

# Objektovo orientované programovanie

Učiteľ:  
**Ing. Jozef Wagner PhD.**

Učebnica:  
<https://oop.wagjo.com/>

# OPG

## Cvičenie 7

1. Hra Hangman
2. Trieda Stav
3. Trieda TextoveUI
4. Trieda Hra

# Hangman - obesenec

Zadanie úlohy:

- Hra obesenec v príkazovom riadku
- Na začiatku hra požiada o slovo, ktoré sa bude hádať
- Hráč zadá písmeno, hra ukáže uhádnuté znaky v slove
- Hráč má 5 pokusov, potom hra končí

# Hangman - obesenec

Návrh riešenia:

- Vytvoríme 3 triedy
- Trieda **Stav** bude spravovať stav hry: hádané slovo, uhádnuté znaky a počet ostávajúcich pokusov
- Trieda **TextoveUI** bude mať na starosť užívateľské rozhranie: výpis na obrazovku a načítavanie z klávesnice
- Trieda **Hra** bude mať hlavnú logiku hry a vstupný bod programu

## Úloha 7.1: Nový projekt

V IntelliJ IDEA si vytvoríme nový Java projekt s názvom **Hangman**.

Pri vytváraní vypneme možnosť vytvorenia *Sample code*, ukážkového kódu.

## Úloha 7.2: Trieda Stav

Vytvorte novú triedu **sk.spse.hangman.Stav** a pridajte do nej atribúty a konštruktor podľa príkladu v učebnici.

# Úloha 7.3: Getter metódy

Vytvorte nasledovné getter metódy v triede Stav:

1. `public boolean isVyhra()` - vráti hodnotu atribútu `vyhra`
2. `public boolean isGameOver()` - vráti `true`, ak je počet pokusov menší ako 1 alebo ak sme vyhrali, ináč vráti `false`
3. `public int getOstavaPokusov()` - vráti zostávajúci počet pokusov
4. `public String getHadaneSlovo()` - vráti hádané slovo. Bonus: slovo vráti, iba ak hra skončila, ináč vyhodí výnimku
5. `public String getSlovo()` - vráti slovo, ktoré sme zatiaľ uhádli. Kód tejto metódy musí vytvoriť reťazec `String` z poľa znakov `uhadnuteZnaky`. Môžete na to použiť statickú metódu `String.valueOf()`

# Úloha 7.4: Inštančná metóda

V triede **Stav** vytvorte inštančnú metódu **public boolean hadaj(char tip)**, ktorá bude meniť stav hry podľa zadaného znaku **tip**. V tele metódy napíšte kód, ktorý urobí nasledovné:

1. Vytvorí pomocnú **boolean** premennú **spravnyTip**, inicializuje ju na **false**
2. V cykle prejdite všetky znaky hádaného slova, a ak sa znak zhoduje so znakom tip, pridajte tento znak do poľa **uhadnuteZnaky** na tú istú pozíciu, akú má v hľadanom slove. Ak sa znak zhoduje, nastavte premennú **spravnyTip** na **true** Pri porovnaní znakov použite metódu **Character.toLowerCase(char)**, aby sme pri nebrali ohľad na veľkosť písmen.
3. Ak nenastal správny tip, dekrementuje atribút **ostavaPokusov**
4. Ak je sme uhádli celé slovo, atribút **vyhra** nastaví na **true**. Uhádnutie zistíte porovnaním hádaného slova zo slovom, ktoré vracia metóda **getSlovo()**
5. Metóda vráti hodnotu premennej **spravnyTip**

## Stav

- String hadaneSlovo
- int ostavaPokusov
- char[] uhadnuteZnaky
- boolean vyhra

- + getHadaneSlovo() : String
- + getOstavaPokusov() : int
- + getSlovo() : String
- + hadaj(char) : boolean
- + isGameOver() : boolean
- + isVyhra() : boolean



# Úloha 7.5: Trieda TextoveUI

Vytvorte novú triedu `sk.spse.hangman.TextoveUI`:

```
package sk.spse.hangman;
```

```
import java.util.Scanner;
```

```
public class TextoveUI {  
    private Scanner scanner;
```

```
    public TextoveUI() {  
        this.scanner = new Scanner(System.in);  
    }  
}
```

## Úloha 7.6: Získanie vstupu

Vytvorte nasledovné inštančné metódy v triede `TextoveUI`. V kóde týchto metód použite atribút `scanner` na načítanie hodnôt z klávesnice.

1. `public char ziskajTip()` - vypíše na obrazovku text "Hádaj písmeno: " a načíta jeden znak z klávesnice. Použije metódu `scanner.nextLine()` a potom pomocou metódy `String#charAt()` vráťte prvý znak, ktorý užívateľ zadal.
2. `public String ziskajHadaneSlovo()` - vypíše na obrazovku text "Zadaj slovo, ktoré sa má hádať: " a načíta slovo z klávesnice. Vráti načítané slovo. Po načítaní 'vyčistite' obrazovku napísaním veľkého množstva nových riadkov (znak `'\n'`)

# Úloha 7.7: Vypísanie informácií

Vytvorte nasledovné inštančné metódy v triede `TextoveUI`.

1. `public void vypisIntro()` - vypíše text "Vitaj v hre Obesenec!". Text môžete pekne orámoviť, ak chcete.
2. `public void vypisStavHry(Stav stav)` - do samostatného riadku vypíše "Slovo: " a za ním vypíše zatiaľ uhádnuté slovo `stav.getSlovo()`
3. `public void vypisVysledokHry(Stav stav)` - Ak je `stav.isVyhra()` `true`, vypíše "Vyhrál si!". V opačnom prípade vypíše "Prehral si, hľadané slovo bolo: " a vypíše hľadané slovo.
4. `public void vypisVysledokHadania(char tip, boolean spravnyTip)` - ak je vstupný argument `spravnyTip` `true`, vypíše "Uhadol si ďalší znak!". V opačnom prípade vypíše "Znak %c sa v hľadanom slove nenachádza", pričom vypíše znak `tip`.

## TextoveUI

- Scanner scanner

- + ziskajTip() : char
- + ziskajHadaneSlovo() : String
- + vypisIntro() : void
- + vypisStavHry(Stav) : void
- + vypisVysledokHry(Stav) : void
- + vypisVysledokHadania(char, boolean) : void

# Úloha 7.8: Hlavná trieda Hra

Vytvorte triedu `sk.spse.hangman.Hra` s kódom podľa učebnice

