

Instrukcja warunkowa w języku Java

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

An aerial night photograph of a roundabout with several lanes and streetlights. A dark semi-transparent rectangle is centered over the roundabout, containing the title text in white. In the bottom right corner of the image, there is a small white box with a grey border containing the source information.

Instrukcja warunkowa w języku Java

Źródło: Osman Rana, domena publiczna.

W tym e-materiale powtarzamy wiadomości ze szkoły podstawowej.

Wiesz już, czym charakteryzują się instrukcje warunkowe. Informacje na ten temat znajdziesz w e-materiale [Instrukcja warunkowa](#).

W tym e-materiale zapoznasz się z ich implementacją w języku Java.

Implementacje w innych językach programowania:

- [Instrukcja warunkowa w języku C++](#),
- [Instrukcja warunkowa w języku Python](#).

Więcej zadań? Sięgnij do: [Instrukcja warunkowa – ćwiczenia](#).

Twoje cele

- Scharakteryzujesz istotę instrukcji warunkowych w języku Java.
- Przeanalizujesz budowę wyrażeń logicznych oraz wymienisz operatory relacji i logiczne.
- Wykorzystasz instrukcję warunkową, by rozwiązać zadania praktyczne.

Przeczytaj

Podsumowanie

Oto zapis przykładowej instrukcji warunkowej w języku Java:

```
1 int x = 5;
2
3 if (x < 10) {
4     System.out.println("Liczba jest mniejsza od 10");
5 }
```

Z przykładu wynika, że zmienna `x` **inicjowana** jest wartością 5. Zapisane w linii 3. wyrażenie warunkowe będzie prawdziwe, gdy liczba będzie mniejsza od 10. W naszym przykładzie po sprawdzeniu warunku zostanie zwrócona wartość `true`.

Definiowanie warunków

```
1 int x = 5;
2
3 if (x > 5) {
4     // warunek niespełniony
5 }
6
7 if (x < 5) {
8     // warunek niespełniony
9 }
10
11 if (x >= 5) {
12     // warunek spełniony
13 }
14
15 if (x <= 5) {
16     // warunek spełniony
17 }
18
19 if (x == 5) {
20     // warunek spełniony
```

```
21 }
22
23 if (x != 5) {
24     // warunek niespełniony
25 }
```

Ważne!

Zwróć uwagę na dwie ostatnie instrukcje warunkowe.

Operator `!=` odczytujemy jako „różne od”. Oznacza to, że sprawdzane wyrażenie rozumiemy w następujący sposób: „jeżeli jest różne od 5”.

Operator `==` używany jest do porównywania wartości w celu sprawdzenia, czy są one takie same. Nie wolno mylić go z operatorem przypisania wartości (`=`).

Złożone wyrażenia logiczne

Instrukcje warunkowe zagnieżdżone

Przykład użycia zagnieżdżonych instrukcji warunkowych:

```
1 public class Start {
2     public static void main(String[] args) {
3         int x = 5;
4         if (x < 10) {
5             if (x > 3) {
6                 System.out.println("Liczba jest większa niż 3 i
7             }
8         }
9     }
10 }
```

Łączenie warunków za pomocą operatorów logicznych

Istnieje możliwość stosowania operatorów logicznych w celu sprawdzenia większej liczby warunków. Przykładem operatora logicznego jest `&&` (oznaczający koniunkcję, odczytywaną jako „i”).

Przykład wyrażenia logicznego, w którym wykorzystano operator „i”:

```

1 public class Start {
2     public static void main(String[] args) {
3         int x = 5;
4         if (x < 10 && x > 3) {
5             System.out.println("Liczba jest większa niż 3 i mniej
6         }
7     }
8 }

```

Zapisana w przykładzie instrukcja warunkowa sprawdza, czy zmienna `x` jednocześnie jest większa od 3 i jest mniejsza od 10, czyli należy do przedziału (3, 10).

