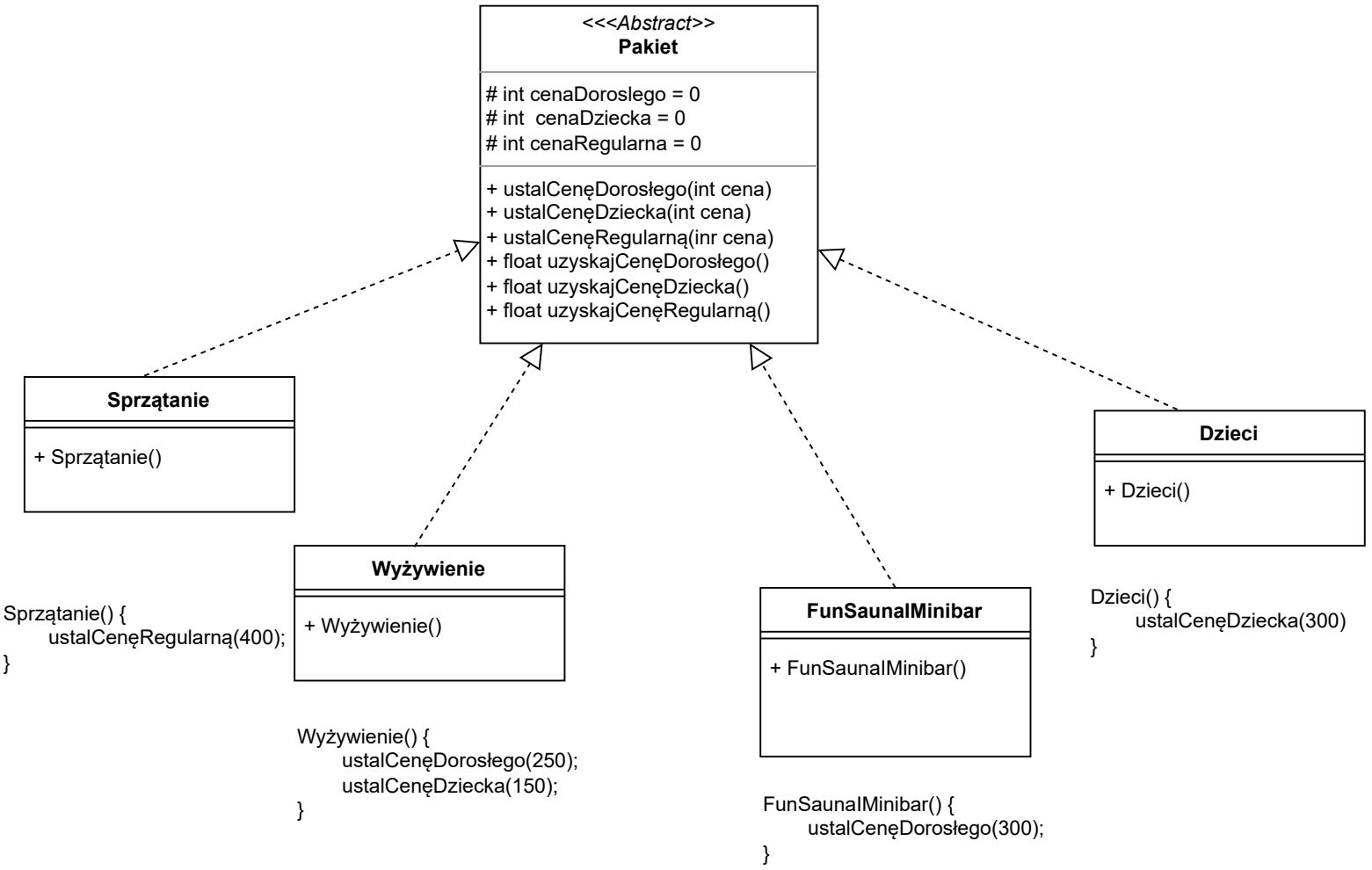
Template Method

DOMINIK GERLACH, 272388

Użyłem tego wzorca do ustalania cen pakietów, ponieważ pakiety mają jeden lub dwa typy cen (cena dla dziecka, dorosłego lub cena regularna). Algorytm ustalania cen dla konkretnego pakietu jest ~~taki~~ identyczny do tych w innych pakietach. Pakiety mogą się jednak różnić wykonywanymi metodami



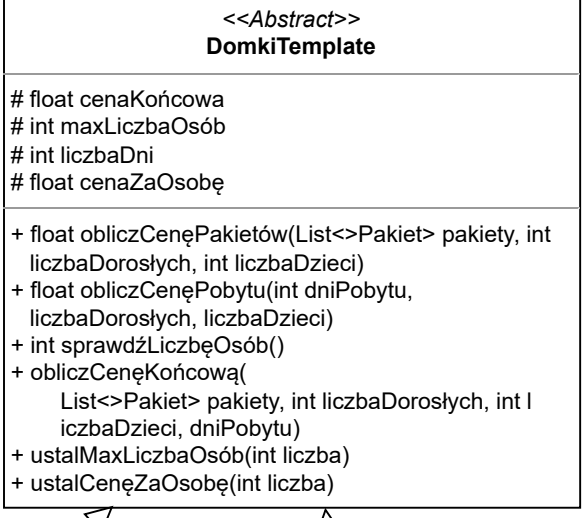
Template Method

Użyłem tego wzorca do obliczania kosztów pobytu dla określonego domku z powodu iż algorytmy obliczające cenę dla konkretnego domku są identyczne, różnią się tylko i wyłącznie parametrami. Wzorzec pozwala na rozszerzenie systemu o kolejne domki

```
float obliczCenęPakietów(List<>Pakiet> pakiety, int
liczbaDorosłych, int liczbaDzieci) {

    float cenaPakietów;
    for (p: pakiety) {
        cenaPakietów += (p.uzyskajCenęDorosłego *
        liczbaDorosłych +  p.uzyskajCenęDziecka *
        liczbaDziecki + p.uzyskajCenęRegularną);
    }

    return cenaPakietów
}
```

