



Wprowadzenie do algorytmów iteracyjnych

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Schemat interaktywny](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Wprowadzenie do algorytmów iteracyjnych

Źródło: Rodion Kutsaev, dostępny w internecie: unsplash.com, domena publiczna.

Dzięki informacjom zdobytym w szkole podstawowej potrafisz dostrzec w życiu codziennym przykłady prostych algorytmów iteracyjnych – np. jedzenia kaszki czy słodzenia herbaty itp. W e-materiałach dla szkoły średniej zapoznasz się z trudniejszymi przykładami algorytmów iteracyjnych i wykorzystasz je do rozwiązywania wielu stawianych przed tobą problemów, np. matematycznych.

W tym e-materiale przypomnimy informacje o algorytmach iteracyjnych oraz omówimy sposoby ich zapisu.

Więcej informacji o algorytmach iteracyjnych znajdziesz w e-materiałach:

- [Algorytmy liczbowe](#),
- [Algorytmy iteracyjne – obliczanie silni](#),
- [Algorytmy iteracyjne i liczbowe – potęgowanie liczb](#).

Więcej zadań? Sięgnij do e-materiału [Algorytmy iteracyjne i liczbowe – zadania maturalne](#).

Twoje cele

- Wyjaśnisz, czym są algorytmy iteracyjne i kiedy je stosujemy.
- Opracujesz schemat blokowy i pseudokod algorytmu iteracyjnego.
- Przeanalizujesz rozwiązania zadań, w których należy wykorzystać iterację.

Przeczytaj

Różne typy algorytmów

Utrwalmy podstawowe informacje na temat algorytmów liniowych i rozgałęzionych.

Już wiesz

Algorytm liniowy:

- występuje jeden możliwy ciąg instrukcji,
- czynności wykonywane są jedna po drugiej,
- nie zawiera instrukcji warunkowych.

Algorytm rozgałęziony:

- algorytm z warunkami,
- kolejna instrukcja w algorytmie może zależeć od spełnienia bądź niespełnienia określonego warunku,
- istnieje kilka alternatywnych ciągów instrukcji,
- schemat blokowy algorytmu tego typu zawiera bloki warunkowe w postaci rombu z dwoma wyjściami.

Wspomnieliśmy o prostym algorytmie słodzenia herbaty. Przypomnijmy sobie jego budowę przedstawioną za pomocą schematu blokowego.



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

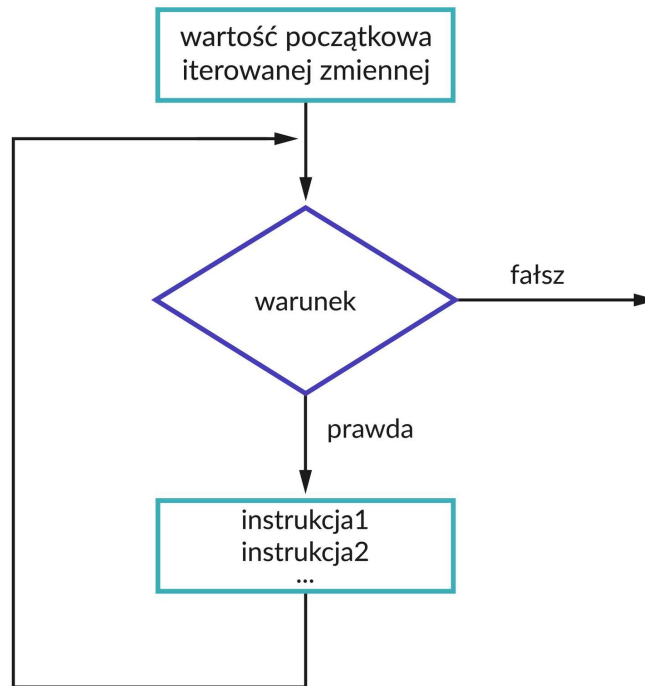
Ten sam ciąg instrukcji powtarzany jest kilkakrotnie. Wielokrotne powtarzanie tych samych czynności nazywamy [iteracją](#).

