

# Objektovo orientované programovanie

Učiteľ:  
**Ing. Jozef Wagner PhD.**

Učebnica:  
<https://oop.wagjo.com/>

# OPG

## Cvičenie 1

1. Inštalácia JDK
2. Inštalácia IDE
3. Vývojový diagram
4. Diagramy ako kód
5. Pseudokód

# Príprava

Pre programovanie je potrebné mať Javu JDK (*Java Development Kit*).

Odporúčaná je distribúcia **Temurin**.

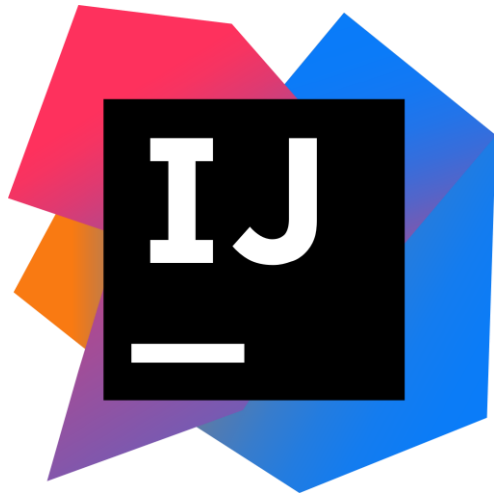
<https://adoptium.net/temurin>

Akú Javu máme:

- príkaz: `java -version`
- nová verzia vychádza každých 6 mesiacov
- posledné stabilné verzie Javy sú 17, 21 a 25



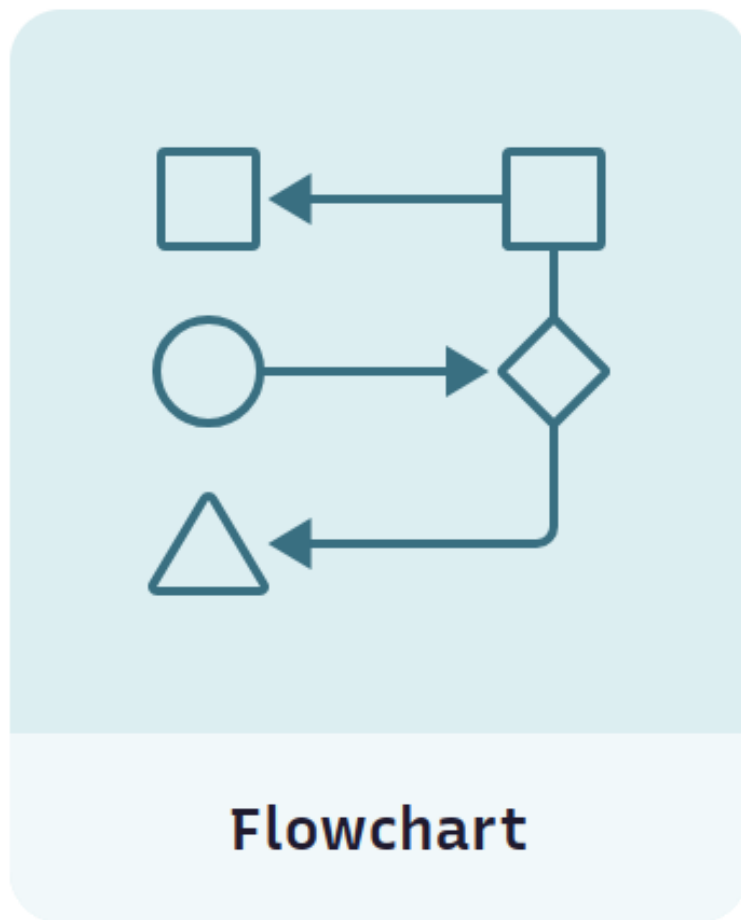
# Príprava



IDE (*Integrated Development Environment*):

- vývojové prostredie
- uľahčuje programovanie
- **IntelliJ IDEA**


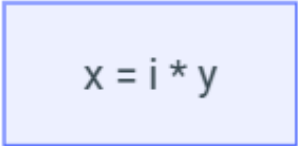
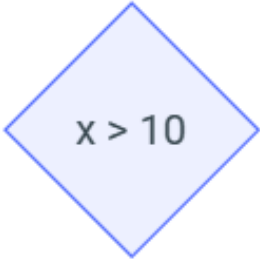
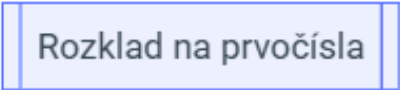
<https://www.jetbrains.com/idea/>