Kolejnym operatorem logicznym jest `||` (oznaczający alternatywę, odczytywaną jako „lub”).

Oto przykład wyrażenia logicznego, w którym wykorzystano operator „lub”:

```

1 int x = 0;
2
3 if (x > 0 || x < 0) {
4     System.out.println("Liczba nie jest zerem");
5 } else {
6     System.out.println("Liczba jest zerem");
7 }

```

Zapisana w tym przykładzie instrukcja warunkowa sprawdza, czy zmienna `x` ma wartość mniejszą od 0 lub większą od 0. Jeżeli którykolwiek z tych warunków jest spełniony, alternatywa uznawana jest za prawdziwą (przypisywana jej zostaje wartość logiczna `true`). Ponieważ wartość zmiennej `x` to 0, warunek nie zostanie spełniony i wykonana będzie instrukcja następująca po słowie kluczowym `else`.

Instrukcja `if-else`

Jeżeli wyrażenie logiczne zapisane w instrukcji warunkowej nie zostanie spełnione, można zastosować instrukcję `if-else`, którą odczytujemy jako „w przeciwnym razie”.

Oto przykład użycia instrukcji `if-else` w języku Java:

```

1 public class Main {
2     public static void main(String[] args) {
3         int x = 5;

```

```

4     if (x < 10 && x > 3) {
5         System.out.println("Liczba jest większa niż 3 i mniej
6     } else {
7         System.out.println("Liczba jest mniejsza lub równa 3
8     }
9 }
10 }

```

Przykładowy problem

Przyjrzyjmy się bardziej rozbudowanemu przykładowi. Wyobraźmy sobie taką sytuację: bilet lotniczy kosztuje 210 zł – osoby, które chcą przewieźć bagaż cięższy niż 5 kg, muszą do ceny podstawowej dopłacić 80 zł. Kryterium to jest warunkiem i można zapisać go takim zdaniem: jeśli osoba podróżuje z bagażem cięższym niż 5 kg, to cena biletu wzrasta o 80 zł.

Pasażerowie mogą dodatkowo ubezpieczyć swój bagaż za 50 zł, ale jeśli dopłacają już za nadbagaż, to ubezpieczenie kosztuje tylko 35 zł. Osoby podróżujące klasą biznesową w cenie biletu mają już ubezpieczenie i opłatę za nadbagaż.

Korzystając z samej instrukcji warunkowej, musielibyśmy stworzyć szereg warunków:

```

1  public class Start {
2      public static void main(String[] args) {
3          doZaplaty = cena;
4          if (klasaBiznes == false) {
5              if (wagaBagazu > 5) {
6                  doZaplaty = cena + 80;
7              }
8
9              if (ubezpieczenie == true) {
10                 doZaplaty += 50;
11                 if (wagaBagazu > 5) {
12                     doZaplaty -= 15;
13                 }
14             }
15         }
16     }
17 }

```

Wykorzystując operatory logiczne, możemy znacznie uprościć kod obliczający cenę biletu:

```
1 public class Start {
2     public static void main(String[] args) {
3         int doZaplaty = cena;
4         if (klasaBiznes == false && wagaBagazu > 5) {
5             doZaplaty = cena + 80;
6         }
7
8         if (klasaBiznes == false && ubezpieczenie == true) {
9             doZaplaty = doZaplaty + 50;
10            if (wagaBagazu > 5) {
11                doZaplaty = doZaplaty - 15;
12            }
13        }
14    }
15 }
```

Słownik

inicjacja zmiennej

nadanie zmiennej wartości początkowej

Schemat interaktywny

Polecenie 1

Napisz program, w którym na podstawie podanej temperatury ocenisz stan zdrowia użytkownika.

Działanie programu przetestuj dla różnych temperatur.

