Tomáš Tomčány
ID: 110980
Cvičenie: Utorok 18:00

Proces objednávky pizze

Zámerom projektu je simulácia procesu objednania pizze. Program prejde procesom od objednania, až po donášku samotnej objednávky. Systém začne so skladaním jednotlivých pízz zákazníkom, kde si môže vybrať s rôznych veľkosti, toppingov, okrajov a základu. Každá časť bude mať určitú cenu a s poskladaním sa vytvorí celková cena pizze. V rámci objednávky budú taktiež implementované zľavové kupóny a grátis akcie. Následne po doskladaní objednávky si môže zákazník zvoliť dokončenie objednávky, kde bude mať na výber, či chce objednávku mať na donášku, alebo si ju vyzdvihnúť sám. V prípade donášky sa zvýši cena celkovej objednávky a následne sa vykoná simulácia času procesu výroby a donášky môže sledovať čas, ktorého veľkosť bude ovplyvnená rôznymi udalosťami a typmi ľudí podieľajúcimi sa na procese výroby a donášky pizze. Po týchto kalkuláciách sa zákazníkovi zobrazí výpis(účet) a čas, dokým si bude môcť zákazník pizzu buď vyzdvihnúť, alebo čas donášky.

Tomáš Tomčány
ID: 110980
Cvičenie: Utorok 18:00

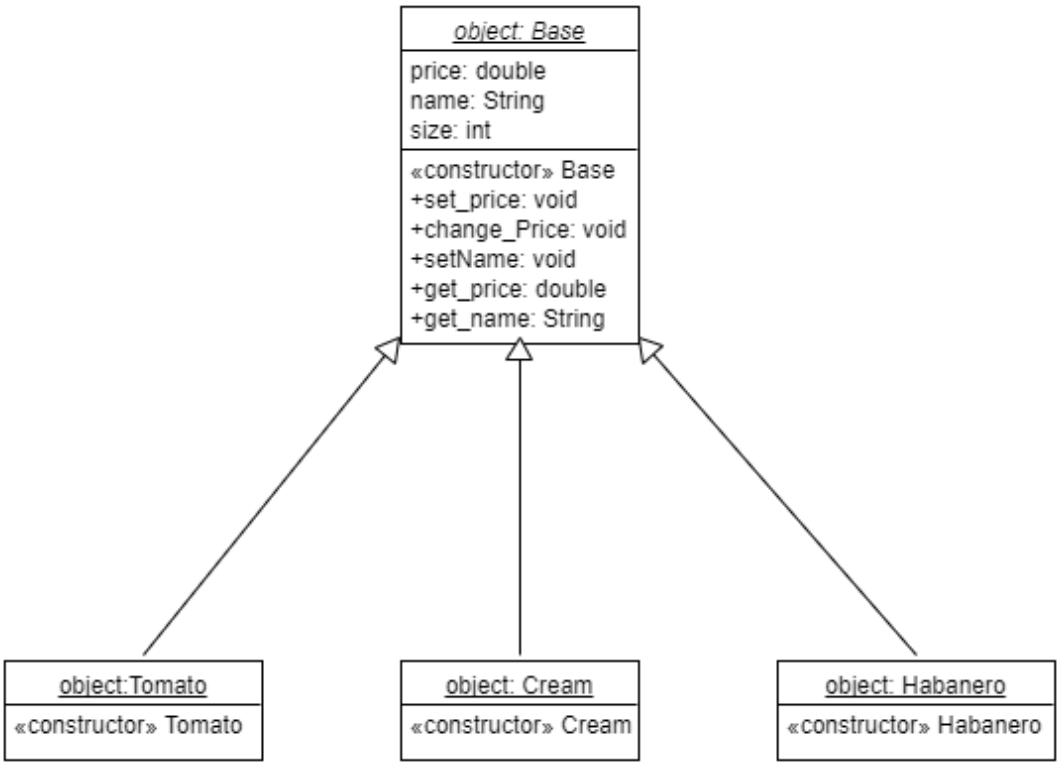
Príklad použitia dedenia

Dedenie môžeme využiť v prípade, že existujú rôzne typy určitej triedy, pri ktorých chceme aby zdedili atribúty a metódy ich nadtriedy.

