

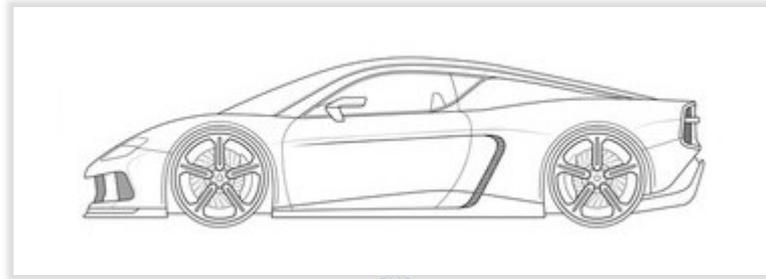
Objektovo orientované programovanie

2. cvičenie

Vladislav Novák

Objekty a triedy

Auto
- farba: Color
- rýchlosť: double
+ naštartuj(): void
+ nastavPlynovýPedál(int): void



trieda – definícia typu
 - typy dát
 - spôsob správania

objekt – používame
 - má stav (hodnoty atribútov)

new

new

new

new



auto1: Auto
farba = červená
rýchlosť = 100.0



auto2: Auto
farba = žltá
rýchlosť = 120.0



auto3: Auto
farba = zelená
rýchlosť = 150.0



auto4: Auto
farba = modrá
rýchlosť = 80.0

Objekty a triedy

```
public class Auto {  
    // .....  
  
}
```

```
static void main(String[] args) {  
    Auto auto1 = new Auto();  
    auto1.nastartuj();  
    // .....  
}
```

