

## Zadaća 2: Ekološko zbrinjavanje otpada (EZO) - Damir Drempetić

### ***Razlog odabira pojedinog uzorka dizajna***

- Singleton (ParametriSingleton, GeneratorBrojevaSingleton, IspisivacSingleton)

Parametri se smiju koristiti samo jednom, a generator slučajnih brojeva treba biti jedinstven.

- Builder (CitacPopisaBuilder)

Ovisno o pozivima metoda za konstrukciju objekta ulaznih datoteka (popisi ulica, spremnika, vozila, komandi, područja, vozila) moguće je učitati različite popise.

- FactoryMethod (PopisFactoryMethod)

FactoryMethod uzorak pomoću definiranih podklasa radi interpretaciju objekata koje je stvorio CitacPopisaBuilder. Podklase definiraju o kojoj vrsti objekata se radi.

- Prototype (PripremateljPrototype)

Objekti ulica i spremnika se instanciraju koristeći postojeći atributi koji se kopiraju preko konstruktora te se dodaju novi atributi.

- **Proxy** (IspisivacProxy)

Koristi se za detaljniji ispis u programu i ispis u datoteku. Sadrži jednu metodu koja prilikom svakog poziva provjerava da li postoji datoteka za ispis. Također sadrži metodu koja provjerava vrijednost parametra ispis kako bi se ispisivale određene poruke u programu ili bi se ispisala samo statistika.

- **Composite** (PodrucjaComposite)

Odabran je za stvaranje strukture područja, podpodručja i ulica. Omogućava vezu dio-cjelina.

- **Iterator** (SpremnikIterator, UlicaIterator)

Odabran je za iteriranje kroz spremnike koje vozilo mora obići. Razlog je što pruža mogućnost pamćenja trenutne pozicije obrade, jednostavan dohvat trenutnog elementa i iteracije na sljedeći.

- **State** (StanjeVozilaState)

Vozila koja prikupljaju otpad mogu se nalaziti u različitim stanjima. Za promjenu stanja korišten je uzorak State gdje su za ConcreteState korištene klase Parkirano, Skupljanje, Kontrola, Kvar, Praznjenje, Pokvareno.

- **Decorator** (VoziloDecorator)

Za vozila je osmišljena nova funkcionalnost prijevoza osoba. Decorator omogućuje dodavanje atributa broj mjesta za vozilo te pamćenje liste osoba koje se trenutno voze u vozilu. Osobe tj. putnike je moguće ukrcati i iskrcati iz vozila.

### *Promjene u odnosu na 1. zadaću*

- dodani uzorci Proxy, Composite, Iterator, State i Decorator
- refaktorirana glavna metoda programa kao i sve ostale kako bi imale najviše 25 linija programskog koda
- dodano čitanje datoteke područja, vozila i dispečera te prilagođeno čitanje ostalih datoteka koje su imale promjene (ulice, vozila imaju dodan ID)
- područja se čitaju u Composite strukturu koja sadrže podpodručja i ulice (uspostavlja se EZO prostorni sustav), napravljen je tablični ispis otpada po ulicama i područjima
- dodijeljeni otpad korisnicima sada se odlaže u pripadajuće spremnike
- vrši se obrada vozila i odvoz otpada
- čitaju se i obrađuju komande dispečera, te na pripadajuće komande odrađuju se promjene stanja vozila, odvozi otpad ili ispisuje
- temeljem parametra preuzimanje određuje se redoslijed obrade ulica i njihovih spremnika za vozilo, što znači da vozila mogu imati zajednički ili zaseban raspored
- temeljem parametra ispis vrši se ispisivanje na zaslone i datoteku koje podrazumijeva isključivo statistiku ili u drugom slučaju detaljne informacije o obavljenim poslovima
- ispisuje se statistika nakon svih obrada

### *Vlastita funkcionalnost dodana projektu – prijevoz putnika vozilima*

Vozilo predstavlja prijevozno sredstvo koje ima određeni broj mjesta i za prijevoz osoba. Ta funkcionalnost je realizirana Decorator uzorkom dizajna koji u ovom slučaju dodaje atribut broja mjesta i popis putnika koji se nalaze u vozilu.

Svaki puta kada se vozilo napuni, pokupi jednog komunalnog redara (putnika) iz ulice te ga ukrca i poveze na mjesto za odvoz otpada. Na mjestu za odvoz otpada se svi putnici iskrcavaju. Vozilo koje prevozi smeće ima maksimalno 3 mjesta za prijevoz osoba.

## Dijagram rješenja