# Úvod do diagramov

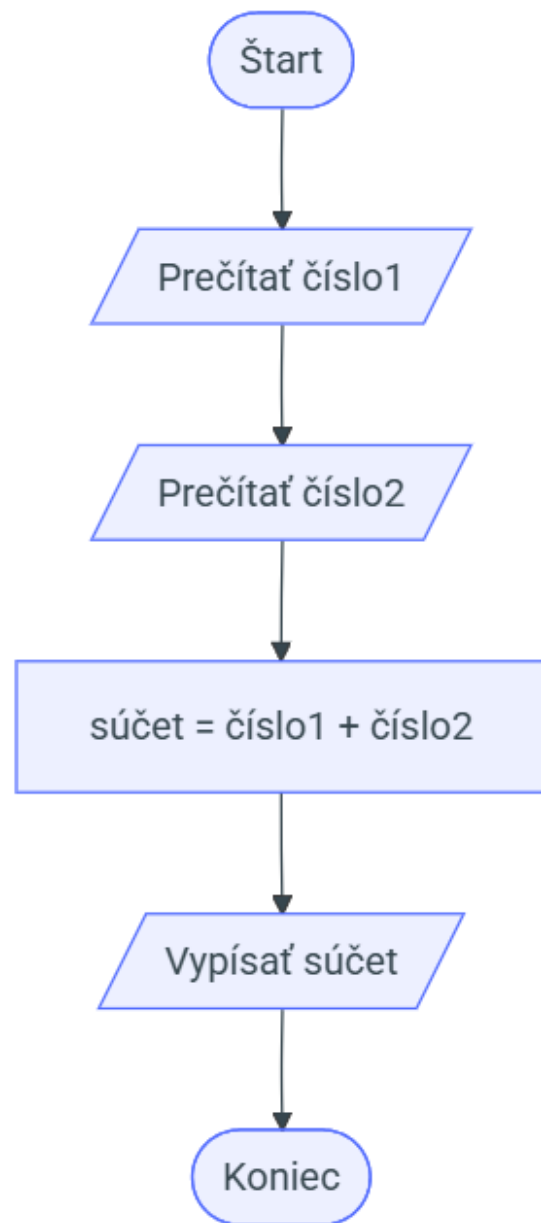
Vývojový diagram znázorňuje kroky procesu alebo algoritmu.

V praxi sa používa v zjednodušenej forme. Dopĺňa iné typy diagramov.

Príklad	Typ symbolu	Popis
	Terminál	Označuje začiatok a koniec procesu. Niekedy nazývaný aj terminátor.
	Vstup/Výstup	Vstup a výstup dát, napr. načítanie čísla, vypísanie výsledku, ...
	Operácia, proces	Bežný príkaz, operácia alebo iný konkrétny krok procesu
	Rozhodovanie	Rozhodovacia funkcia, podmienka. Má viacero výstupov, označených podľa toho, aký výsledok rozhodovania môže nastať.
	Podproces	Podprogram, metóda alebo funkcia, ktorej proces je zakreslený na inom mieste vývojového diagramu



### Príklad 1.1: Súčet dvoch čísel



# Diagramy ako kód

*"Diagram as Code"* alebo *DaC*  
sú diagramy vo forme textu.

Vieme ich dať do  
dokumentácie alebo  
zdrojového kódu.

Používame nástroj  
<http://mermaid.live>



Príklad vývojového diagramu v jazyku Mermaid:

flowchart TD

Start([Štart])

Input1[/Prečítať číslo1/]

Input2[/Prečítať číslo2/]

Sum[súčet = číslo1 + číslo2]

Output[/Vypísať súčet/]

End([Koniec])

Start --> Input1

Input1 --> Input2

Input2 --> Sum

Sum --> Output

Output --> End

# Diagramy ako kód

## Vlastnosti:

- generované automaticky z textu
- ľahšie sa udržiava aktuálny
- vie ho vytvárať a meniť AI

## Nevýhody:

- nezrelosť nástrojov
- musíte vedieť špecifickú syntax
- strata precíznej kontroly vizuálu

# Pseudokód

Pseudokód je zápis algoritmu v prirodzenom jazyku, bez pravidiel. Pomáha nám vysvetliť algoritmus a plánovať kód.

Príklad pseudokódu pre súčet dvoch čísel:

Začať

Prečítať číslo1

Prečítať číslo2

súčet = číslo1 + číslo2

Vypísať súčet

Koniec