

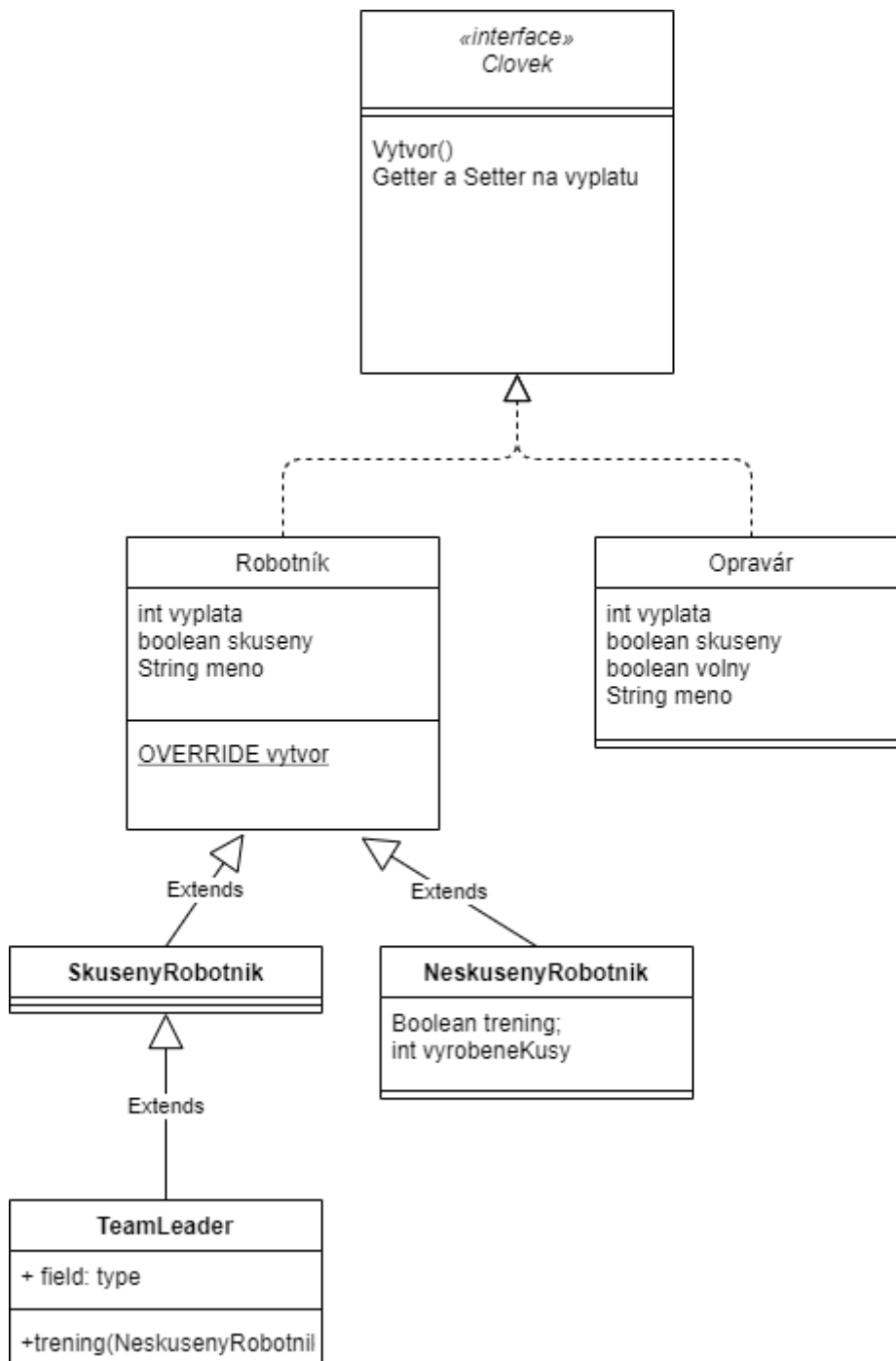
### **Automobilový priemysel (Palubovky do áut )**

Prišla nová objednávka, ktorá najprv smeruje do logistické centra. Skontroluje sa či sa požadovaný tovar nenachádza na sklade. Ak sa objednaný tovar nenachádza na sklade posunie objednávku na výrobného manažéra, ktorý následne skontroluje či je možné splniť časovú požiadavku a množstvo materiálu potrebného na výrobu. Pri nezrovnalostiach sa vrátia požiadavky na logistické centrum na predĺženie času alebo na doobjednanie materiálu. Pri doobjednaní smerujú požiadavky na účtovnícke centrum na sprístupnenie financií.

