



Pętle iteracyjne – zadania maturalne

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



Pętle iteracyjne – zadania maturalne

Źródło: Tine Ivanič, domena publiczna.

W tym e-materiale powtarzamy wiadomości ze szkoły podstawowej.

Poznaliśmy już pojęcie iteracji i [algorytmy iteracyjne](#). Omówiliśmy także [pętlę for](#) – wiemy, że wykorzystuje się ją do operowania na kolejnych elementach zbioru lub do wykonania fragmentu kodu określoną liczbę razy.

W tym e-materiale zapoznamy się z przykładowymi zadaniami maturalnymi dotyczącymi tego zagadnienia.

Implementacje pętli for w poszczególnych językach programowania przedstawiamy w e-materiałach:

- [Pętle iteracyjne w języku C++](#),
- [Pętle iteracyjne w języku Java](#),
- [Pętle iteracyjne w języku Python](#).

Twoje cele

- Zaimplementujesz program z wykorzystaniem pętli for w wybranym języku programowania.
- Rozwiążesz samodzielnie kilka zadań maturalnych.

- Sprawdzisz, w jaki sposób oceniane są zadania maturalne.

Przeczytaj

Powtórzenie informacji na temat pętli for

Pętle pozwalają wielokrotnie wykonywać określone polecenia zapisane w kodzie programu. Pętli for używamy w sytuacjach, w których wiemy, ile razy należy wykonać pewną czynność (czyli wtedy, gdy znaczenie ma liczba powtórzeń) oraz w przypadku przechodzenia po zbiorze elementów (liście, tablicy lub kolekcji), bez potrzeby znajomości liczby jego elementów (jest to możliwe np. w języku Python).

Oto przykładowe zastosowania pętli for:

1. wypisanie pewnej wartości dokładnie pięć razy:

```
1 napis ← "pies"
2
3 dla i = 0, ..., 4 wykonuj:
4     wypisz napis
```

2. wypisanie dokładnie dwóch pierwszych liter pewnego napisu:

```
1 napis ← "pies"
2
3 dla i = 0, 1 wykonuj:
4     wypisz napis[i]
```

3. przypisanie wartości każdemu elementowi tablicy:

```
1 tablica[0..4]
2
3 dla i = 0, ..., 4 wykonuj:
4     tablica[i] ← i
```

Przykładowe zadania maturalne

Zadanie 1. Bitwa podwórkowa

Bartek i Tomek bawią się w bitwę podwórkową z dziećmi z pobliskiego osiedla. Ich kolega Maciek pilnuje trzepaka kilka bloków dalej. Chłopcy chcą przekazać mu wiadomość o nadchodzącym ataku na jego posterunek. Sami nie mogą jednak opuścić swojego posterunku.

Bartkowi udało się zwerbować do drużyny młodszego brata, Kacpra, w roli posłańca. Chłopcy boją się jednak, że Kacper zostanie złapany wraz z wiadomością. Żeby temu zaradzić, Tomek wymyślił algorytm, dzięki któremu przeciwnik nie będzie umiał odczytać wymienianych wiadomości.

Przedstawiony algorytm przekształca wiadomość s o długości n za pomocą odpowiednich przestawień. Po wykonaniu algorytmu przekształcona wiadomość znajdzie się w zmiennej s :

Algorytm:

```
1 s[0..n-1] #wiadomość do przekształcenia
2 dla i = n - 1, n - 2, ..., 0 wykonuj:
3     jeżeli i mod 2 = 0:
4         a ← s[i]
5         s[i] ← s[i div 2]
6         s[i div 2] ← a
7     w przeciwnym wypadku:
8         a ← s[i]
9         s[i] ← s[i mod 2]
10        s[i mod 2] ← a
```

Operator `mod` oznacza resztę z dzielenia, natomiast `div` – dzielenie całkowitoliczbowe.

Podaj wynik działania przedstawionego algorytmu Tomka dla słów **ATAK** oraz **TRZEPAK**.

Specyfikacja problemu:

Dane:

- n – długość łańcucha s ; liczba naturalna
- $s[0..n-1]$ – łańcuch znaków o długości n zawierający wiadomość do przekształcenia

Wynik:

- $s[0..n-1]$ – łańcuch znaków o długości n zawierający przekształconą wiadomość

Zapisz algorytm w postaci programu w języku C++, Java lub Python.

Rozwiązanie

Przeanalizujmy działanie algorytmu.

