

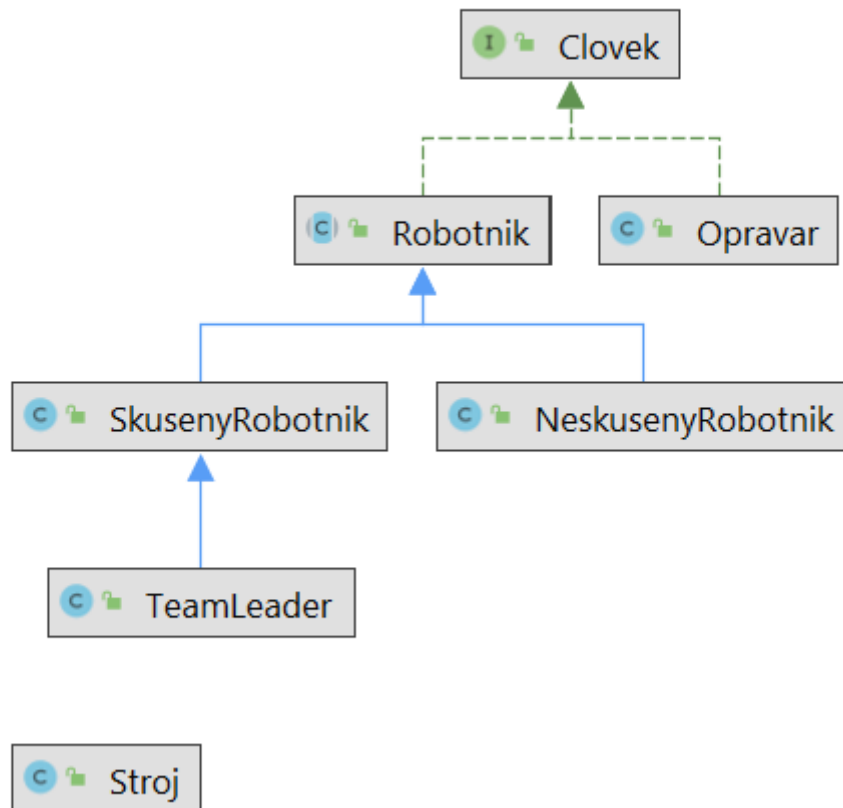
Automobilový priemysel (Palubovky do áut)

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 nastaví sa výrobný plán pre produkčného manažéra, ktorý následne skontroluje množstvo materiálu potrebného na výrobu.

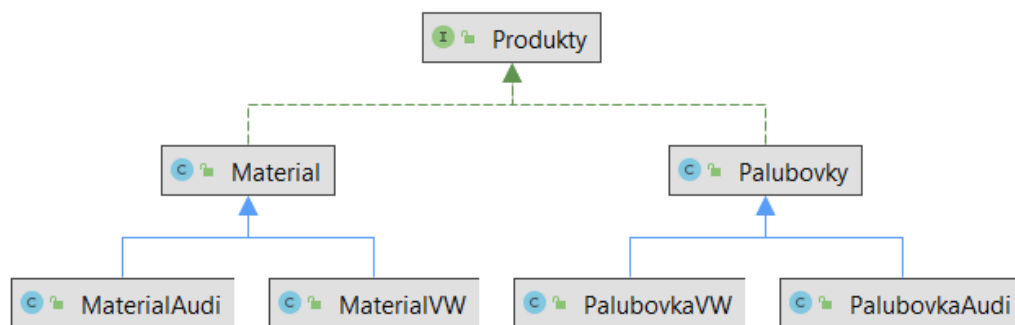
Produkčný manažér posunie do výroby počty produktov, ktoré pracovníci vyrábajú. Pracovník podľa skúseností vyrába finálne produkty. Skúsený pracovník sa nezapája do práce ktorú zvládne samotný neskúsený. Neskúsený robotník berie znalosti z výroby a neskôr môže byť povýšený. Taktiež popri výrobe sa môže stať že sa stroj pokazí a musí ho ísť opravár opraviť.

Program bude riešiť vyššie spomínané udalosti.

2 hierarchie dedenia tried (3 vrstvová) vrátane použitia rozhraní.



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.



Materiál a Palubovky dedia od produktov základné funkcie. V 2 vrstve sa nachádzajú funkcie na upozornenie svojich sledovateľov.