Problem słodzenia herbaty jest mało skomplikowany, zastosowanie iteracji nie wpływa znacząco na jego czytelność. Może natomiast skrócić zapis (tak w przypadku pseudokodu, schematu blokowego, jak i języków programowania).

Już wiesz

W algorytmie iteracyjnym wykorzystujemy bloki warunkowe występujące w algorytmach rozgałęzionych. Kolejna instrukcja zależy od spełnienia bądź niespełnienia określonego warunku. Oprócz warunku, w pewnym momencie wykonywania algorytmu nastąpi powrót do jednej z wcześniejszych instrukcji. Taki powrót nazywamy [pętlą](#).

Schemat blokowy algorytmu iteracyjnego

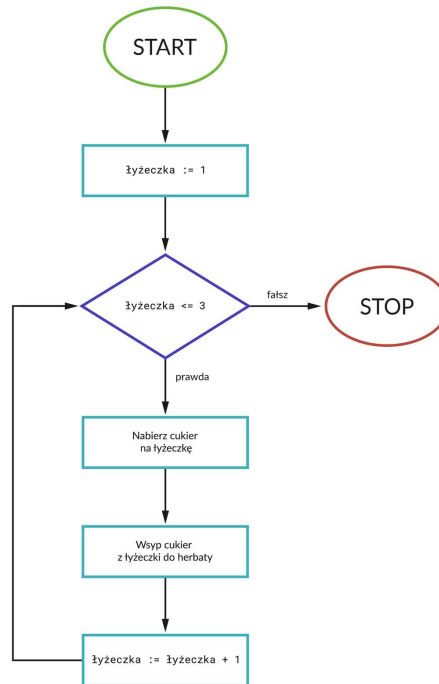


Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

W pierwszym bloku operacyjnym zainicjowana zostaje wartość początkowa iterowanej zmiennej (np. posłodzenie herbaty pierwszą łyżeczką cukru). Dzięki temu możemy zliczać liczbę iteracji. Następnie sprawdzany jest warunek. Jeżeli jest on fałszywy, wychodzimy z pętli i nie powtarzamy więcej operacji w niej zawartych. Natomiast jeżeli warunek będzie spełniony, wykonujemy ciąg odpowiednich instrukcji. Po wykonaniu ciągu instrukcji wracamy do sprawdzenia warunku. Jeżeli jest spełniony, ponownie wykonujemy te same instrukcje itd.

Polecenie 1

Przeanalizuj schemat blokowy algorytmu słodzenia herbaty wykorzystujący pętlę.



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Algorytm iteracyjny wypisujący liczby z podanego przedziału

Za pomocą pseudokodu zapiszemy [algorytm iteracyjny](#) wyświetlający na ekranie liczby całkowite z zakresu $\langle 10, 25 \rangle$.

Pseudokod algorytmu iteracyjnego możemy zrealizować za pomocą instrukcji iteracyjnych. Pętlę utworzymy z wykorzystaniem słów **dopóki** .. **wykonuj**. Jeżeli wyrażenie zawarte między **dopóki** i **wykonuj** będzie prawdziwe, będą wykonywane instrukcje wymienione bezpośrednio pod nimi. Instrukcje te zapiszemy z użyciem wcięcia. Ma to na celu wskazanie, które instrukcje znajdują się w pętli.

```
1 dopóki wyrażenie wykonuj:
2     instrukcja1
3     instrukcja2
```

Zapiszmy w postaci pseudokodu algorytm wyświetlający na ekranie liczby całkowite z zakresu $\langle 10, 25 \rangle$.

```
1 i := 10
2 dopóki i <= 25 wykonuj:
```

```
3   wypisz i
4   i := i + 1
```

W pierwszym wierszu pseudokodu inicjujemy wartość początkową $i := 10$. To będzie pierwsza liczba wypisana przez algorytm. Następnie tworzymy pętlę, w której sprawdzane jest to, czy i jest mniejsze lub równe 25. Jeżeli warunek jest spełniony, wypisujemy wartość i , a następnie zwiększamy ją o 1, w celu wypisania kolejnej liczby całkowitej. Instrukcje będą wykonywane, dopóki i nie przekroczy wartości 25.

