000	
7	13	001101	
8	21	010101	
9	34	100010	
10	55	110111	

Napisz program, który znajdzie wszystkie liczby pierwsze wśród liczb $F_1, F_2, \dots, F_{40}$. Zapisz każdą z liczb w osobnym wierszu pliku `pierwsze.txt`.

Do oceny oddajesz:

- plik `pierwsze.txt` zawierający odpowiedź (wszystkie liczby pierwsze wśród liczb $F_1, F_2, \dots, F_{40}$, każda z nich zapisana w osobnym wierszu);
- plik(i) z komputerową realizacją zadania (kodem programu).

Zadanie zostało opracowane przez CKE i pojawiło się w zbiorze zadań maturalnych z informatyki. Cały zbiór można znaleźć na stronie internetowej Centralnej Komisji

Egzaminacyjnej.

Przetestuj działanie programu dla danych z treści zadania. Twój program powinien wypisywać rozwiązanie na ekran, a nie zapisywać je do pliku.

Pokaż ćwiczenia:   



JĘZYK C++

Twoje zadania

1. Program w języku C++ wypisuje wszystkie liczby ciągu Fibonacciego od 1 do 40 wyrazu, które są liczbami pierwszymi, każda w osobnej linii.





JĘZYK JAVA

Twoje zadania

1. Program w języku Java wypisuje wszystkie liczby ciągu Fibonacciego od 1 do 40 wyrazu, które są liczbami pierwszymi, każda w osobnej linii.





JĘZYK PYTHON

Twoje zadania

1. Program w języku Python wypisuje wszystkie liczby ciągu Fibonacciego od 1 do 40 wyrazu, które są liczbami pierwszymi, każda w osobnej linii.

Odpowiedź do zadania

pierwsze.txt

Plik o rozmiarze 33.00 B w języku polskim

Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Analiza podejścia rekurencyjnego i iteracyjnego – zadania maturalne

Grupa docelowa:

Szkoła ponadpodstawowa, liceum ogólnokształcące, technikum, zakres podstawowy i rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres podstawowy. Uczeń:

2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy:

e) obliczania wartości elementów ciągu metodą iteracyjną i rekurencyjną, w tym wartości elementów ciągu Fibonacciego.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

2) do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych;

3) objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

3) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

I + II. Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

3) objaśnia, a także porównuje podstawowe metody i techniki algorytmiczne oraz struktury danych, wykorzystując przy tym przykłady problemów i algorytmów, w szczególności:

b) rekurencję (do generowania ciągów liczb, potęgowania, sortowania liczb, generowania fraktali),

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Scharakteryzujesz różnice między podejściem iteracyjnym a rekurencyjnym.
- Przeanalizujesz sposób rozwiązywania typowych zadań maturalnych wykorzystujących zarówno rekurencję, jak i iterację.
- Rozwiążesz samodzielnie kilka zadań maturalnych wymagających zastosowania technik rekurencyjnych i iteracyjnych.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- ćwiczenia praktyczne.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- oprogramowanie dla języka C++, w tym kompilator GCC/G++ 4.5 (lub nowszej wersji) i Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio;
- oprogramowanie dla języka Java SE 8 (lub nowszej wersji), w tym Eclipse 4.4 (lub nowszej wersji);
- oprogramowanie dla języka Python 3 (lub nowszej wersji), w tym PyCharm lub IDLE.

Przebieg lekcji

Przed lekcją:

1. Nauczyciel prosi uczniów o powtórzenie najważniejszych informacji dotyczących rekurencji oraz iteracji.

Faza wstępna:

1. W ramach sprawdzenia przygotowania uczniów do lekcji nauczyciel prosi wybrane osoby o przypomnienie najważniejszych informacji dotyczących rekurencji oraz iteracji.
2. Nauczyciel inicjuje rozmowę wprowadzającą w temat lekcji. Przedstawia cele zajęć oraz kryteria sukcesu.

Faza realizacyjna:

1. **Praca z tekstem.** Nauczyciel prosi uczniów, aby w parach przeanalizowali rozwiązania Zadania 1.1 oraz Zadania 1.2 zaprezentowane w sekcji „Przeczytaj”. Następnie wyjaśnia, jakie typy zadań mogą pojawić się na egzaminie maturalnym.
2. **Praca z multimediami.** Uczniowie indywidualnie rozwiązują w wybranym języku programowania Zadanie 1.3 z sekcji „Prezentacja multimedialna”. Następnie porównują swoje rozwiązanie z przedstawionym w prezentacji.
3. **Ćwiczenie umiejętności.** Uczniowie w parach wykonują Zadanie 2 z sekcji „Sprawdź się” w wybranym języku programowania. Następnie porównują swoje rozwiązanie z inną grupą uczniów.

Faza podsumowująca:

1. Nauczyciel wyświetla na tablicy temat lekcji i cele zawarte w sekcji „Wprowadzenie”. W kontekście ich realizacji podsumowuje przebieg zajęć, a także wskazuje mocne i słabe strony pracy uczniów.
2. Nauczyciel prosi uczniów o podsumowanie zgromadzonej wiedzy w zakresie programowania.

Praca domowa:

1. Uczniowie wykonują pracę domową z sekcji „Przeczytaj”.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka C++.
- Oficjalna dokumentacja techniczna dla kompilatora GCC/G++ 4.5 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio.
- Oficjalna dokumentacja techniczna dla języka Java SE 8 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Eclipse 4.4 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla języka Python 3 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania PyCharm lub IDLE.

Wskazówki metodyczne:

- E-materiał może zostać wykorzystany do powtórzenia wiadomości na temat ciągu Fibonacciego.