Zacznijmy od słowa ATAK. Wypiszmy etapy przekształceń z zastosowaniem przedstawionego algorytmu:

```
1 n ← 4
2 s[0..n-1] ← 'ATAK'
3
4 i ← 3
5     a ← s[3] = 'K'
6     s[3] ← s[1] = 'T'
7     s[1] ← a
8     # s = 'AKAT'
9
10 i ← 2
11     a ← s[2] = 'A'
12     s[2] ← s[1] = 'K'
13     s[1] ← a
14     # s = 'AAKT'
15
16 i ← 1
17     a ← s[1] = 'A'
18     s[1] ← s[1] = 'A'
19     s[1] ← a
20     # s = 'AAKT'
21
22 i ← 0
23     a ← s[0] = 'A'
24     s[0] ← s[0] = 'A'
25     s[0] ← a
26     # s = 'AAKT'
```

Analogicznie analizujemy kolejne [iteracje](#) pętli dla słowa TRZEPAK:

```
1 n ← 7
2 s[0..n-1] ← 'TRZEPAK'
```

```
3
4 i ← 6
5     a ← s[6] = 'K'
6     s[6] ← s[3] = 'E'
7     s[3] ← a
8     # s = 'TRZKPAE'
9
10 i ← 5
11     a ← s[5] = 'A'
12     s[5] ← s[1] = 'R'
13     s[1] ← a
14     # s = 'TAZKPRE'
15
16 i ← 4
17     a ← s[4] = 'P'
18     s[4] ← s[2] = '1'
19     s[2] ← a
20     # s = 'TAPKZRE'
21
22 i ← 3
23     a ← s[3] = 'K'
24     s[3] ← s[1] = 'A'
25     s[1] ← a
26     # s = 'TKPAZRE'
27
28 i ← 2
29     a ← s[2] = 'P'
30     s[2] ← s[1] = 'P'
31     s[1] ← a
32     # s = 'TPKAZRE'
33
34 i ← 1
35     a ← s[1] = 'P'
36     s[1] ← s[1] = 'P'
37     s[1] ← a
38     # s = 'TPKAZRE'
39
40 i ← 0
41     a ← s[0] = 'T'
42     s[0] ← s[0] = 'T'
43     s[0] ← a
44     # s = 'TPKAZRE'
```

Wynik: AAKT i TPKAZRE

Schemat oceniania

2 pkt – za obie poprawne odpowiedzi

1 pkt – za jedną poprawną odpowiedź

0 pkt – za błędną odpowiedź lub brak odpowiedzi

Słownik

dekrementacja

zmniejszenie wartości zmiennej o jeden

inkrementacja

zwiększenie wartości zmiennej o jeden

iteracja

technika programowania polegająca na powtarzaniu tej samej operacji w pętli określoną liczbę razy lub aż do spełnienia określonego warunku; termin ten używany jest także w węższym znaczeniu – nazywa się nim operacje wykonywane wewnątrz takiej pętli

nieskończona pętla

pętla, która nigdy nie zostanie zakończona, ponieważ warunek wyjścia z pętli nigdy nie zostaje spełniony

zagnieżdżone pętle

wywołanie pętli wewnątrz innej pętli

Prezentacja multimedialna

Zadanie 2. Poczta bajtcka

Tomek pracuje jako listonosz w Bajtocji. Codziennie otrzymuje z urzędu informację, ile paczek musi dostarczyć danego dnia.

W przyszły wtorek Tomek ma zaplanowaną wizytę u lekarza, natomiast w sobotę miał zamiar pojechać nad jezioro. W związku z tym musi wiedzieć, ile paczek będzie miał do dostarczenia w te dni. Dzięki znajomości z urzędnikami udało mu się zdobyć informację na temat sposobu przydzielania liczby paczek w zależności od dnia tygodnia. Algorytm wygląda następująco:

```
1 funkcja paczki(dzien_tygodnia):
2     a ← dzien_tygodnia mod 3
3     b ← dzien_tygodnia div 2
4     liczba_paczek ← 0
5     dla i = 1, 2, ..., dzien_tygodnia wykonuj:
6         jeżeli a > b wykonaj:
7             liczba_paczek ← liczba_paczek + (a * i + b)
8         w przeciwnym wypadku wykonaj:
9             liczba_paczek ← liczba_paczek + (b * i + a)
10    a ← paczki div 3
11    b ← paczki mod 2
12    zwróć paczki
```

Na podstawie algorytmu podaj, ile paczek Tomek będzie musiał dostarczyć we wtorek, a ile w sobotę. Przyjmij numerację dni tygodnia, rozpoczynającą się od poniedziałku, a zatem: poniedziałek to 1, wtorek to 2 – itd.

