

==	Equal to
!=	Not equal
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

%	Modulus	Returns the division remainder
++	Increment	Increases the value of a variable by 1
--	Decrement	Decreases the value of a variable by 1
&&	Logical and	Returns true if both statements are true
	Logical or	Returns true if one of the statements is true

Ukoncenie prebiehajúceho programu – Ctrl + F2

Generovanie nahodneho cisla:

```
int n = 10; // od -10 po 10
Random random = new Random();
int nahodneCislo = random.nextInt(2*n + 1) - n;
```

`Character.isLetter(char ch)`: This method returns true if the specified character is a letter. It includes uppercase and lowercase letters from all languages.

`Character.isDigit(char ch)`: This method returns true if the specified character is a digit (0-9).

ULOHY

1. Generovanie nahodneho cisla

Vytvorte metodu

```
int nahodneCislo(int limit)
```

ktora vrati nahodne vygenerovane cislo s pouzitim triedy `java.util.Random`, kde parameter `limit` urcuje maximalne a minimalne zaporne vygenerovane cislo, vratane cisla `n`. Metodu otestujte.

IF PODMIENKY

1. Kladne/zaporne cislo

Vasou ulohou je naprogramovat metodu

```
boolean jeKladne(int cislo)
```

ktora overi, ci je cislo kladne, alebo zaporne. Cislo 0 berte ako kladne. Do triedy main implementujte vypis z tejto metody "cislo 5 je kladne", kde cislo "5" nahradite vstupnym parametrom.

Vzorovy I/O:

cislo = 5 -> true

cislo = 0 -> true

cislo = -11 -> false

```
Cislo 5 je kladne
Cislo 0 je kladne
Cislo -11 je zaporne
```

2. Delitelnost cisla

Vasou ulohou je naprogramovat metodu

```
boolean jeDelitelne(int cislo, int delitel)
```

ktora overi, ci je cislo delitelne delitelom. V triede main ju vyuzite a vypiste o tom spravu napr.: "cislo 6 nie je delitelne cislom 5".

Vzorovy I/O:

cislo = 5, delitel = -15 -> false

cislo = -15, delitel = 5 -> true

```
Cislo 5 nie je delitelne cislom -15
Cislo -15 je delitelne cislom 5
```

3. Je znak pismeno alebo cislo?

Vasou ulohou je naprogramovat metodu

```
boolean jeCislo(char znak)
```

ktora zisti, ci zadani znak je cislo, alebo pismeno. Pouzite metody triedy Character. V triede main tuto metodu implementujte s vypisom "znak '4' je cislo".

Vzorovy I/O:

znak = 'a' -> false

znak = '1' -> true

```
Znak 'a' je pismeno  
Znak '1' je cislo
```

4. Znamka

Naprogramujte metodu

```
void vypisZnamku(int pocetPercent)
```

Która na základne parametra pocetPercent vypíše, akú známku študent dostal.

Legenda:

A 100-91

B 90-81

...

E 60-51

FX <51.

Ak študent mal inú známku ako FX, vypíšte "študent presiel so známkou A a počtom percent 95". V opačnom prípade vypíšte "študent nepresiel. Pre minimálnu známku chýbalo 19%".

Ošetríte aj prípad, ak pocetPercent je záporný. Tuto metódu použijete v triede main.

Vzorový výstup:

```
Student presiel so známkou A a počtom percent 95  
Student presiel so známkou C a počtom percent 79  
Student nepresiel. Pre minimálnu známku chýbalo 19%
```

5. Najväčšie číslo z 3

Naprogramujte metodu

```
int najvacsieCisloZTroch(int cislo1, int cislo2, int cislo3)
```

ktorá vráti najväčšie číslo z troch čísel uvedených v parametre metódy. Metódu použijete v triede main so správou: "z čísel 1, 2, 3 je najväčšie 3".

Vzorový I/O:

cislo1 = 5, cislo2 = 0, cislo3 = -15 -> 5

```
Z čísel 5, 0, -15 je najväčšie 5
```

FOR CYKLY

1. Nekonečný cyklus

Vytvorte metódu

```
void nekonecnyCyklus()
```

ktora bude obsahovat nekonecny for loop a bude do consoly vypisovat string "toto je nekonecny cyklus".

```
Toto je nekonecny cyklus
Toto je nekonecny cyklus
Toto je nekonecny cyklus
Toto je nekonecny cyklus
Toto je nekonecny cyklus
Toto je nekonecny cyklus
Toto je nekonecny cyklus
Toto je nekonecny cyklus
Toto je nekonecny cyklus
Toto je nekonecny cyklus
Toto je nekonecny cyklus
Toto je nekonecny cyklus
Toto je nekonecny cyklus
Toto je nekonecny cyklus
Toto je nekonecny cyklus
```

2. Ohraniceny cyklus

Vytvorte metodu

```
void ohranicenyCyklus(int start, int end)
```

v ktorej implementujete for cyklus so zaciatkom start a koncovou iteraciou end. Cyklus bude vypisovat do konzoly retazec "ohraniceny rastuci/klesajuci cyklus, i=5", kde za cislo "5" dosadite cislo iteracie cyklu.

Vzorovy I/O:

start = 0, end = 2

```
Ohraniceny rastuci cyklus, i=0
Ohraniceny rastuci cyklus, i=1
Ohraniceny rastuci cyklus, i=2
```

Taktiez overte, ci start je vacsi/mensi ako end a podla toho cyklus modifikujte. Na overenie vytvorte nasledovnu metodu. Premyslite si, ci tato metoda bude public, alebo private.

```
boolean jeStartMensi(int start, int end).
```

Vzorovy I/O:

start = 1, end = 2 -> true

start = 5, end = 2 -> false

3. Hviezdicky

Vytvorte metodu

```
void hviezdicky(int vyska, int sirka, int variacia)
```

v ktorej implementujete vypisovanie hviezdiciiek * do tvarov, ktore su na obrazku. Variacia urcuje, ktory obrazok sa ma vykreslit.

Write a program to print following :

```
i)  *****
    *****
    *****
    *****

ii) *
    **
    ***
    ****
    *****

iii)      *
        **
        ***
        ****
        *****

iv)      *
        ***
        *****
        *****
        *****

v)      1
        222
        33333
        4444444
        555555555

vi)      1
        212
        32123
        4321234
        543212345
```

4. Faktorial

Naprogramujte metodu

```
int suma(int n)
```

ktora vrati sumu cisel od 1 po cislo n. Pouzite ju v triede main so spravou: "suma cisel po 4 je 10".

Vzorovy I/O:

n = 5 -> 15

```
Suma cisel po 5 je 15
```

5. Maximum z cisel

Naprogramujte metodu

```
int maximum(int[] cisla)
```

ktora vrati maximum zo zoznamu cisel. Pouzite enhanced for cyklus. Metodu pouzite v triede main s vypisom spravy: "z cisel {1,2,3,4,5,6,7} je maximum 7".

Vzorovy I/O:

cisla = { 1,2,3,5,0,19,7,-6,18 } -> 19

```
Z cisel [1, 2, 3, 5, 0, 19, 7, -6, 18] je maximum 19
```

Vypis medzery—spravit metodku na to

~~method overload~~

~~Factorial, ale rekurzivne~~

~~printf namiesto println~~

~~skrateny zapis returnov~~

~~ternarny operator na vypisy~~

~~while cyklus~~

Guess the number

exception v jeDelitelne – delenie 0