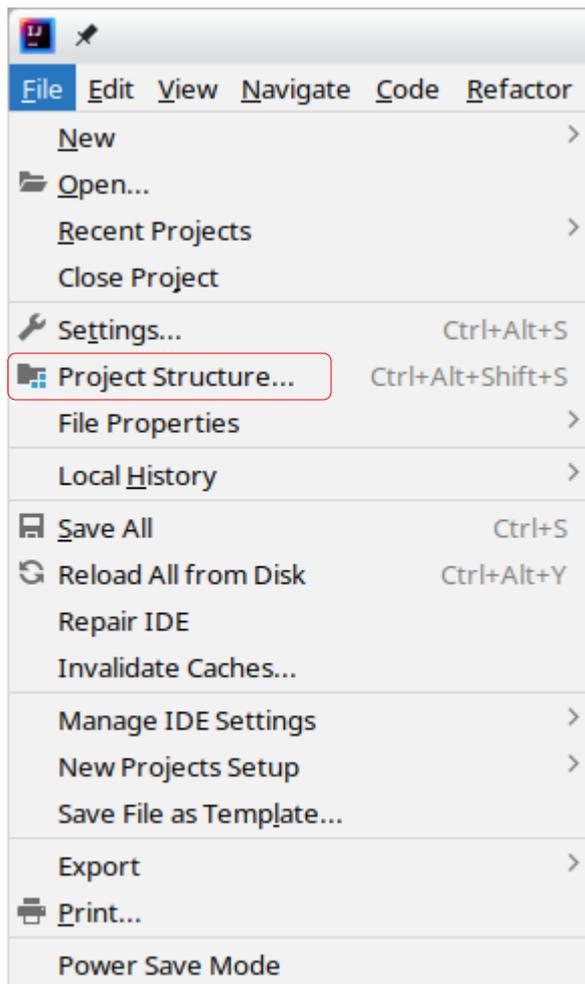
Príklad Counter

- trieda
- objekt
- člen objektu / triedy
- balík
- unit testy

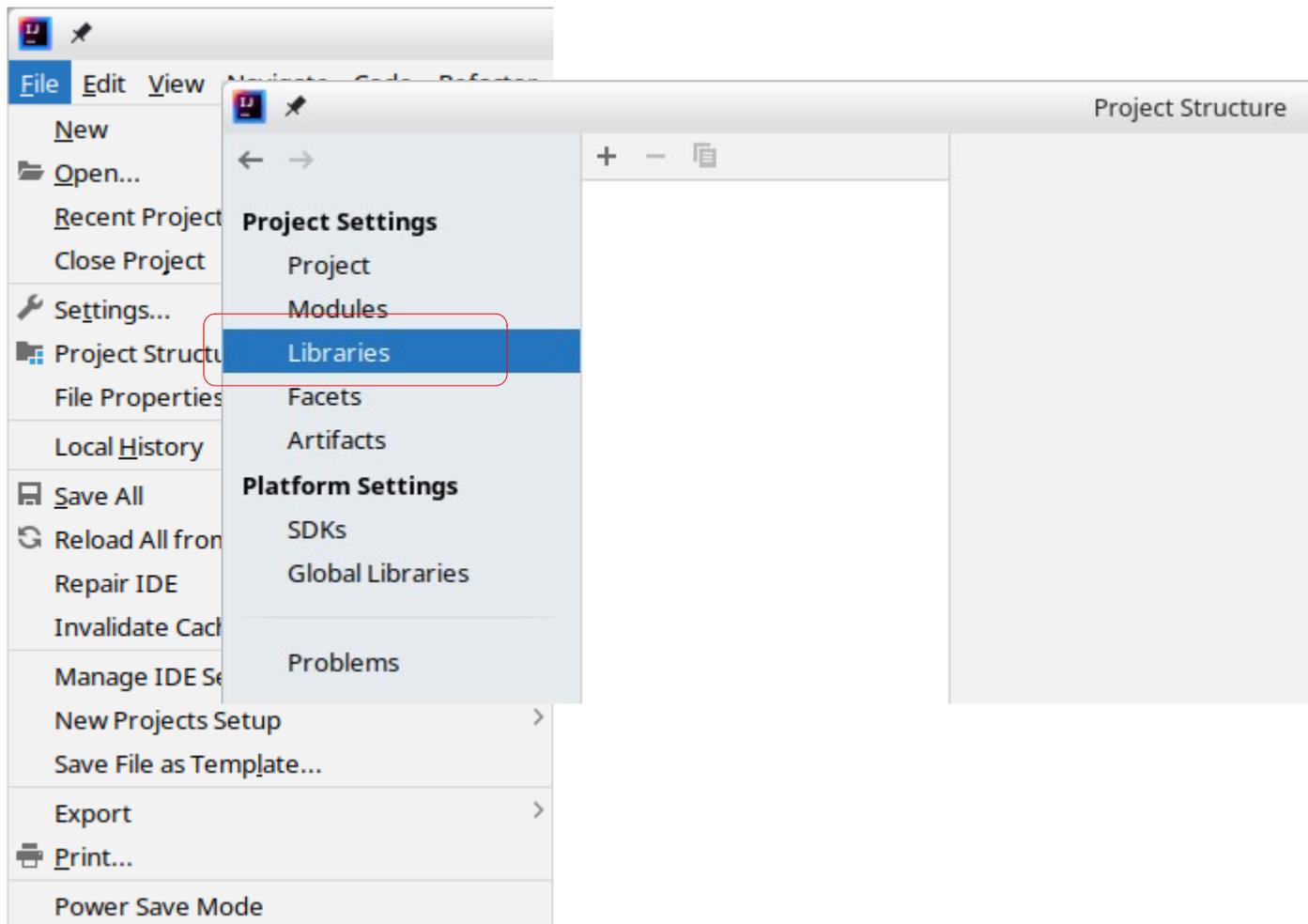
Unit testy

knižnica: junit-jupiter
<https://junit.org>

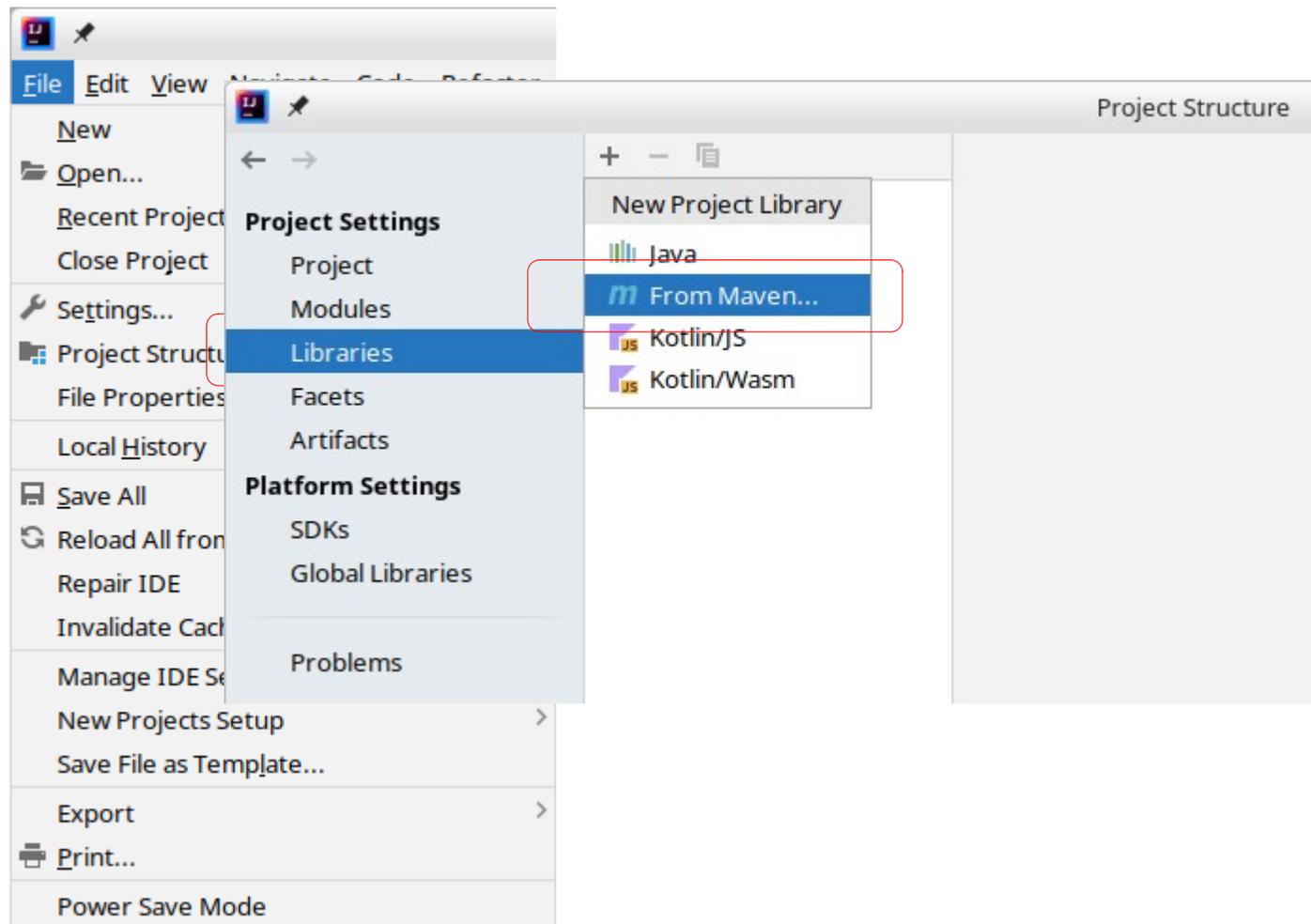
Unit testy



Unit testy



Unit testy



Unit testy

The image shows the IntelliJ IDEA interface with the 'Libraries' menu item selected in the 'Project Structure' dialog. The 'Download Library from Maven Repository' dialog is open, showing the search field containing 'org.junit.jupiter:junit-jupiter:5.11.4'. A blue callout bubble points to the version number '5.12.2' in the search field. The 'Download to' field is set to '/home/novak/workspace/oop_2025/cv05/oop2025_cv05_prog/lib'. The 'Transitive dependencies' checkbox is checked, and the 'OK' button is highlighted.