Słownik

algorytm iteracyjny

algorytm, w którym wykorzystywane są pętle w celu kilkakrotnego powtórzenia ciągu poleceń

iteracja

wielokrotne wykonywanie tych samych czynności

pętla

instrukcja iteracyjna, która umożliwia cykliczne wykonywanie ciągu instrukcji

Schemat interaktywny

Polecenie 1

Opracuj algorytm obliczania sumy kolejnych liczb naturalnych z zakresu $\langle 1, n \rangle$, gdzie n jest liczbą podaną przez użytkownika. Przedstaw go w postaci schematu blokowego lub listy kroków.

Specyfikacja problemu:

Dane:

- n – liczba całkowita dodatnia

Wynik:

- Na standardowym wyjściu zostaną wyświetlone kolejne liczby od 1 do n .

Schemat blokowy:

Lista kroków:

1

Polecenie 2

Porównaj swoje rozwiązanie z prezentacją.

Zadaniem do wykonania jest znalezienie sumy kolejnych liczb naturalnych z zakresu $\langle 1, n \rangle$, gdzie n jest liczbą podaną przez użytkownika. W przedstawionej prezentacji przećwiczymy pisanie algorytmów za pomocą listy kroków, schematu blokowego oraz pseudokodu.

1

2

Zacznijmy od napisania algorytmu za pomocą listy kroków. Pierwszymi krokami algorytmu będzie jego rozpoczęcie oraz wczytanie liczby n .

1. Rozpocznij algorytm.
2. Wczytaj liczbę n .

Następnie należy zainicjować wartość początkową zmiennej i wartością 1, ponieważ

3

taki jest lewy koniec przedziału liczb, których sumę chcemy obliczyć. Oprócz tego inicjujemy wartość zmiennej *suma* wartością 0. Do tej zmiennej będziemy zapisywać aktualną sumę liczb.

1. Rozpocznij algorytm.
2. Wczytaj liczbę *n*.
3. Zmiennej *i* przypisz wartość 1, a zmiennej *suma* wartość 0.

4

Kolejnym krokiem jest sprawdzenie, czy *i* jest mniejsze lub równe *n*, to znaczy czy aktualna liczba mieści się w zakresie podanym przez użytkownika. Jeżeli jest to prawda, możemy przejść do kolejnego kroku. Natomiast jeżeli warunek nie jest spełniony, należy wyjść z pętli, to znaczy przejść do kroku 7, w którym nastąpi wypisanie sumy.

1. Rozpocznij algorytm.
2. Wczytaj liczbę *n*.
3. Zmiennej *i* przypisz wartość 1, a zmiennej *suma* wartość 0.
4. Jeżeli $i \leq n$, przejdź do kroku 5, w przeciwnym wypadku przejdź do kroku 7.

Teraz należy aktualną liczbę naturalną *i* dodać do sumy wszystkich liczb i zwiększyć *i* o jeden w celu uzyskania kolejnej liczby naturalnej. Po wykonaniu ciągu tych dwóch instrukcji, za

5

pomocą pętli należy przejść do ponownego sprawdzenia warunku, to znaczy do kroku nr 4.

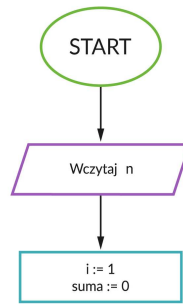
1. Rozpocznij algorytm.
2. Wczytaj liczbę n .
3. Zmiennej i przypisz wartość 1, a zmiennej suma wartość 0.
4. Jeżeli $i \leq n$ przejdź do kroku 5, w przeciwnym wypadku przejdź do kroku 7.
5. Do zmiennej suma dodaj i .
6. Zwiększ wartość zapisaną w zmiennej i o jeden i przejdź do kroku 4.

6

Siódmym krokiem jest wypisanie na ekranie wartości suma zawierającej sumę wszystkich liczb naturalnych z przedziału $\langle 1, n \rangle$. Ostatnim krokiem jest zakończenie algorytmu.