Príklad v programe:
Pizza sa skladá zo základu, okraju a toppingov. So spomenutých častí pizze môžu byť vytvorené ich rôzne typy, napr. základ môže byť paradajkový, smotanový, habanero a tieto triedy zdedia atribúty a metódy ich nadtriedy Základ s prípadným zmenením hodnôt atribútov. V kóde určime dedenie pomocou kľúčového slova extends.

```
public class Tomato  extends Base {
```

Diagram dedenia:



Tomáš Tomčány
ID: 110980
Cvičenie: Utorok 18:00

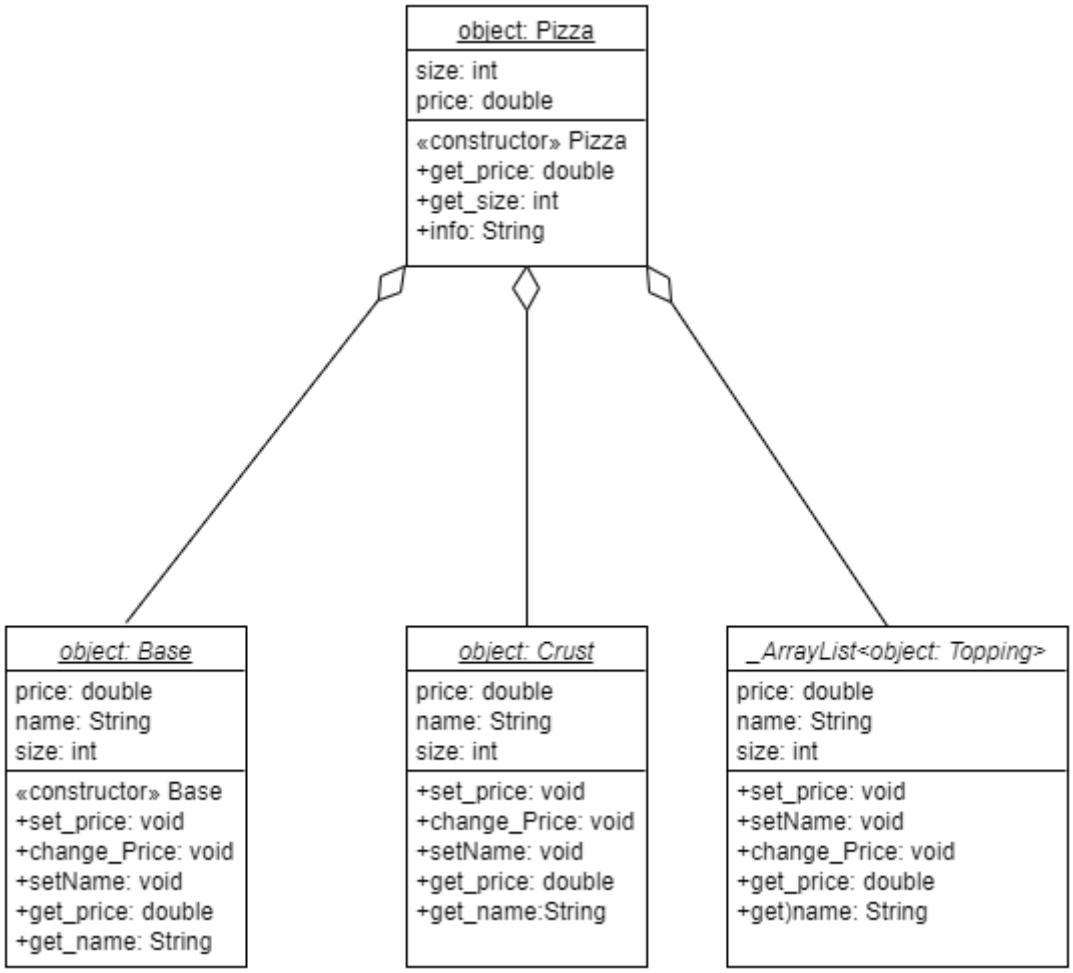
Príklad použitia agregácie

V prípade, že trieda má v atribútoch referenciu na inú triedu, je to agregácia.