Po odsúhlasení sa pripíšu peniaze a objednávka smeruje do výroby kde výrobný manažér prideli prácu voľným a práce schopným pracovníkom. Pracovník je primárne zaučený na jeden produkt ale môže pomôcť aj na inom len mu to bude dlhšie trvať. Taktiež môže набраť aj nových pracovníkov ale aj prepustiť. Po výrobe smeruje tovar na kontrolu kvality či je tovar možné poslať. Po schválení kvality, tovar je zabalený a poslaný.

Program bude riešiť vyššie spomínané udalosti.

V zdrojových súboroch ukážte aspoň 1 hierarchiu dedenia tried (3 vrstvovú).



```
public interface Clovek {
    void setVyplata(int ciastka);
    int getVyplata();
    void vytvor(boolean skusenost);
}
```

```
public class Opravar implements Clovek {  
    protected int vyplata;  
    protected boolean skuseny;  
    protected boolean volny;  
    protected String meno;
```

```
public abstract class Robotnik implements Clovek{  
    protected int vyplata;  
    protected boolean skuseny;  
    protected String meno;
```

```
public class SkusenyRobotnik extends Robotnik{
```

```
public class NeskusenyRobotnik extends Robotnik {  
    protected boolean trening= false;
```

```
public class TeamLeader extends SkusenyRobotnik {
```

Od triedy „Clovek“ dedí vlastnosti „SkusenyRobotnik“ a „NeskusenyRobotnik“. od skúseného dedí „TeamLeader“, ktorý bude môcť povýšiť neskúseného robotníka na skúseného. Dedenie je od minula už 3 vrstvové.

## Ukážte využitie polymorfizmu= aspoň jedno prekonávanie metódy. vo vybudovanej hierarchii tried

Toto prekonávanie bolo ukázané pri prezentácii na hodine, toto sa už v projekte nenachádza.

```
@Override
public void vytvor(boolean skusenost) {
    this.skusenost = skusenost;
    if(skusenost){
        this.vyplata = (new Random().nextInt( bound: 3)+8)*100;
    }else{
        this.vyplata = (new Random().nextInt( bound: 3)+5)*100;
    }

    int leftLimit = 65; // písmeno 'A'
    int rightLimit = 90; // písmeno 'Z'
    int targetStringLength = 3;
    Random random = new Random();

    this.meno = random.ints(leftLimit, randomNumberBound: rightLimit + 1)
        .limit(targetStringLength)
        .collect(StringBuilder::new, StringBuilder::appendCodePoint, StringBuilder::append)
        .toString();
}
```

Nové prekonávanie nastáva pri funkcií „vykonaj“. Trieda Opravár, ktorá pri tejto funkcií opravuje stroje a robotníci ktorý vyrábajú palubovky.

Trieda Robotník:

```
@Override
public void vykonaj(String model) {
    if (model.equals("VW")){
        sklad.setPalubovkaVW(1);
    }else if (model.equals("Audi")){
        sklad.setPalubovkaAudi(1);
    }
}
```

Trieda Opravar:

```
@Override
public void vykonaj(String model) {
    if (model.equals("VW")){
        strojVW.setFunkcnost(true);
    }else if (model.equals("Audi")){
        strojAudi.setFunkcnost(true);
    }
}
```

**Ukážte použitie agregácie (aspoň jeden atribút je typu trieda).**

Agregácia je vyriešená Singleton triedou sklad. Trieda „Firma“ si vypýta inštanciu sklad.

```
Sklad sklad = Sklad.getInstance();
```

Vytváranie v triede „Sklad“.