Specyfikacja:

Dane:

temperatura – liczba rzeczywista dodatnia

Wynik:

- Dla temperatury równej $36,6^{\circ}\text{C}$ program na standardowym wyjściu wyświetla komunikat „Jesteś zdrowy”.
- Dla temperatury mniejszej niż $36,6^{\circ}\text{C}$ program na standardowym wyjściu wyświetla komunikat „Jesteś osłabiony”.
- Dla temperatury większej niż $36,6^{\circ}\text{C}$ program na standardowym wyjściu wyświetla komunikat „Masz gorączkę”.

Polecenie 2

Porównaj swoje rozwiązanie z filmem.

Trwa wczytywanie danych ..



Instrukcja warunkowa

Warunek w Javie



Film dostępny pod adresem </preview/resource/Rrc0at5rs3642>

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

Film nawiązujący do treści materiału: Instrukcja warunkowa.

Przycisk do pobrania TXT z kodem źródłowym z filmu.

Plik o rozmiarze 485.00 B w języku polskim

Polecenie 3

Napisz program wyliczający cenę zakupu biletów lotniczych. Weź pod uwagę kilka założeń. Podstawowa cena biletu oferowanego przez linie lotnicze to 250 zł. Dopłata za przewożenie bagażu cięższego niż 6 kg wynosi 70 zł. Ubezpieczenie bagażu to dodatkowa kwota w wysokości 40 zł – jeśli jednak pasażerowie dopłacają już za nadbagaż, to ubezpieczenie kosztuje ich tylko 25 zł. Osoby podróżujące klasą biznesową mają cenę zarówno nadbagażu jak i ubezpieczenia wliczoną w koszt biletu, którego cena jest wyższa o 150 zł od podstawowej.

Przetestuj swój program dla pewnej rodziny. Rodzina (matka i syn) nie leci klasą biznes. Matka bierze ze sobą 9 kg bagażu. Syn bierze ze sobą jedynie 4 kg bagażu. Oboje decydują się na ubezpieczenie.

Ile zapłacą za bilety?

Specyfikacja:

Dane:

- cena – liczba naturalna
- wagaBagazuMatka – liczba naturalna
- wagaBagazuSyn – liczba naturalna
- klasaBiznes – wartość logiczna
- ubezpieczenie – wartość logiczna

Wynik:

- doZapłaty – liczba naturalna; sumaryczny koszt podróży matki i syna

Zmodyfikuj swój program tak, by sprawdzić, ile rodzina zapłaci za bilety w klasie biznes.

Problem 1

Uczniowie przygotowali program, który wyliczał, ile muszą zapłacić za bilety PKS w zależności od tego, jaką trasę pokonują.

Polecenie 4

Przeanalizuj prezentację, w której opisano algorytm wyliczania rabatów dla pewnego sklepu. Na jej podstawie spróbuj wytłumaczyć zasadę działania instrukcji warunkowej dowolnej osobie z twojego otoczenia. Zwróć uwagę na posługiwanie się prawidłowym słownictwem.

Przetestuj działanie programu dla klienta, który kupuje 10 sztuk towaru, którego jednostkowa cena wynosi 5 zł. W przypadku tego klienta rabat naliczany jest od zakupu powyżej 4 sztuk produktów i dla każdej sztuki wynosi 10%.

Specyfikacja:

Dane:

- `liczbaSztuk` – liczba sztuk produktu; liczba naturalna
- `cenaTowaru` – jednostkowa cena towaru; liczba naturalna
- `rabatPowyzejSztuk` – od ilu produktów naliczany jest rabat; liczba naturalna
- `wysokoscRabatu` – rabat liczony od `cenyTowaru` dla każdej sztuki powyżej `rabatPowyzejSztuk`; liczba rzeczywista dodatnia

Wynik:

- `doZaplat` – sumaryczny koszt zakupionego towaru; liczba rzeczywista




Trwa wczytywanie danych ..