Rozwiązanie

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P10YCctr6>

Musimy przeanalizować działanie algorytmu krok po kroku. Na początku obliczamy wartości zmiennych a i b przed wejściem do pętli. Zakładamy numerowanie dni tygodnia od 1, zatem wtorek jest drugim dniem tygodnia:

i	a	b	liczba_paczek
1	$2 \bmod 3 = 2$	$2 \operatorname{div} 2 = 1$...

2

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P10YCctr6>

Zmienna a jest większa od b , zatem dodajemy do zmiennej `liczba_paczek` wyrażenie $a * i + b$:

i	a	b	liczba_paczek
1	$2 \bmod 3 = 2$	$2 \operatorname{div} 2 = 1$	$0 + 2 * 1 + 1 = 3$

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P10YCctr6>

3

Przechodzimy do kolejnej iteracji i aktualizujemy zmienne a oraz b :

i	a	b	liczba_paczek
1	$2 \bmod 3 = 2$	$2 \operatorname{div} 2 = 1$	$0 + 2 * 1 + 1 = 3$
2	$3 \operatorname{div} 3 = 1$	$3 \bmod 2 = 1$...

4

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P10YCctr6>

Tym razem zmienne a oraz b są równe, zatem do zmiennej `liczba_paczek` dodajemy wyrażenie $b * i + a$

i	a	b	liczba_paczek
1	$2 \bmod 3 = 2$	$2 \text{ div } 2 = 1$	$0 + 2 * 1 + 1 = 3$
2	$3 \text{ div } 3 = 1$	$3 \bmod 2 = 1$	$3 + 1 * 2 + 1 = 6$

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P10YCctr6>

5

Działanie pętli dobiegło końca. We wtorek Tomek dostarczy zatem tylko 6 przesyłek.

Analogicznie analizujemy wykonanie algorytmu dla soboty (szóstego dnia tygodnia):

i	a	b	liczba_paczek
1	$6 \bmod 3 = 0$	$6 \text{ div } 2 = 3$	$0 + 3 * 1 + 0 = 3$
2	$3 \text{ div } 3 = 1$	$3 \bmod 2 = 1$	$3 + 1 * 2 + 1 = 6$
3	$6 \text{ div } 3 = 2$	$6 \bmod 2 = 0$	$6 + 2 * 3 + 0 = 12$
4	$12 \text{ div } 3 = 4$	$12 \bmod 2 = 0$	$12 + 4 * 4 + 0 = 28$
5	$28 \text{ div } 3 = 9$	$28 \bmod 2 = 0$	$28 + 9 * 5 + 0 = 73$

i	a	b	liczba_paczek
6	$73 \text{ div } 3 = 24$	$73 \text{ mod } 2 = 1$	$73 + 24 \cdot 6 + 1 = 218$

6

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P10YCctr6>

Liczba paczek, którą Tomek będzie musiał dostarczyć w przyszłą sobotę, wynosi 218.

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

Schemat oceniania

- **2 pkt** – za dwie poprawne odpowiedzi
- **1 pkt** – za jedną poprawną odpowiedź
- **0 pkt** – za błędną odpowiedź lub jej brak

Sprawdź się

Zadanie 3. Ciągi geometryczne

W Uniksie Północnej przeprowadzono badanie wieku mieszkańców. Ze względów praktycznych obywatele nie przekazywali swoich danych pojedynczo, lecz grupowo, w ramach gospodarstw domowych, w których zamieszkiwali. Władze dodatkowo poprosiły, by wiek domowników posortować niemalejąco.

Dany jest plik `wiek.txt` zawierający 100 ciągów liczbowych. Każdy ciąg zapisany jest w dwóch wierszach: w pierwszym znajduje się liczba mieszkańców gospodarstwa domowego, natomiast w drugim wiek domowników uszeregowany niemalejąco.

`wiek.txt`

Plik o rozmiarze 1.50 KB w języku polskim

Napisz program wyznaczający, ile w Uniksie Północnej jest gospodarstw domowych, w przypadku których wiek ich mieszkańców posortowany niemalejąco tworzy ciąg geometryczny. Wynik zapisz do pliku `domostwa.txt`.

Uwaga:

Ciąg jest geometryczny, jeśli każdy jego kolejny wyraz, od drugiego począwszy, jest iloczynem wyrazu poprzedniego i pewnej stałej nazywanej ilorazem ciągu.