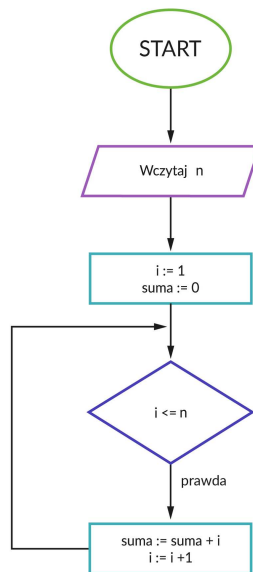
Oto gotowa lista kroków algorytmu:

1. Rozpocznij algorytm.
2. Wczytaj liczbę n .
3. Zmiennej i przypisz wartość 1, a zmiennej suma wartość 0.
4. Jeżeli $i \leq n$, przejdź do kroku 5, w przeciwnym wypadku przejdź do kroku 7.
5. Do zmiennej suma dodaj i .
6. Zwiększ wartość zapisaną w zmiennej i o jeden i przejdź do kroku 4.
7. Wypisz wartość zmiennej suma.
8. Zakończ algorytm.



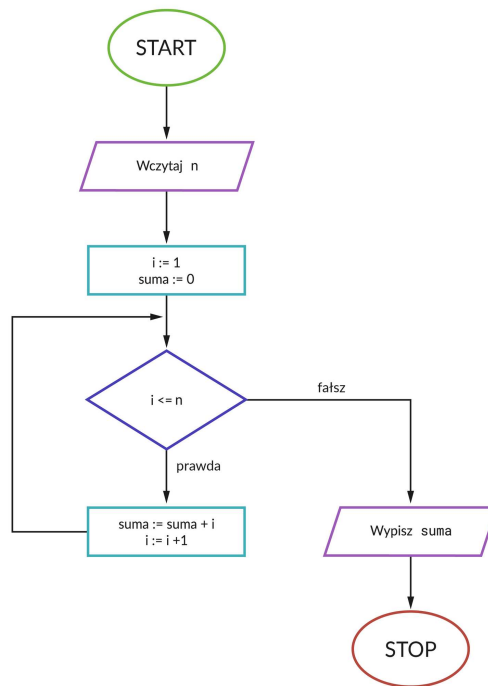
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Przejdźmy teraz do schematu blokowego algorytmu. Pierwsze trzy bloki schematu realizujemy z wykorzystaniem trzech pierwszych kroków zawartych w liście kroków. Rozpoczynamy algorytm blokiem START, wczytujemy n za pomocą bloku wyjścia/wejścia i nadajemy wartości początkowe zmiennym i oraz $suma$.



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Następnie sprawdzany jest warunek. Jeżeli jest spełniony, przechodzimy do bloku operacyjnego realizującego piąty i szósty krok z opracowanej już listy kroków. Po tym następuje powrót do wcześniejszych instrukcji, a konkretnie do bloku warunkowego.



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Jeżeli warunek nie jest spełniony, wypisujemy wartość zmiennej suma i kończymy algorytm.

10

Ostatnią częścią naszego zadania jest napisanie pseudokodu. Pseudokod został zrealizowany na podstawie schematu blokowego i listy kroków. Instrukcją iteracyjną będzie pętla dopóki . . . wykonuj. Dopóki wyrażenie $i \leq n$ będzie spełnione, będą wykonywane instrukcje zapisane bezpośrednio pod nim, ponadto zawierające wcięcia. W momencie, gdy wyrażenie nie będzie spełnione, program przejdzie do wykonania linijki siódmej pseudokodu, a następnie zakończy algorytm.

```

1 wczytaj n
2 i := 1
3 suma := 0
4 dopóki i <= n wykonuj:
5     suma := suma + i
6     i := i + 1
7 wypisz suma
  
```

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Polecenie 3

Zapisz algorytm w wybranym przez siebie języku programowania. Przetestuj jego działanie dla zakresu $\langle 13, 1120 \rangle$.

