

KORIŠTENI UZORCI DIZAJNA

Factory Method - ovaj je uzorak korišten kod čitanja ulaznih datoteka. Kreirano je sučelje "Datoteka" - ono definira metode koje klase namijenjene čitanju datoteka implementiraju. Klasa "DatotekaFactory" predstavlja tvornicu i temeljem stringa koji je proslijeđen se utvrđuje koja će se klasa instancirati. Također, navedni je uzorak korišten kod odabira algoritma za provjeru uređaja. Kreirano je sučelje "Algoritam" s metodom "dajMjesto", njega implementiraju sva tri kreirana algoritma. Tvornica je klasa "AlgoritamFactory" i putem metode "odaberiAlgoritam" se utvrđuje koja će algoritamska klasa biti instancirana.

Prototype - ovaj je uzorak korišten kod dodijeljivanja slučajnog broja uređajima unutar pojedinog mjesta. Također, namijenjen je zamjeni uređaja ukoliko se pokaže da je koji od njih neispravan. Klase "Senzor" i "Aktuator" sadrže "overridanu" metodu "clone" kojom se ovaj uzorak ostvaruje.

Singleton - uzorak je korišten kod klase "aplikacija" koja sadrži svu logiku zadatka, to je klasa koja se smije samo jednom instancirati. Također, singleton je korišten kod klase generatora koja se instancira samo jednom i postavlja joj se sjeme u početku programa. Uz navedeno, tu je i klasa "IzlaznaDatoteka" koja je singleton. U početku izvršavanja postavlja se maksimalan broj linija za spremnik te naziv izlazne datoteke.

Iterator - uzorak je zadan kao obavezan u zadatku. Kreiran je vanjski iterator kojim je moguće upravljati s klijentske strane s obzirom da je to dostatno za potrebe ove zadaće. Iterator ima nekoliko metoda koje omogućuju dohvaćanje prvog, trenutnog i sljedećeg elementa iz dane liste mjesta te metodu koja provjerava postojanje sljedećeg elementa.

Visitor - uzorak koji je zadan kao obavezan u zadatku. Klasa "IzlaznaDatoteka" implementira "Visitable" sučelje i time omogućava pristup "Visitor" klase. Visitor dodaje pojedine podatke u zapis spremnika.

PROMJENE U ODNOSU NA 1. ZADAĆU

Napravljene su veće promjene u odnosu na 1. zadaću. Adapter koji je korišten za čitanje datoteka u 1. zadaći promijenjen je u Factory Method. Uočeno je kako je drugi uzorak ispravniji pa je iz tog razloga promjena napravljena. Umjesto abecednog algoritma implementiran je algoritam uzlazno koji mjesta sortira prema rastućoj vrijednosti dodijeljenog slučajnog broja. Generator slučajnih brojeva je promijenjen prema nalogu iz zadatka. Logika zadatka je izdvojena u posebnu klasu koja prima argumente iz pokretanja programa (što je određeno zadaćom). Uređajima su dodane metode i svojstva potrebne za implementaciju rješenja. Kreiran je iterator koji omogućava iteriranje kroz niz mjesta. Dretva je promijenjena i osposobljena za provođenje ciklusa. Za provjeru ulaznih argumenata se više ne koristi regex već je napravljena switch petlja.

DODATNA FUNKCIONALNOST

Kreirano je sučelje "Visitable" kojeg implementira "IzlazDatoteka" i time omogućuje pristup visitoru. Funkcionalnost koju ovaj uzorak omogućuje jest zapis podataka o trenutnom sistemskom vremenu, apsolutnoj putanji projekta i broju linija koje spremnik zapisuje. Svaki puta kada se spremnik napuni i krene u zapis u datoteku poziva se metoda visit koja omogućava visitoru da odradi svoj dio.

