

Materiał pomocniczy do kursu „Podstawy programowania”

Autor: Grzegorz Góralski

ggoralski.com

Operatory logiczne

Podstawowe operatory logiczne, składanie
wyrażeń z użyciem operatorów logicznych

Podstawowe operatory logiczne

operator	znaczenie	opis
&&	AND	Zwraca true jeśli oba wyrażenia są prawdziwe.
	OR	Zwraca true jeśli przynajmniej jedno z wyrażen jest prawdziwe.

✱ Przykłady:

wyrażenie	wynik
<code>int x = 5;</code>	
<code>x == 5 && x > 0</code>	<code>true</code>
<code>x == 5 x > 0</code>	<code>true</code>
<code>x == 5 && x < 0</code>	<code>false</code>
<code>x == 5 x < 0</code>	<code>true</code>

Priorytety operatorów

- * Podobnie jak w matematyce (np. mnożenie przed dodawaniem), także w programowaniu należy brać pod uwagę priorytety operatorów.
- * Poznane dotychczas operatory można uporządkować, w porządku od tych z największym priorytetem:
 1. <, >, <=, >=
 2. ==, !=
 3. &&
 4. ||

Łączenie i porządkowanie

- * Domyślne priorytety operatorów niekoniecznie muszą odpowiadać temu co chcemy, żeby komputer zrobił.
- * Na przykład w matematyce $2+4*2$ to nie to samo co $(2+4)*2$.
- * Ponadto często chcemy połączyć różnego rodzaju porównania w dość skomplikowany układ warunków, który powinien być rozpatrywany w określonej kolejności.
- * Możemy sobie pomóc używając nawiasów.
- * Ma to znaczenie także dla przejrzystości kodu.

Przykład

- * Piszemy aplikację, która określa grupę krwi (A, B, 0) na podstawie dostarczonego genotypu (użytkownik podaje osobno dwa allele).
- * Jak wiadomo grupa krwi w systemie A, B, 0, zależy od genu który w populacji ludzkiej występuje w postaci trzech alleli: I^A , I^B , i^0
- * Tabela pokazuje jakie genotypy odpowiadają poszczególnym grupom krwi:

grupa krwi	genotypy
AB	$I^A i^B$
A	$I^A I^A, I^A i^0$
B	$I^B I^B, I^B i^0$
0	$i^0 i^0$

Realizacja

- * Dla uproszczenia, allele przy wprowadzaniu oznaczymy jako A, B, 0
- * Użytkownik wprowadzi w konsoli najpierw jeden (zmienna `allele1`), a później drugi allele (zmienna `allele2`)
- * Trzeba wziąć pod uwagę, że kolejność wprowadzonych alleli może być dowolna (`allele1="A"`, `allele2="0"` jest równoważne `allele1="0"`, `allele2="A"`), genotypy $I^A i^0$ oraz $i^0 I^A$ są przecież tożsame.

grupa krwi	genotypy	układy zmiennych
AB	$I^A i^B$	A B, B A
A	$I^A I^A, I^A i^0$	A A, A 0, 0 A
B	$I^B I^B, I^B i^0$	B B, B 0, 0 B
0	$i^0 i^0$	0 0

Projektujemy program

- * Zaprojektujemy ogólną konstrukcję programu używając **pseudokodu**:

Jeśli (warunek na grupę krwi AB) to wypisz „Grupa AB”

Jeśli (warunek na grupę krwi A) to wypisz „Grupa A”

Jeśli (warunek na grupę krwi B) to wypisz „Grupa B”

Jeśli (warunek na grupę krwi O) to wypisz „Grupa O”

Jeśli żaden z powyższych to wypisz „Złe dane!”