Polymorfizmu

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

Trieda Robotnik:

```
@Override
public String vykonaj(String model, PalubovkaVW vw, PalubovkaAudi audi) {
    boolean vyrobil = true;
    if (model.equals("VW")) {
        if (skusený) {
            if (materialVW.getPocet() >= 2) {
                vw.pridajPocet(cislo: 2);
                materialVW.odoslanie(cislo: 2);
            } else {
                vyrobil = false;
            }
        } else {
            if (materialVW.getPocet() >= 1) {
                vw.pridajPocet(cislo: 1);
                materialVW.odoslanie(cislo: 1);
            } else {
                vyrobil = false;
            }
        }
    }
    materialVW.upovdeomSledovateľov();
    vw.upovdeomSledovateľov();
}
```

Trieda Opravar:

```
@Override
public String vykonaj(String model, PalubovkaVW vw, PalubovkaAudi audi) {

    StringBuilder sprava = new StringBuilder();
    if (strojVW.isFunkcnost()){
        strojVW.setFunkcnost(true);
        sprava.append("POKAZENÝ STROJ VW, OPRAVUJEM \n");
        sprava.append("OPRAVENÉ\n");
    }
    if (strojAudi.isFunkcnost()){
        strojAudi.setFunkcnost(true);
        sprava.append("POKAZENÝ STROJ AUDI, OPRAVUJEM \n");
        sprava.append("OPRAVENÉ\n");
    }

    return sprava.toString();
}
```

Použitie agregácie a zapúzdrenie

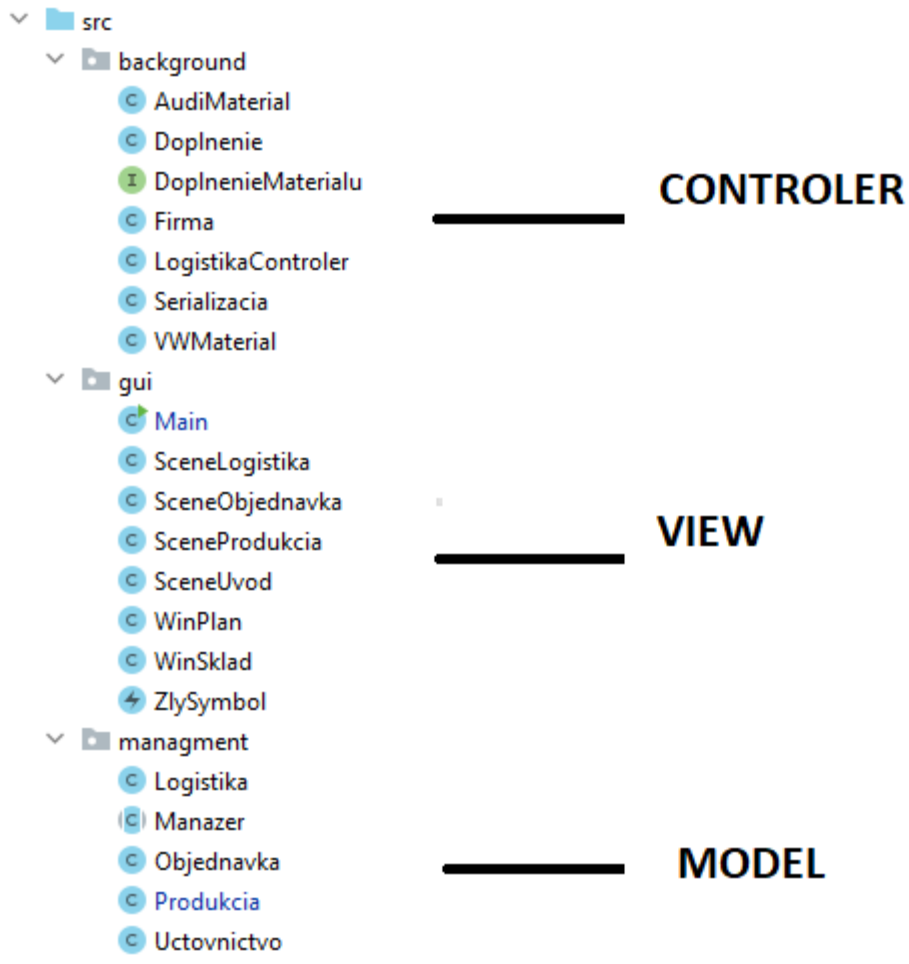
```
public class Produkcia extends Manazer {

    private TeamLeader teamLeader;
    private final Stroj strojVW = new Stroj( meno: "VW");
    private final Stroj strojAudi = new Stroj( meno: "Audi");
    private Opravar opravavar ;
}
```

Agregácia v produkcií. Nik nebude používať tieto objekty iba funkcie v produkcií preto sú private.