1

1

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1

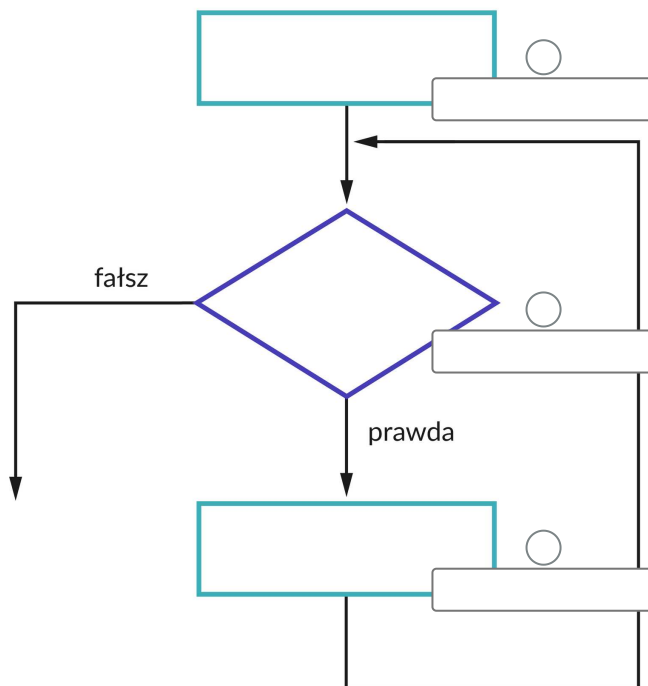


Zaznacz, które stwierdzenia są prawdziwe.

- ☐ Pętla występuje w algorytmach iteracyjnych.
- ☐ Pętla występuje w algorytmach liniowych.
- ☐ Pętla to instrukcja iteracyjna.
- ☐ Za pomocą pętli możemy powrócić do poprzednich kroków algorytmu.

Ćwiczenie 2

Uzupełnij schemat blokowy pętli.



warunek

instrukcje

inicjowanie wartości początkowej zmiennej

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Ćwiczenie 3



Uzupełnij zapisany za pomocą pseudokodu algorytm czytania książki, której liczba stron wynosi 121. Zaczynij od strony numer 1.

strona :=

dopóki 121 wykonuj:

Przeczytaj stronę o numerze "strona"

jeżeli strona mod 2 == 1 wykonaj:

Przerzuć kartkę

:= strona +

Ćwiczenie 4



Połącz w pary.

algorytm liniowy

wielokrotne powtarzanie ciągu instrukcji

algorytm iteracyjny

czynności wykonywane jedna po drugiej

iteracja

instrukcja iteracyjna

algorytm rozgałęziony

algorytm wykorzystujący możliwość powrotu do poprzednich instrukcji

pętla

wykorzystuje instrukcje warunkowe

Ćwiczenie 5



Uzupełnij listę kroków algorytmu wyświetlającego liczby całkowite z przedziału $\langle 8, 7 \rangle$ w kolejności malejącej.

1. Rozpocznij algorytm.

2. Zmiennej a przypisz wartość .

3. Jeżeli $- 8$, przejdź do kroku

, w wypadku przejdź do kroku .

4. Wypisz liczbę .

5. Wykonaj operację $:= a$
1, a następnie przejdź do kroku .

6. algorytm.

Ćwiczenie 6



Uporządkuj listę kroków algorytmu obliczającego sumę liczb naturalnych kończących się cyfrą 3 z przedziału $\langle 3, a \rangle$, gdzie a jest wartością podaną przez użytkownika.

Jeżeli $i \leq a$ przejdź do kroku 5, w przeciwnym wypadku przejdź do kroku 3.

Zwiększ wartość zapisaną w zmiennej i o 10 i przejdź do kroku 4.

Rozpocznij algorytm.

Wczytaj liczbę a .

Do zmiennej suma dodaj i .

Zakończ algorytm.

Wypisz liczbę suma .

Zmiennej i przypisz wartość 3, a zmiennej suma wartość 0.