- * Zastosujemy konstrukcję: **if ... else if ... else ...**
- * Kluczową sprawą jest teraz prawidłowe sformułowanie warunków dla poszczególnych grup krwi

grupa krwi	genotypy	układy zmiennych
AB	$I^A i^B$	A B, B A
A	$I^A I^A, I^A i^0$	A A, A 0, 0 A
B	$I^B I^B, I^B i^0$	B B, B 0, 0 B
0	$i^0 i^0$	0 0

* Zacznijmy od najprostszej grupy 0:

* Pierwszy allel musi być 0 i drugi allel musi być 0:

```
if (allele1.equalsIgnoreCase("0") && allele2.equalsIgnoreCase("0"))
    System.out.println("Grupa krwi: 0");
```

* Teraz AB:

* Pierwszy allel musi być A i drugi B, **lub** pierwszy B i drugi A

```
else if ((allele1.equalsIgnoreCase("A") && allele2.equalsIgnoreCase("B")) ||
         (allele1.equalsIgnoreCase("B") && allele2.equalsIgnoreCase("A")))
    System.out.println("Grupa krwi: AB");
```


grupa krwi	genotypy	układy zmiennych
AB	$I^A i^B$	A B, B A
A	$I^A I^A, I^A i^0$	A A, A 0, 0 A
B	$I^B I^B, I^B i^0$	B B, B 0, 0 B
0	$i^0 i^0$	0 0

* Teraz grupa A:

* Pierwszy allel A i drugi allel A lub pierwszy allel A i drugi allel 0 lub pierwszy allel 0 i drugi allel A.

```
else if ((allel1.equalsIgnoreCase("A") && allel2.equalsIgnoreCase("A")) ||
         (allel1.equalsIgnoreCase("A") && allel2.equalsIgnoreCase("0")) ||
         (allel1.equalsIgnoreCase("0") && allel2.equalsIgnoreCase("A")))
    System.out.println("Grupa krwi: A");
```

grupa krwi	genotypy	układy zmiennych
AB	$I^A i^B$	A B, B A
A	$I^A I^A, I^A i^0$	A A, A 0, 0 A
B	$I^B I^B, I^B i^0$	B B, B 0, 0 B
0	$i^0 i^0$	0 0

* Grupa B analogicznie do A:

* Pierwszy allel B i drugi allel B **lub** pierwszy allel B i drugi allel 0 **lub** pierwszy allel 0 i drugi allel B

```
else if ((allel1.equalsIgnoreCase("B") && allel2.equalsIgnoreCase("B")) ||
         (allel1.equalsIgnoreCase("B") && allel2.equalsIgnoreCase("0")) ||
         (allel1.equalsIgnoreCase("0") && allel2.equalsIgnoreCase("B")))
    System.out.println("Grupa krwi: B");
```

* i w końcu:

```
else System.out.println("Złe dane!");
```

```

String allele1, allele2;
Scanner skaner = new Scanner(System.in);
System.out.print("Podaj pierwszy allel [A/B/0]: ");
allele1 = skaner.nextLine();
System.out.print("Podaj drugi allel [A/B/0]: ");
allele2 = skaner.nextLine();

if (allele1.equalsIgnoreCase("0") && allele2.equalsIgnoreCase("0"))
    System.out.println("Grupa krwi: 0");
else if ((allele1.equalsIgnoreCase("A") && allele2.equalsIgnoreCase("B")) ||
         (allele1.equalsIgnoreCase("B") && allele2.equalsIgnoreCase("A")))
    System.out.println("Grupa krwi: AB");
else if ((allele1.equalsIgnoreCase("A") && allele2.equalsIgnoreCase("A")) ||
         (allele1.equalsIgnoreCase("A") && allele2.equalsIgnoreCase("0")) ||
         (allele1.equalsIgnoreCase("0") && allele2.equalsIgnoreCase("A")))
    System.out.println("Grupa krwi: A");
else if ((allele1.equalsIgnoreCase("B") && allele2.equalsIgnoreCase("B")) ||
         (allele1.equalsIgnoreCase("B") && allele2.equalsIgnoreCase("0")) ||
         (allele1.equalsIgnoreCase("0") && allele2.equalsIgnoreCase("B")))
    System.out.println("Grupa krwi: B");
else
    System.out.println("Złe dane!");

```

Zadanie:

- * Poprawić program rozpoznawania płci po chromosomach (z prezentacji dot. instrukcji wyboru) grupując warunki z użyciem poznanych operatorów logicznych .