Project Structure

- New Project Library
- Java
- From Maven...
- Kotlin/JS
- Kotlin/Wasm

Download Library from Maven Repository

org.junit.jupiter:junit-jupiter:5.11.4

Found: 663
Showing: 663

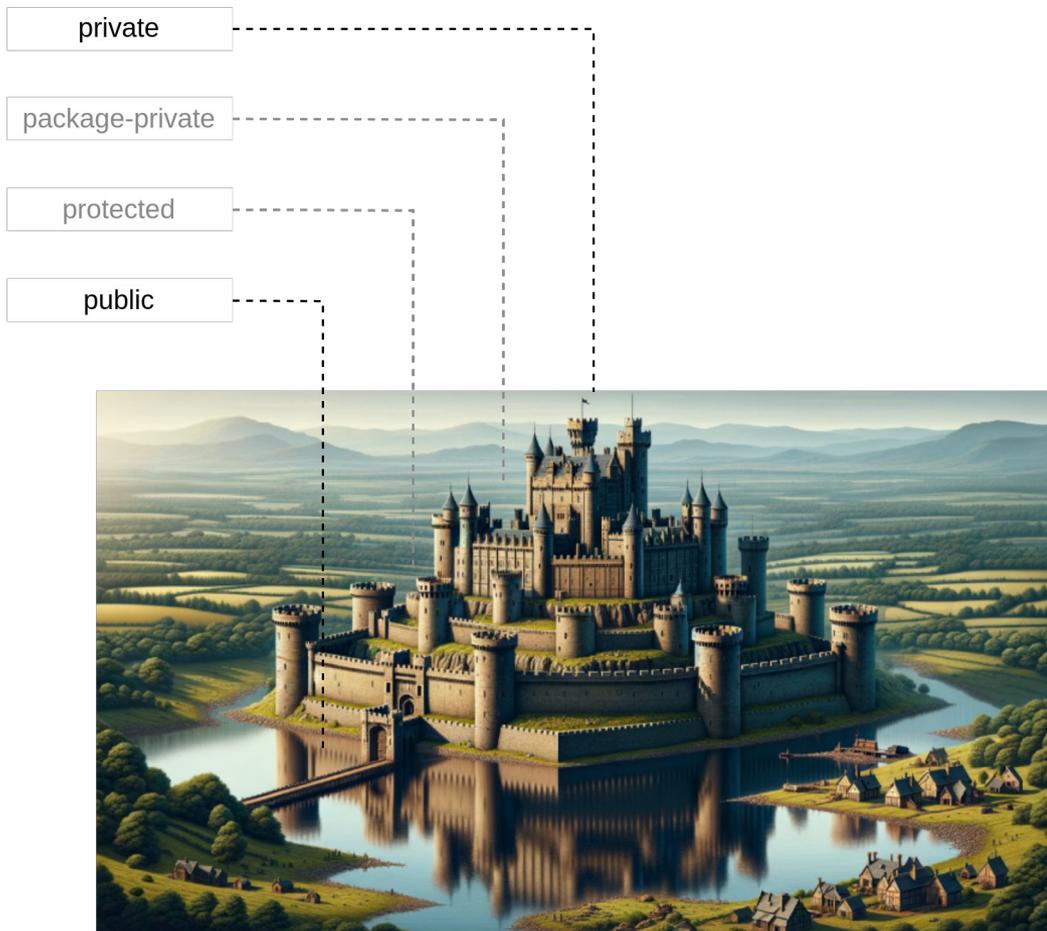
keyword or class name to search by or exact Maven coordinates, i.e. 'spring', 'Logger' or 'ant:ant-junit:1.6.5'

Download to: /home/novak/workspace/oop_2025/cv05/oop2025_cv05_prog/lib

Transitive dependencies Sources Javadocs Annotations

OK Cancel

Zapúzdzrenie (člen triedy/objektu)



Zapúzdrenie (člen triedy/objektu)

```
public class Auto {  
    // clen dostupny len v triede, v ktorej je definovany  
    private .....  
  
    // clen dostupny v baliku, v ktorom je trieda definovana  
    /* bez uvedenia modifikatora: package-private (default) */  
  
    // clen dostupny v podtriedach a v baliku, v ktorom je trieda  
    protected .....  
  
    // clen dostupny vsade  
    public .....  
}
```