Ćwiczenie 7



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Ćwiczenie 8



Zapisz algorytm wypisujący n wykrzykników w postaci pseudokodu. Dane pobierz od użytkownika.

Wykrzykniki powinny zapisywane być w kolejnych liniach.

Specyfikacja:

Dane:

n – dodatnia liczba całkowita

Wynik:

Na standardowym wyjściu zostanie wypisanych n wykrzykników.

1



Ćwiczenie 9



Zapisz algorytm wypisujący n wykrzykników w wybranym przez siebie języku programowania. Przetestuj jego działanie dla $n = 9$.

Specyfikacja:

Dane:

n – liczba naturalna

Wynik:

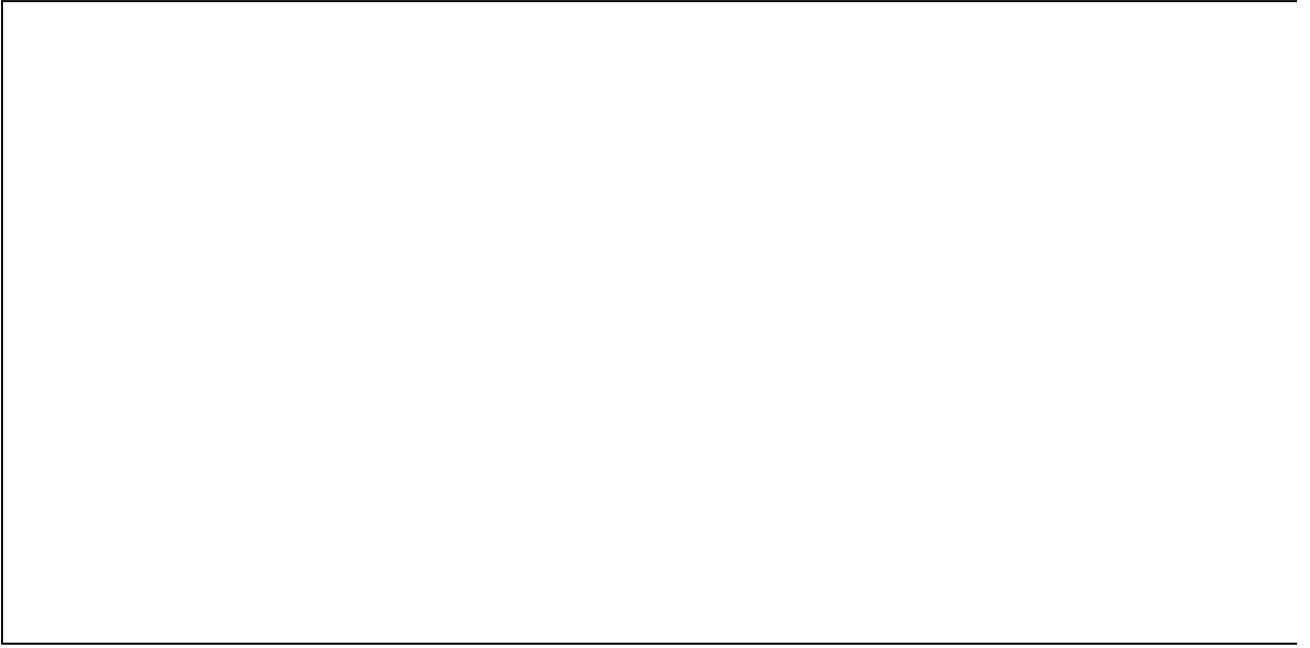
Na standardowym wyjściu zostanie wypisanych n wykrzykników.

Twoje zadania

1. Na standardowym wyjściu zostanie wypisanych n wykrzykników.

1

1



Dla nauczyciela

Autor: Zespół autorski Contentplus.pl sp. z o.o.

Przedmiot: Informatyka

Temat: Wprowadzenie do algorytmów iteracyjnych

Materiał jest powtórzeniem wiadomości ze szkoły podstawowej.

