



## Tablice jednowymiarowe w języku Java

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Film samouczek](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



## Tablice jednowymiarowe w języku Java

Źródło: Roman Synkevych, domena publiczna.

W tym e-materiale powtarzamy wiadomości ze szkoły podstawowej.

W e-materiale [Tablice jednowymiarowe](#) zapoznaliśmy się ze złożoną strukturą danych, jaką jest tablica. We współczesnych językach programowania tablice znalazły szerokie zastosowanie do przechowywania danych tego samego typu. Za pomocą tych struktur danych można przetwarzać różnego rodzaju wartości, informacje, wyniki pomiarów, a także tworzyć złożone bazy danych. Teraz zajmiemy się implementacją tablic jednowymiarowych w języku Java.

Ciekawi cię, jak wyglądają implementacje w innych językach programowania? Możesz się z nimi zapoznać w dwóch pozostałych e-materiałach z tej serii:

- [Tablice jednowymiarowe w języku C++](#),
- [Tablice jednowymiarowe w języku Python](#).

Więcej zadań? Sięgnij do [Tablice jednowymiarowe – zadania maturalne](#).

### Twoje cele

- Scharakteryzujesz tablice jednowymiarowe.
- Wykorzystasz tablice do rozwiązywania przykładowych problemów.
- Napiszysz program wykorzystujący tablice jednowymiarowe.



# Przeczytaj

---

## Co to jest tablica jednowymiarowa?

Przypomnijmy: **tablica jednowymiarowa** jest „pojemnikiem”, który przechowuje dane tego samego typu. Każda przechowywana tam wartość ma indeks odpowiadający jej położeniu w tablicy. Dobrym porównaniem może tu być choćby szatnia na kurtki. Gdy oddajemy kurtkę do szatni, otrzymujemy numerkę oznaczający miejsce, na którym została powieszona nasza kurtka. Zwracając numerkę, dostajemy kurtkę z powrotem. Kurtki w tym przykładzie to wartości przechowywane w tablicy, a numerki to ich indeksy – dzięki nim nawet w ogromnej szatni z łatwością odnajdziemy swoją kurtkę.

## Implementacja w języku Java

Deklaracja tablicy jednowymiarowej w języku Java wygląda następująco:

```
1 int tablica[] = new int[3];
```

lub ewentualnie:

```
1 int[] tablica = new int[3];
```

Obie te formy są poprawne.

Rozbijmy tę deklarację na części:

1. Pierwszą czynnością jest określenie typu wartości, jakie będą przechowywane w naszej tablicy. W tym przypadku jest to `int`.
2. Następnie nadajemy tablicy nazwę. Będzie ona potrzebna, aby w przyszłości móc się do niej odwołać.
3. Znaki `[ ]` nie stanowią części nazwy, ale są wymagane przy deklaracji tablicy. Nie ma jednak znaczenia, czy znaki te pojawią się przy typie danych, czy też przy nazwie.
4. Kolejnym elementem jest słowo kluczowe `new`, po którym po raz kolejny pojawia się typ danych do przechowania.
5. Na koniec określamy rozmiar tablicy. Rozmiar jest liczbą elementów, które będą w tablicy przechowywane. Tablice w języku Java są **statyczne**.

### Ciekawostka

Istnieje jeszcze jeden sposób **inicjalizacji** tablicy w języku Java:

```
1 int tablica[] = {2, 6, 7};
```

Tego sposobu używamy, gdy chcemy stworzyć tablicę od razu wypełnioną danymi. Rozmiar tablicy będzie równy liczbie elementów zamkniętych w klamrach.

### Ciekawostka

Jeżeli chcemy się dowiedzieć, jaki rozmiar ma tablica, możemy użyć:

```
1 tablica.length
```

### Ważne!

Indeksy w tablicach zaczynają się od 0. Przykładowo, gdy deklarujemy tablicę o rozmiarze 5, możliwe indeksy to: {0, 1, 2, 3, 4}.

Tablica została stworzona. Teraz zapiszemy w niej dane. Powiedzmy, że do tablicy `tablica` chcemy zapisać liczbę `33`. W naszej tablicy nie ma jeszcze żadnych danych, co oznacza, że pierwszym wolnym indeksem jest indeks 0 – i tam właśnie zapiszemy liczbę:

```
1 tablica[0] = 33;
```

Aby zapisać dane w tablicy, musimy użyć nazwy, którą wcześniej jej nadaliśmy. Bezpośrednio po nazwie podajemy w nawiasie kwadratowym indeks miejsca w tablicy, gdzie dana liczba ma być zapisana. Na koniec przypisujemy żądaną wartość.

Odczytywanie danych z tablicy jest bardzo podobne do zapisywania. Oto przykład:

```
1 System.out.println(tablica[0]);
```

Wynikiem tej liniiki będzie wypisanie wartości `33`, którą wcześniej zapisaliśmy w miejscu o indeksie 0.

## Do czego wykorzystujemy tablice?

Tablice w języku Java są zazwyczaj wykorzystywane, gdy mamy wiele powiązanych ze sobą danych tego samego typu. Ponieważ tablice są statyczne, zazwyczaj liczba zawartych w nich elementów jest znana lub też istnieje wartość maksymalna, która nie powinna zostać przekroczona przez program. Przykładem takiej implementacji może być tablica z zapisanymi imionami wszystkich pasażerów samochodu. Jeżeli samochód jest

pięcioosobowy, tablica też będzie miała pięć elementów. Taka implementacja mogłaby wyglądać następująco:

```
1 String imionaPasazerow[] = new String[5];
```

## Słownik

**inicjalizacja zmiennej**

stworzenie zmiennej i nadanie jej wartości początkowej

**tablica statyczna**

tablica, która wymaga podania rozmiaru przy inicjalizacji i nie jest możliwa jego zmiana

# Film samouczek

---

## Problem 1

### Specyfikacja problemu:

Napisz program znajdujący maksymalny, niemalejący, spójny podciąg w ciągu liczb.

