

# Popis programu

## Program implementuje následující třídy

- `KvR` - základní třída reprezentující kvadratickou rovnici ve tvaru  $y = ax^2 + bx + c$ 
  - Obsahuje koeficienty `a`, `b`, `c`
  - Poskytuje veřejné gettery pro všechny koeficienty
  - Poskytuje chráněné settery pro všechny koeficienty
- `SpoctiYKvR` - třída pro výpočet hodnot `y` kvadratické rovnice
  - Metoda `calc(KvR, x)` - vypočítá hodnotu `y` pro jednu hodnotu `x`
  - Metoda `calcY(KvR, x[])` - vypočítá hodnoty `y` pro pole hodnot `x`
- `KvREx` - rozšířená třída `KvR`, která navíc obsahuje bod `(x, y)` ležící na křivce
  - Dědí vlastnosti třídy `KvR`
  - Přidává souřadnice `x` a `y`
  - Obsahuje metodu `calcY()` pro výpočet `y` souřadnice z aktuální hodnoty `x`
- `SpoctiYKvRTest` - testovací třída pro `SpoctiYKvR`
  - Testuje výpočet pro jednotlivé hodnoty
  - Testuje výpočet pro pole hodnot
  - Testuje speciální případy (prázdné pole, nulové koeficienty, atd.)

## Vysvětlení implementace

- Proč chráněné (protected) settery v `KvR`?
  - Settery jsou označeny jako chráněné, aby je mohly využívat pouze odvozené třídy, ale ne externí kód
  - Tím je zajištěna zapouzdřenost a integritu třídy
- Vztah `SpoctiYKvR` a `KvR`
  - `SpoctiYKvR` využívá kompozice místo dědičnosti - dostane `KvR` jako parametr
  - Tento přístup umožňuje větší flexibilitu a lepší oddělení zodpovědností

### Implementace `KvREx`

- `KvREx` rozšiřuje `KvR` a přidává souřadnice bodu na křivce
- V konstruktoru se automaticky vypočítá hodnota `y` pro zadané `x`
- Metoda `calcY()` počítá hodnotu `y` podle vzorce  $y = ax^2 + bx + c$

# Testování

- Třída `SpocktiYKvRTest` pokrývá základní funkčnost pro výpočet hodnot
- Testuje jednotlivé výpočty, pole hodnot i speciální případy

Kompletní pokrytí testy Pro kompletní pokrytí tříd `KvR` a `KvREx` testy by bylo třeba:

Pro `KvR`:

- Testovat konstruktor s různými hodnotami koeficientů
- Testovat všechny gettery
- Testovat všechny settery (to by vyžadovalo vytvoření testovací třídy, která dědí z `KvR`, aby měla přístup k `protected` metodám)
- Testovat hraniční hodnoty koeficientů

Pro `KvREx`:

- Testovat konstruktor s různými hodnotami koeficientů a souřadnice  $x$
- Testovat všechny zděděné metody z `KvR`
- Testovat gettery pro souřadnice  $x$  a  $y$
- Testovat metodu `calcY()` pro různé hodnoty koeficientů
- Ověřit, že hodnota  $y$  v konstruktoru je skutečně vypočítána podle vzorce