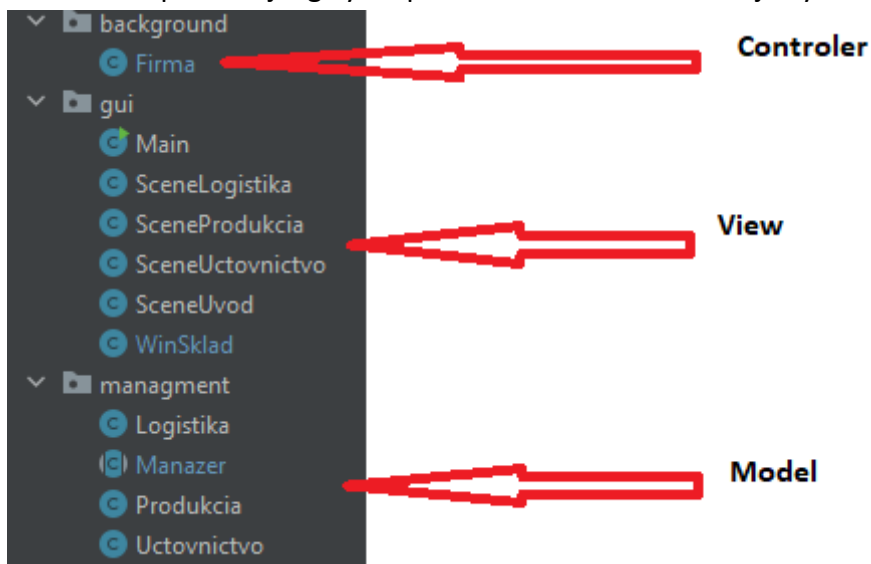
```
static Sklad sklad = new Sklad();
public static Sklad getInstance() { return sklad; }
```

Ďalšia agregácia pri opravárovi, ktorý má na starosti stroje.

```
Stroj strojVW = new Stroj( meno: "VW");
Stroj strojAudi = new Stroj( meno: "Audi");
```

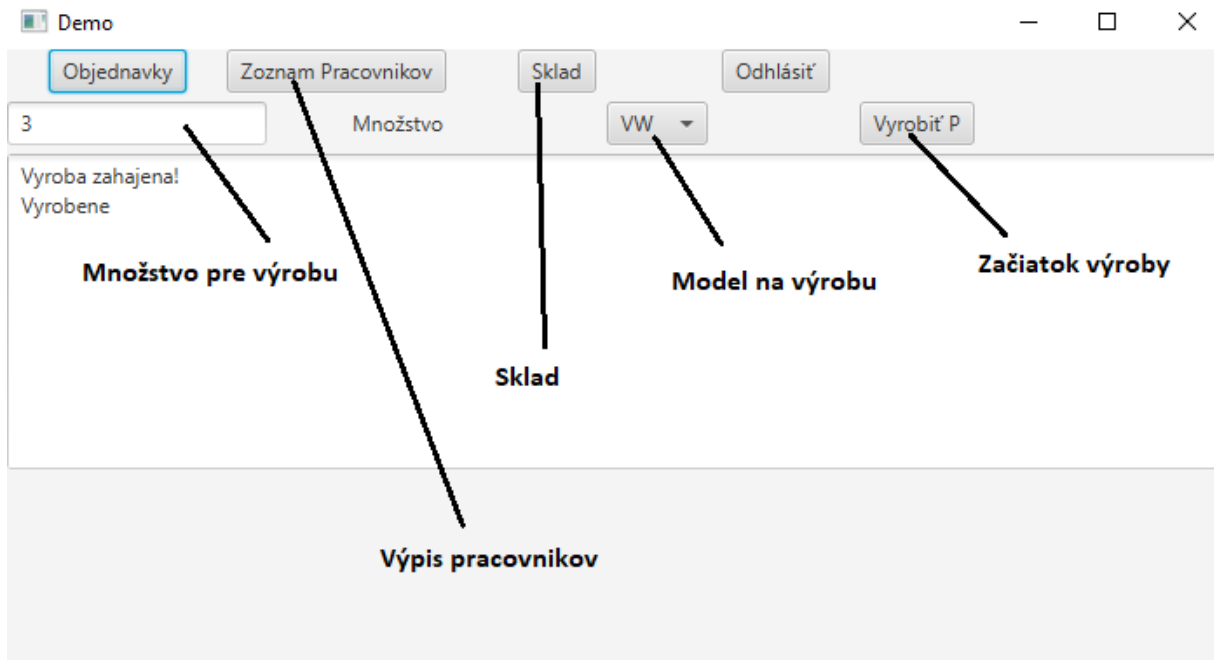
**Ukážte oddelenie aplikačnej(business) logiky od používateľského rozhrania.**

Oddelenie aplikačnej logiky od používateľského rozhrania je vyriešená modelom MVC.



## Funkcionalita

Zatiaľ je funkčný iba produkčný manažér.



Tlačidlo objednávky bude funkčné až pri serializácii.