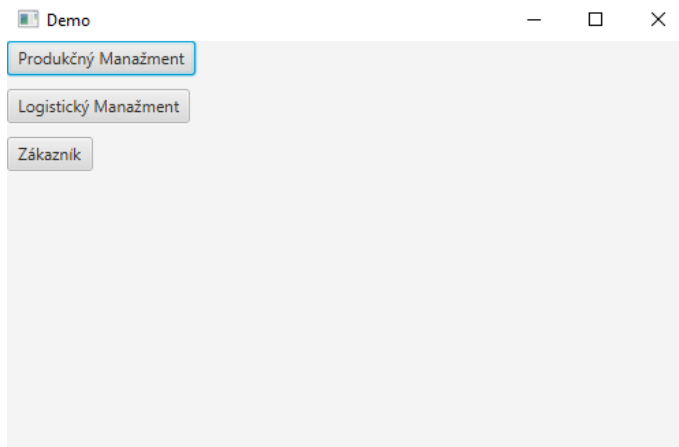
Oddelenie aplikačnej(business) logiky od používateľského rozhrania a organizácia balíkov.

Oddelenie aplikačnej logiky od používateľského rozhrania je vyriešená modelom MVC.

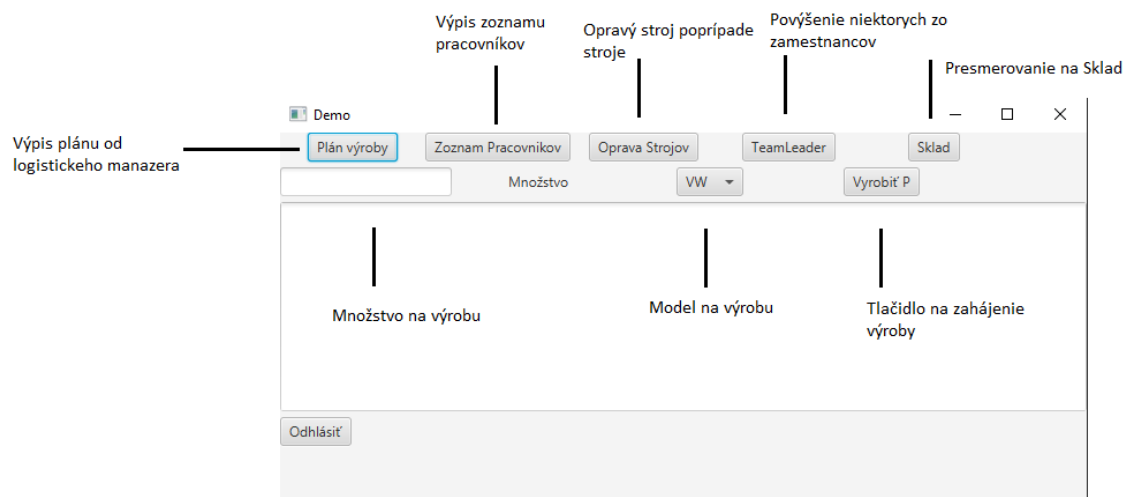


Funkcionalita (Grafické rozhranie – ďalšie kritérium)

Úvod



Produkčný manažér



Logistický manažér

Počet materiálu na objednanie Pre ktorý model sa doplní material Objednanie materiálu

Demo — □ ×

Počet Materiálu VW Objednat Material

Presmerovanie na Sklad Sklad

Objednávky Č. Objednávky Odoslať objednávku

Zadat' plán

Zadá plán pre produkciu Zobrazí objednávky od zákazníka Číslo objednávky na expedíciu Tlačidlo na odoslanie objednávky

Odhlásiť

Zákazník

Demo — □ ×

Meno firmy — Meno pre koho je dodávané

Značka VW Aký model paluboviek

Množstvo — Množstvo

Objednaj Objednanie

Odhlásiť

Ďalšie kritéria

Návrhový vzor Observer

Subjekty:

```
MaterialAudi      public class MaterialAudi extends Material implements Serializable {
MaterialVW
PalubovkaAudi     transient private List<SledovatelSkladu> sledovatelA = new ArrayList<>();
PalubovkaVW

    public MaterialAudi () { super( typ: "Audi"); }
    public void pridajSledovateľaAudi(SledovatelSkladu sledovatel){
        this.sledovatelA.add(sledovatel);
        sledovatel.upovedomAudi();
    }

    public void upovedomSledovateľov(){
        for(SledovatelSkladu sledovatel : this.sledovatelA){
            sledovatel.upovedomMaterialAudi();
        }
    }
}
```