Príklad v programe:
V kóde má trieda Pizza referencie na triedy Base, Crust a ArrayList<Topping>. Relácia medzi nimi je has-a triedy môžu od seba existovať nezávisle.

```java
public class Pizza {
    private final int size;
    private double price;
    private final Base base;
    private final Crust crust;
    private final ArrayList<Topping> toppings;
```

Diagram agregácie:



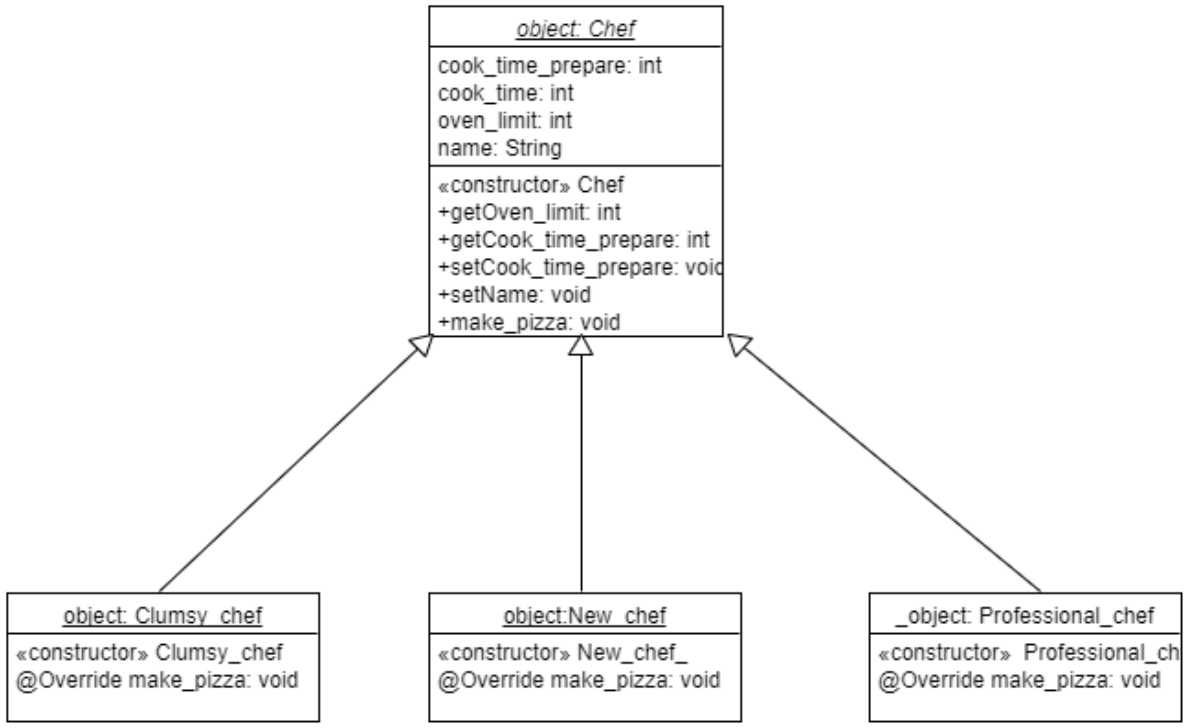
Tomáš Tomčány
ID: 110980
Cvičenie: Utorok 18:00

Príklad použitia polymorfizmu

Polymorfizmus sa môže vyskytnúť pri dedení, kde môže nastať method overriding, ktorá mení čo sa bude vykonávať pri volaní tejto metódy, na základe toho, na ktorý objekt nejakej triedy sa volá.

Príklad v programe:
V kóde máme abstraktnú nadtriedu Chef a jej podtriedy Clumsy_chef, New_chef a Professional_chef, ktoré majú metódu make_pizza(Order order), ktorá vykonáva niečo iné na základe triedy. Následne v triede Order máme atribút Chef chef a metódu do_order() v ktorej sa náhodne pridelí do premennej chef nová inštancia objektu typu nejakej podtriedy Chef. Následne sa zavolá chef.make_pizza(this), ktorá na základe toho, aký typ objektu sa v premennej chef nachádza, vykoná metódu podľa definície v tej triede.