Nagranie dostępne pod adresem <https://zpe.gov.pl/a/DpPqQLbc>

Nagranie głosowe

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

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Napisz program, który sprawdzi, czy podana liczba x jest mniejsza od 10. Przetestuj jego działanie dla liczby $x = 7$.

Specyfikacja:

Dane:

x – liczba naturalna

Wynik:

Program wyświetla słowo: tak, gdy liczba jest mniejsza od 10; w przeciwnym przypadku program wypisuje słowo nie.

Ćwiczenie 2



Napisz program, który sprawdzi, czy podana liczba x jest parzysta, większa od 50, ale mniejsza od 100.

Sprawdź jego działanie dla $x = 73$.

Specyfikacja:

Dane:

x – liczba naturalna dodatnia

Wynik:

Gdy liczba x jest parzysta, większa od 50 i mniejsza od 100, program wyświetla słowo: **tak**; w przeciwnym przypadku program wyświetla słowo: **nie**.

Ćwiczenie 3



Uczniowie pisali sprawdzian z informatyki, z którego mogli uzyskać maksymalnie 100 punktów. Punktacja przedstawiała się następująco:

- 0 – 39 pkt – ndst
- 40 – 54 pkt – dop
- 55 – 69 pkt – dst
- 70 – 84 pkt – db
- 85 – 98 pkt – bdb
- 99 – 100 pkt – cel

Napisz program, który przypisze ocenę do podanej liczby punktów.

Przetestuj swoje rozwiązanie dla 22 pkt.

Specyfikacja:

Dane:

punkty – liczba naturalna

Wynik:

Program wyświetla, w zależności od liczby uzyskanych punktów, jedno ze słów: ndst, dop, dst, db, bdb lub cel.

Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Instrukcja warunkowa w języku Java

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres podstawowy

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.

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).
- 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.

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);
- 2) do realizacji rozwiązań problemów prawidłowo dobiera środowiska informatyczne, aplikacje oraz zasoby, wykorzystuje również elementy robotyki;

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 istotę instrukcji warunkowych w języku Java.
- Przeanalizujesz budowę wyrażeń logicznych oraz wymienisz operatory relacji i logiczne.
- Wykorzystasz instrukcję warunkową, by rozwiązać zadania praktyczne.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

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

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 Java SE 8 (lub nowszej wersji), w tym Eclipse 4.4 (lub nowszej wersji).

Przebieg lekcji

Przed lekcją:

1. Uczniowie zapoznają się z filmem umieszczonym w sekcji „Schemat interaktywny”.

Faza wstępna:

1. Wyświetlenie przez nauczyciela tematu i celów lekcji. Określenie wiążących dla uczniów kryteriów sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Uczniowie tworzą pytania dotyczące tematu zajęć, na które odpowiedzą w trakcie lekcji.

Faza realizacyjna:

1. Uczniowie rozwiązują Polecenie 1 z sekcji „Schemat interaktywny”.
2. Uczniowie rozwiązują Polecenie 2 z sekcji „Schemat interaktywny”.
3. Chętna lub wybrana osoba prezentuje swój program na forum klasy. Uczniowie dyskutują na jego temat.
4. Ćwiczenia nr 1–3 z sekcji „Sprawdź się” uczniowie wykonują w grupach czteroosobowych, a następnie porównują swoje odpowiedzi z inną grupą.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat i cele lekcji zawarte w sekcji „Wprowadzenie”. W kontekście ich realizacji następuje omówienie ewentualnych problemów z rozwiązaniem ćwiczeń z sekcji „Sprawdź się”.
2. Na koniec zajęć z programowania w Javie nauczyciel prosi uczniów o rozwinięcie zdania: „Na dzisiejszych zajęciach nauczyłam/łem się jak...”.

Praca domowa:

1. Uczniowie rozwiązują Problem 3 z sekcji „Schemat interaktywny”.

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”, „Schemat interaktywny”, „Sprawdź się” jako materiał do lekcji powtórkowej.