Observer:

```
public interface SledovatelSkladu {
    void upovedomVW();
    void upovedomAudi();
    void upovedomMaterialVW();
    void upovedomMaterialAudi();
}

public class Sklad implements SledovatelSkladu, Serializable {

    PalubovkaVW palubovkaVW;
    PalubovkaAudi palubovkaAudi;
    MaterialVW materialVW;
    MaterialAudi materialAudi;

    public Sklad(PalubovkaVW vw, PalubovkaAudi audi, MaterialVW vwMat, MaterialAudi audiMat){
        this.palubovkaVW =vw;
        this.palubovkaAudi = audi;
        this.materialAudi = audiMat;
        this.materialVW = vwMat;
    }
}
```


Návrhový vzor Factory:

Návrhový vzor factory je použitý pri používaní materiálu a jeho doplnení.

```
public interface DoplnenieMaterialu {  
    String doplnenie(int pocet);  
    void pouzitie(int pocet);  
}
```

```
public class VWMaterial implements DoplnenieMaterialu{  
    MaterialVW materialVW;  
  
    public VWMaterial(MaterialVW vw) { this.materialVW = vw; }  
  
    @Override  
    public String doplnenie(int pocet) {  
        if(this.materialVW.getPocet() > 200){  
            return "Sklad je plny";  
        }else{  
            if (this.materialVW.getPocet()+pocet > 200){  
                this.materialVW.pridajPocet( cislo: 200 - this.materialVW.getPocet());  
                return "Sklad doplneny naplno";  
            }else {  
                this.materialVW.pridajPocet(pocet);  
            }  
        }  
  
        materialVW.upovedomSledovateLov();  
        return "Doplnene o " + pocet + " materialu VW";  
    }  
  
    @Override  
    public void pouzitie(int pocet) {  
        this.materialVW.odoslanie(pocet);  
    }  
}
```

```
public class AudiMaterial implements DoplnenieMaterialu{  
    MaterialAudi audi;  
  
    public AudiMaterial(MaterialAudi audi) { this.audi = audi; }  
  
    @Override  
    public String doplnenie(int pocet) {  
  
        if(this.audi.getPocet() > 200){  
            return "Sklad je plny";  
        }else{  
            if (this.audi.getPocet()+pocet > 200){  
                this.audi.pridajPocet( cislo: 200 - this.audi.getPocet());  
                return "Sklad doplneny naplno";  
            }else {  
                this.audi.pridajPocet(pocet);  
            }  
        }  
  
        audi.upovedomSledovateLov();  
  
        return "Doplnene o " + pocet + " materialu Audi";  
    }  
  
    @Override  
    public void pouzitie(int pocet) {  
        ...  
    }  
}
```

Vlastná výnimka

```
public void alert (){  
    Alert a = new Alert(Alert.AlertType.ERROR);  
    a.setTitle("Chyba");  
    a.setContentText("Nezadaný správny symbol");  
    a.showAndWait();  
}
```

```
try {  
    int number;  
  
    if (Pattern.matches( regex: "[a-zA-Z]+", pocet)) {  
        throw new ZlySymbol();  
    }  
  
    number = Integer.parseInt(pocet);  
    String model = vyber.getValue();  
    logistika.pridajObjednavku(name,model,number);  
}  
catch (ZlySymbol ex){  
    ex.alert();  
    return "Neobjednane";  
}
```

Vlastná výnimka funguje na vyhodenie chybového okna pri zadaní zlého symbolu, keď sa na miesto čísla zadá text.

Serializácia

Serializácia je použitá pri sklade paluboviek a materiálu.

```
private void nacistaj() throws ClassNotFoundException, IOException {
    ObjectInputStream in = new ObjectInputStream(new FileInputStream("src/palubovkaAudi.out"));
    this.serializacia = (Serializacia) in.readObject();
    palubovkaAudi.pridajPocet(serializacia.getPocetPaluboviekAudi());
    palubovkaVW.pridajPocet(serializacia.getPocetPaluboviekVW());
    materialAudi.pridajPocet(serializacia.getPocetMaterialuAudi());
    materialVW.pridajPocet(serializacia.getPocetMaterialuVW());
    System.out.println(serializacia.getPocetMaterialuAudi() + " " + serializacia.getPocetMaterialuVW() + " ");
    in.close();
}
```

Lambda symbol

Lambda symbol je využitý pri výpise pracovníkov.

```
this.skuseny.forEach((n)-> {sprava.append(n.getMeno());});
this.neskuseny.forEach((n)-> {sprava.append(n.getMeno());});
```