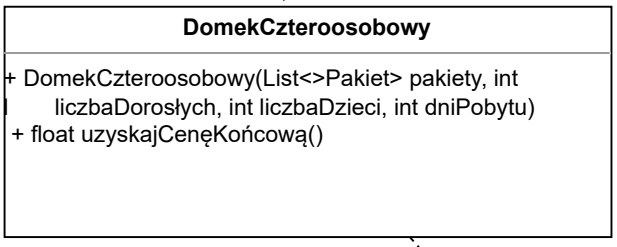
```
obliczCenęKońcową(List<>Pakiet> pakiety, int liczbaDorosłych, int
liczbaDzieci, dniPobytu) {
      cenaKońcowa = obliczCenęPakietów(pakiety,
      liczbaDorosłych, lliczbaDzieci)  + obliczCenę(dniPobytu,
      liczbaDorosłych, liczbaDzieci)
}
```

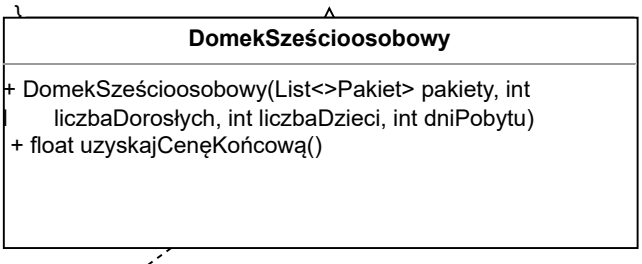
```
float obliczCenęPobytu(int dniPobytu, liczbaDorosłych,
liczbaDzieci) {
      float cenaPobytu = (liczbaDorosłych +
      liczbaDzieci) *cenaZaOsobę * dniPobytu;

      return cenaPobytu;
}
```




```
DomekCzteroosobowy(List<>Pakiet> pakiety, int liczbaDorosłych,
                   int liczbaDzieci, int dniPobytu) {

   ustalMaxLiczbęOsób(4)
   ustalCenęZaOsobę(120)
   if (sprawdźLiczbęOsób(liczbaDorosłych, liczbaDzieci) == 0) {
         obliczCenęKońcową(pakiety, liczbaDorosłych,  liczbaDzieci,
         dniPobytu);
   }
}
```

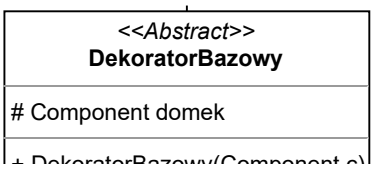
```
int sprawdźLiczbęOsób(liczbaDorosłych, liczbaDzieci) {

      if (liczbaDorosłych + liczbaDzieci > maxLiczbaOsób) {
            throw Exception("Nie można pomieścić w tym
                              domku tylu osób);
            return 1
      }
      return 0
}
```

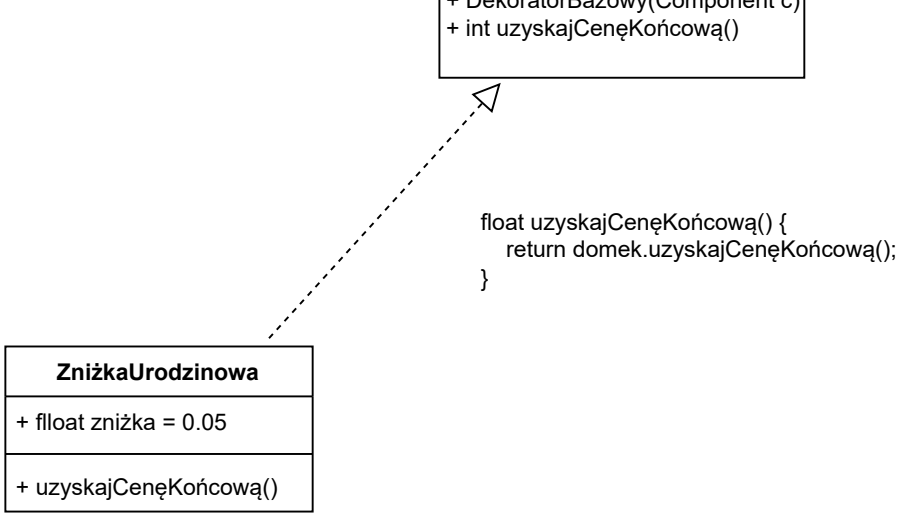




Dekorator

Uzywam dekorator do nabicia zniżki. Świetnie się on do tego nadję, gdyż pozwala na dodawanie nowych własności już istniejącym obiektom. Jest też otwarty na rozszerzenia






```
float uzyskajCenęKońcową() {
      float cena =
            super::uzyskajCenęKońcową()
      cena = (1 - zniżka) * cena
      return cena
}
```

Klient

```
int liczbaDorosłych = 3
int liczbaDzieci = 1
int dniPobytu = 4

List<Pakiet> pakiety = new List<>();

pakiety.add(new Sprzątanie, new Wyżywienie);

DomkiTemplate domek = new DomekCzteroosobowy(pakiety, liczbaDorosłych

, liczbaDzieci, dniPobytu )

DomekZniżka = domekPoZniżce = new ZniżkaUrodzinowa(domek)
```