Grupa docelowa:

Liceum ogólnokształcące i technikum, Szkoła podstawowa IV-VIII, liceum ogólnokształcące, szkoła podstawowa, 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.
- III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi, w tym znajomość zasad działania urządzeń cyfrowych i sieci komputerowych oraz wykonywania obliczeń i programów.

Cele kształcenia – wymagania ogólne

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

Treści nauczania – wymagania szczegółowe

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres podstawowy. Uczeń:

1) planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania).

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

1) w zależności od problemu rozwiązuje go, stosując metodę wstępującą lub zstępującą;

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

Zakres podstawowy. Uczeń:

1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2);

Treści nauczania – wymagania szczegółowe

Klasy VII i VIII

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

1) formułuje problem w postaci specyfikacji (czyli opisuje dane i wyniki) i wyróżnia kroki w algorytmicznym rozwiązywaniu problemów. Stosuje różne

sposoby przedstawiania algorytmów, w tym w języku naturalnym, w postaci schematów blokowych, listy kroków;

2) stosuje przy rozwiązywaniu problemów podstawowe algorytmy:

a) na liczbach naturalnych: bada podzielność liczb, wyodrębnia cyfry danej liczby, przedstawia działanie algorytmu Euklidesa w obu wersjach iteracyjnych (z odejmowaniem i z resztą z dzielenia),

b) wyszukiwania i porządkowania: wyszukuje element w zbiorze uporządkowanym i nieuporządkowanym oraz porządkuje elementy w zbiorze metodą przez proste wybieranie i zliczanie;

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

1) projektuje, tworzy i testuje programy w procesie rozwiązywania problemów. W programach stosuje: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje oraz zmienne i tablice. W szczególności programuje algorytmy z działu I pkt 2;

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):

- Wyjaśnisz, czym są algorytmy iteracyjne i kiedy je stosujemy.
- Opracujesz schemat blokowy i pseudokod algorytmu iteracyjnego.
- Przeanalizujesz rozwiązania zadań, w których należy wykorzystać iterację.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimedium 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.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Wprowadzenie do algorytmów iteracyjnych”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Chętna lub wybrana osoba przedstawia najważniejsze informacje dotyczące algorytmów iteracyjnych.
2. Nauczyciel prosi wybraną osobę o odczytanie tematu lekcji, a następnie określa cele i kryteria sukcesu.

Faza realizacyjna:

1. **Praca z tekstem.** Jeżeli przygotowanie uczniów do lekcji jest niewystarczające, nauczyciel prosi o indywidualne zapoznanie się z treścią zawartą w sekcji „Przeczytaj”. Każdy uczestnik zajęć podczas cichego czytania wynotowuje najważniejsze kwestie poruszane w tekście.
2. **Praca z multimediami.** Nauczyciel wyświetla zawartość sekcji „Schemat interaktywny”. Uczniowie wspólnie zapoznają się z prezentacją przedstawiającą algorytm znajdowania sumy kolejnych liczb naturalnych w postaci listy kroków, pseudokodu oraz schematu blokowego. Następnie, na podstawie prezentacji, opracowują samodzielnie algorytm znajdowania iloczynu kolejnych liczb naturalnych.
3. **Ćwiczenie umiejętności.** Uczniowie realizują indywidualnie ćwiczenia nr 1-7, po ich wykonaniu porównują otrzymane wyniki z inną osobą.

Faza podsumowująca:

1. Nauczyciel zadaje pytania podsumowujące, np.
 - jaką nazwę nosi wielokrotne wykonywanie tych samych czynności?
 - do czego wykorzystujemy pętlę?
 - co to jest algorytm iteracyjny?
2. Na koniec zajęć nauczyciel prosi uczniów o rozwinięcie zdania: „Na dzisiejszych zajęciach nauczyłam/łem się jak...”.

Praca domowa:

1. Uczniowie wykonują ćwiczenie 8 z sekcji „Sprawdź się”. Następnie ten sam program zapisują w wybranym przez siebie języku programowania.

Wskazówki metodyczne:

- Treści w sekcji „Przeczytaj” można wykorzystać jako podsumowanie i utrwalenie wiedzy uczniów.