Diagram polymorfizmu:

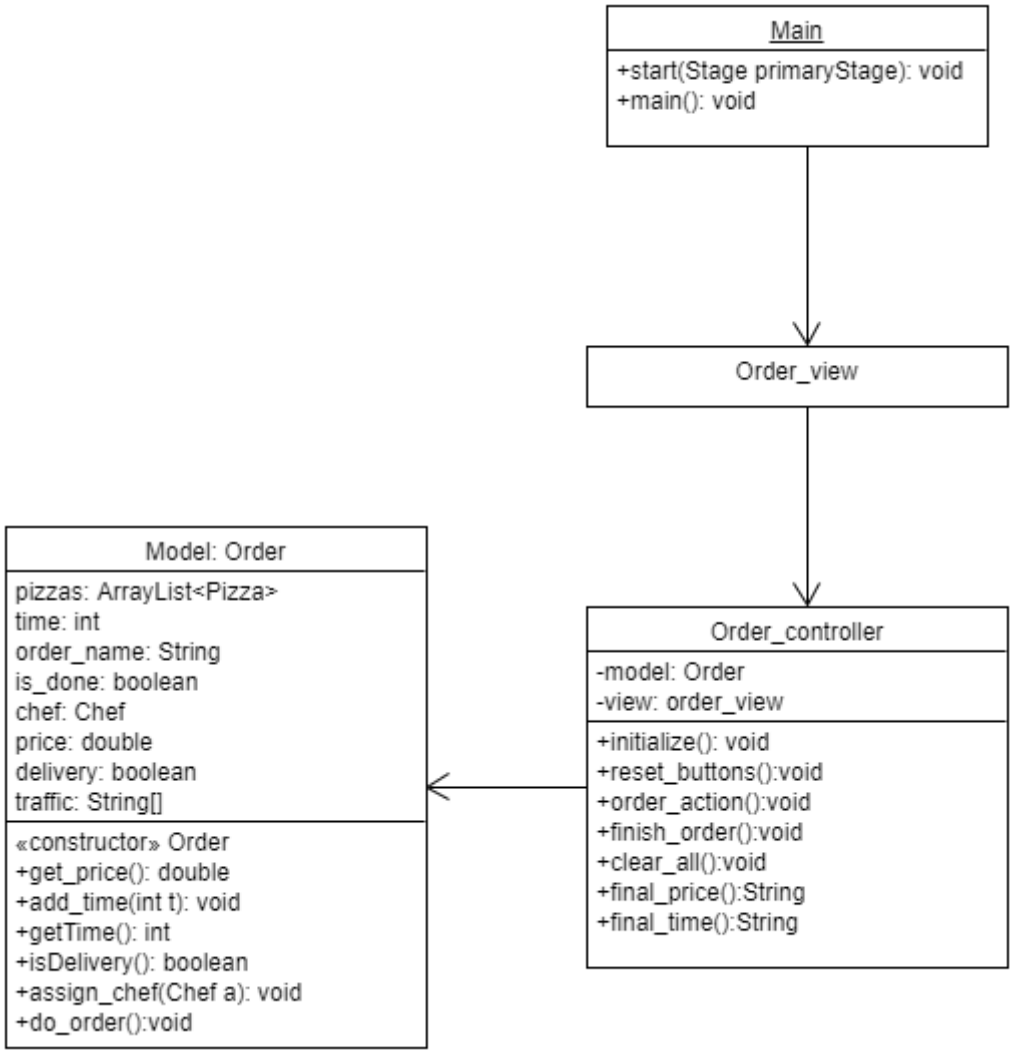


Tomáš Tomčány
ID: 110980
Cvičenie: Utorok 18:00

Oddelenie aplikačnej logiky od uživateľského rozhrania pomocou návrhového rozhrania Model-view-controller

Návrhový vzor MVC je určený na oddelenie aplikačnej logiky od UI vďaka trom častiam. Model – časť aplikačnej logiky, View – uživateľské rozhranie a Controller – prepája prvé 2 časti na základe použitia aplikačnej logiky závislo od interakcie v uživateľskom rozhraní.

Diagram MVC:



Hlavnou triedou je Main v package main.

TODO list

-sklad

-zoznam objednávok(vykonaných/nevykonaných)

Tomáš Tomčány
ID: 110980
Cvičenie: Utorok 18:00

-kritériá