Zapúzdrenie (nevnorená trieda)

```
// public  
public class Auto {  
    // .....  
}
```

trieda je prístupná
všade

```
// package-private (default)  
class Auto {  
    // .....  
}
```

trieda je prístupná
len v balíku,
v ktorom je
definovaná

Balíky

```
package sk.stuba.fei.uim.oop.cv02.preteky;
```

```
import org.tinylog.Logger;  
import static java.lang.Math.PI;
```

```
public class Auto {  
    // .....  
}
```

umiestnenie
definovanej triedy

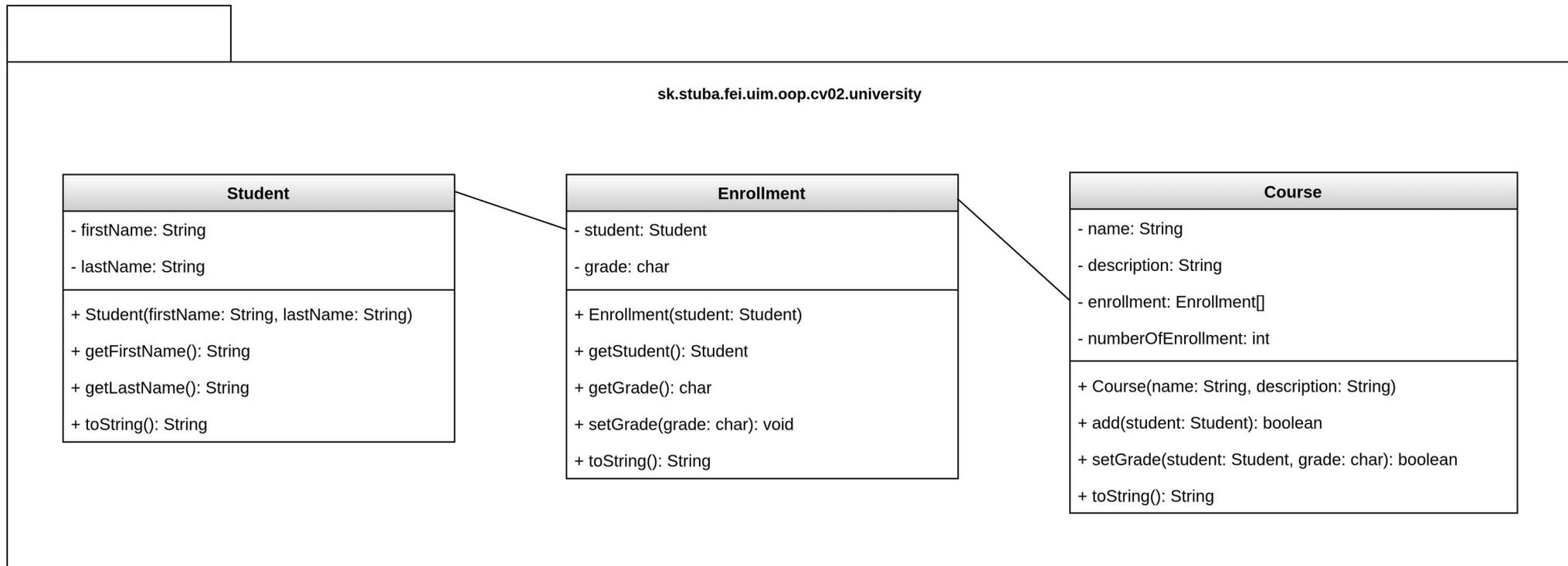
identifikácia
použitých trieda a ich
členov

aby sme nižšie
nemuseli písať celé
názvy

Štruktúra

- Moduly
- Balíky
- Triedy (objekty/inštancie)

Príklad univerzita



import static

Záznam priebehu programu

- (logovanie)
- tinylog
 - <https://tinylog.org>