**Opis projekta**

U projektu nalaze se šest paketa i jedna Main klasa. Main klasa pod nazivom Program vrši sve pozive i radnje operacije koje su potrebne za izvršavanje programa. Budući da je zadatak u ovoj zadaći koristi MVC uzorak podijelio sam u tri paketa (Models, Views, Controllers) klase koje su vezane uz taj uzorak dizajna. Kontroleri i Pogledi imaju sufikse Controller i View dok s druge strane obične klase koje su služile kao kostur za punjenje podataka su nazivane po potrebi. Nadalje data paket sadrži dvije klase. Prva klasa je napravljena kako bi kreirali objekt koji sadrži sve podatke iz naredbene linije (CommandLineArguments), a druga klasa pod nazivom SystemDatabase nam služi za spremanje svih podataka te za pisanje jednostavnih upita. Zadnja dva paketa threads i helpers sadrže kako im i samo ime govori u prvom radnu dretvu odnosno WorkingThread koja u određenim ciklusima izvršava zadane operacije te u drugom se nalaze klase pomagaći koje sadrže razne metode koje nam olakšavaju rad u programu

**Builder**

Budući da se u komandnoj liniji nalazi puno informacija i podataka odlučio sam te podatke odvojiti u posebnu klasu te na toj klasi iskoristit Builder uzorak koji se koristi kako bi olakšao gradnju novih objekata kod kojih bi inače imali više parametara u konstruktoru. Primjer uzorka pronalazimo u klasi CommandLineArguments.

**Singleton**

I ovoga puta koristio sam Singleton uzorak dizajn jer mi je on omogućio najjednostavniji način za kreiranje objekta. Koristi se samo jedna klasa koja je odgovorna za kreiranje objekta te je taj objekt instanciran samo jednom. Primjere pronalazimo u klasam kao što su SystemDatabase, Generator.

**MVC**

Ili kako ime kaže Model-View-Controller se koristi kako bi se odvojila logika programa na tri razine. Model nam reprezentira JAVA POJO objekte koji nam nose podatke, View nam vizualno reprezentira naše podatke, dok Controller vrši radnje i prema View-u te i prema Controlleru te kontrolira unos podataka u model te osvježava pogled kad god se promjene podaci. Konkretno na ovom projektu odvojio sam logiku ispisa na komandnim linijama u poglede, za modele sam koristio POJO klase koje sam napunio podacima iz datoteka te ih spremio u „SystemDatabase“ koji je služio kao neki kontekst iz kojeg sam izvlačio podatke te sam u kontrolerima odvojio svu poslovnu logiku koja se tiče obavljanja operacija. Primjere možemo pronaći u klasama Modeli: Actuator, Devices, Order, Place, Sensor; Controlleri: ConsoleController, DeviceController; Viewovi: CosnoleView, DeviceView.