Do oceny oddajesz:

- plik `domostwa.txt` zawierający odpowiedź (liczbę naturalną)
- plik(i) z komputerową realizacją zadania (kodem programu)

Praca domowa

Przedstaw rozwiązanie zadania, pisząc program w języku C++, Java lub Python. Odpowiedź znajdziesz w pliku pod sekcją ćwiczeń.

Uwaga!

Z uwagi na format modułów sprawdzających rozwiązania, w ćwiczeniach przyjmujemy liczebność każdego domostwa równą 5.

Pokaż ćwiczenia:   

Ćwiczenie 1



JĘZYK JAVA

Ćwiczenie 2



JĘZYK PYTHON

Ćwiczenie 3



Schemat oceniania:

- **2 pkt** – za poprawną odpowiedź
- **0 pkt** – za błędną odpowiedź lub jej brak

Odpowiedź do zadania

domostwa.txt

Plik o rozmiarze 1.00 B w języku polskim

Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Pętle iteracyjne – zadania maturalne

Grupa docelowa:

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

Podstawa programowa:

Zakres podstawowy i rozszerzony

Cele kształcenia – wymagania ogólne

- 1) Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.
- 2) Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.
- 3) Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi, w tym: znajomość zasad działania urządzeń cyfrowych i sieci komputerowych oraz wykonywania obliczeń i programów.

Treści nauczania – wymagania szczegółowe

Zakres podstawowy

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

4. porównuje działanie różnych algorytmów dla wybranego problemu, analizuje algorytmy na podstawie ich gotowych implementacji;
5. sprawdza poprawność działania algorytmów dla przykładowych danych.

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

Zakres rozszerzony

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

2. stosuje zasady programowania strukturalnego i obiektowego w rozwiązywaniu problemów;
3. sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

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):

- Zaimplementujesz program z wykorzystaniem pętli `for` w wybranym języku programowania.
- Rozwiążesz samodzielnie kilka zadań maturalnych.
- Sprawdzisz, w jaki sposób oceniane są zadania maturalne.

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. Uczniowie przypominają sobie najważniejsze informacje dotyczące pętli iteracyjnych. Nauczyciel sprawdza przygotowanie uczniów do lekcji.
2. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Pętle iteracyjne – zadania maturalne”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj” dotyczącymi programowania.

Faza wstępna:

1. Nauczyciel sprawdza przygotowanie uczniów do zajęć.
2. Nauczyciel wyświetla i odczytuje temat lekcji oraz cele zajęć. Prosi uczniów o sformułowanie kryteriów sukcesu.
3. **Rozpoznanie wiedzy uczniów.** Nauczyciel zadaje uczniom pytania dotyczące ich aktualnego stanu wiedzy w obszarze poruszanego tematu i programowania, np.
 - podaj kilka przykładowych zastosowań pętli for,
 - na czym polega szyfrowanie?Chętni uczniowie udzielają na nie odpowiedzi.

Faza realizacyjna:

1. **Praca z tekstem.** Nauczyciel ocenia, na podstawie informacji na platformie, stan przygotowania uczniów do zajęć. Jeżeli jest ono niewystarczające prosi o ciche zapoznanie się z treścią w sekcji „Przeczytaj”.
2. **Praca z multimediami.** Uczniowie zapoznają się z treścią prezentacji multimedialnej w sekcji „Prezentacja multimedialna”. Następnie w parach przygotowują implementację rozwiązania, wykorzystując wybrany język programowania. Na forum klasy chętne osoby dzielą się swoimi spostrzeżeniami.
3. **Ćwiczenie umiejętności.** Uczniowie indywidualnie wykonują ćwiczenie z sekcji „Sprawdź się”. Następnie wymieniają się z kolegą lub koleżanką rozwiązaniami i je analizują. Dyskutują w parach o swoich rozwiązaniach.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat lekcji zawarty w sekcji „Wprowadzenie” i inicjuje krótką rozmowę na temat zrealizowanych celów (czego uczniowie się nauczyli).

Praca domowa:

1. Uczniowie wykonują ćwiczenie z sekcji „Sprawdź się”, implementując program w lokalnym środowisku. Wczytują do swojego programu dane z pliku.

Wskazówki metodyczne:

- Nauczyciel może wykorzystać multimedium w sekcji „Prezentacja multimedialna” do pracy przed lekcją. Uczniowie zapoznają się z jego treścią i przygotowują do pracy na zajęciach w ten sposób, żeby móc samodzielnie rozwiązać zadania dołączone do e-materiału „Pętle iteracyjne – zadania maturalne”.