Ciąg liczb do przetestowania 6, 7, 1, 3, 4.

### Dane:

Algorytm opiera swoje działanie na trzech zmiennych:

- długości aktualnie analizowanego podciągu
- długości aktualnie najdłuższego podciągu
- indeksie, na którym rozpoczyna się najdłuższy podciąg

### Wynik:

Na standardowym wyjściu program prezentuje maksymalny niemalejący, spójny podciąg.

## Polecenie 1

Porównaj swoje rozwiązanie z filmem poniżej.

# Wystąpił błąd



# Wprowadzenie do tablic – tablice jednowymiarowe

Realizacja tablic w języku Java



Film dostępny pod adresem </preview/resource/R10a713ZTA25N>

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

Film dotyczy wprowadzenia do tablic – tablic jednowymiarowych.

---

## Polecenie 2

Wyobraź sobie, że jako analityk dostałeś zestaw liczb i musisz znaleźć pośród nich 3 wartości: maksimum, minimum oraz średnią. Do rozwiązania tego problemu przyda się pętla for, dlatego poniżej krótkie przypomnienie.

Pętla for jest pętlą iteracyjną, w której następuje określona liczba iteracji. Zmienna sterująca (zazwyczaj nazywana „i”) pozwala na określenie, w której iteracji program się znajduje.

Przeanalizuj prezentację, a następnie zaimplementuj omówione rozwiązania w swoim programie.

## Wystąpił błąd

Nagranie

---



# Sprawdź się

---

Pokaż ćwiczenia:   

## Ćwiczenie 1



Napisz program, który sprawdzi, ile jest liczb parzystych w tablicy `tablica`.

### Specyfikacja problemu:

*Dane:*

- `n` – liczba naturalna dodatnia, liczba elementów w tablicy
- `tablica` – `n`-elementowa tablica liczb całkowitych

*Wynik:*

Na standardowym wyjściu program wypisuje liczbę liczb parzystych w zadanej tablicy.

## Ćwiczenie 2



Napisz program, który wypisze wszystkie elementy n-elementowej tablicy `tablica` podzielne przez `m`. Przetestuj działanie programu dla tablicy `tablica[] = {999, 3542, 654, 254326, 859, 432, 678, 765, 435, 645}` oraz `n = 3`.

### Specyfikacja problemu:

#### *Dane:*

- `m` – liczba naturalna
- `n` – liczba naturalna dodatnia, liczba elementów w tablicy
- `tablica` – m-elementowa tablica liczb całkowitych

#### *Wynik:*

Na standardowym wyjściu program wypisuje liczby podzielne przez `n` w zadanej tablicy. Każdą liczbę zapisuje w nowej linii.

# Dla nauczyciela

---

**Autor:** Maurycy Gast

**Przedmiot:** Informatyka

**Temat:** Tablice jednowymiarowe w języku Java

**Grupa docelowa:**

Liceum ogólnokształcące i technikum, 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 + II. Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

2) wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów:

c) znajdowania w ciągu podciągów o różnorodnych własnościach, np. najdłuższego spójnego podciągu niemalejącego, spójnego podciągu o największej sumie,

**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 tablice jednowymiarowe.

- Wykorzystasz tablice do rozwiązywania przykładowych problemów.
- Napiszesz program wykorzystujący tablice jednowymiarowe.

### **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-materiałach;
- tablica interaktywna/tablica, pisak/kreda;
- oprogramowanie dla języka Java SE 8 (lub nowszej wersji), w tym Eclipse 4.4 (lub nowszej wersji).

### **Przebieg lekcji**

#### **Przed lekcją:**

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

#### **Faza wstępna:**

1. Nauczyciel wyświetla uczniom temat, wskazuje cele zajęć oraz ustala z uczestnikami zajęć kryteria sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel prosi wybranego ucznia lub uczniów o przedstawienie sytuacji problemowej związanej z tematem lekcji.

#### **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”. Uczniowie analizują przedstawione tam przykłady i powtarzają zaprezentowane rozwiązania na swoim komputerze.
2. **Praca z multimediami.** Nauczyciel wyświetla zawartość sekcji „Film samouczek”. Uczniowie w parach wykonują polecenie 1, a następnie porównują swoje rozwiązanie z przedstawionym w filmie.
3. **Ćwiczenie umiejętności.** Poszukiwanie najefektywniejszego rozwiązania problemu. Uczniowie wykonują w parach ćwiczenie nr 1 z sekcji „Sprawdź się”, a następnie porównują swój kod omawiając go wspólnie na forum. Nauczyciel ocenia efektywność zastosowanego rozwiązania.
4. Uczniowie w parach wykonują ćwiczenie nr 2. Nauczyciel sprawdza poprawność zadania, omawiając je wraz z uczniami.

### **Faza podsumowująca:**

1. Na koniec zajęć z programowania w Javie nauczyciel prosi uczniów o rozwinięcie zdania: „Na dzisiejszych zajęciach nauczyłam/łem się jak...”.

### **Praca domowa:**

1. Uczniowie dodają do swoich kodów rozbudowany komentarz.
2. Uczniowie wykonują polecenie 3 i 4 z sekcji „Film samouczek”.

### **Materiały pomocnicze:**

- Oficjalna dokumentacja techniczna dla języka Java SE 8 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Eclipse 4.4 (lub nowszej wersji).

### **Wskazówki metodyczne:**

- Uczniowie mogą wykorzystać treści w sekcjach: „Przeczytaj”, „Film samouczek”, „Sprawdź się” jako materiał do lekcji powtórkowej.