Popis plánované implementace

F1 - Stavění domu

Na základě Configuration, vytvoří objekt Simulation pomocí Factory dům s jeho obyvateli. Při tvorbě domu pužije HouseFactory, která bude používat FloorBuilder, který poskytne API pro přidávání různých druhů místností (kuchyň, obývák…). Samotné místnosti pak budou vytvářeny pomocí RoomFactory.

F2, F5 – API zařízení

Různé zařízení budou implementovat interfacy (traity), které budou poskytovat API pro provádění akcí.

Každé zařízení bude mít taky svůj DeviceState, který bude určovat stav, ve kterém se zařízení nachází a jeho spotřebu.

F3, F4 – Spotřeba zařízení

Device bude implementovat rozhraní HasConsumption. Které bude poskytovat API pro zaznamenání spotřeby na konci časové jednotky simulace. Spotřeba bude přičtena podle stavu, ve kterém se zařízení nachází.

HasConsumption bude také definovat API pro získaní spotřeby.

F6, F10 - House a OutsideWorld

Jestliže bude zrovna Person v House, pak bude přihlášen na odběr Eventu a bude je generovat. Pokud zrovna bude sportovat v OutsideWorld, pak nebude Eventy ani odebírat, ani odbavovat.

Jestliže chce jít osoba do OutsideWorld, pak musí být dostupné nějaké sportovní vybavení.

F7 - Odbavováni eventů

Bude řešeno pomocí Observer patternu. Aby se každý obyvatel domu nemusel registrovat na každé zařízení zvlášť, tak bude Observable pouze dům jako takový. Lifecycle Eventu od vzniku po odbavení bude následující: Device nebo Inhabitant notifikují House a předají mu Event nějakého typu -> House si tento Event uloží -> na začátku další časové jednotky simulace se House podívá, jestli se na tento Event zaregistrovali nějací Observers (párování Observera a Eventu bude pravděpodobně řešeno Hashmapou) -> zavolá metodu Notify(vzniklý_event) na daném Observerovi -> Event se přesune do nějaké jiné kolekce pouze pro pozdější tvorbu EventReportu.

Vysvětleni interfaců spojeného s odbavováním eventů:

- EventPublisher a EventConsumer – Device a Inhabitant jsou EventPublisher a notifikují dům o vzniku Eventu.
- Observer a Observable – Device a Inhabitant jsou Observer a jsou notifikování domem o vzniku Eventu, jehož typ odebírají.¨

F8 – Reporty

HouseConfigurationReport
Každý element v Hierarchii domu bude akceptovat ConfigurationVisitor. Který posbírá nutné data a posune se na další uzly v hierarchii.

EventReport
Všechny Eventy budou uloženy v House.

ActivityAndUsageReport
Použití zařízení vygeneruje Event typu Alert. Tyto Alerty budou uloženy v House.

ConsumptionReport
Dům bude poskytovat DeviceIterator. Každý Device bude akceptovat ConsumptionVisitor, který posbírá data o spotřebě.

F9 – Manual, warranty a rozbite zarizeni

Pokud se zařízení rozbije, tak vygeneruje Event, který odbaví některý Adult, který je v domě. Použijem zde Lazy Initialization pattern. Pro každý typ spotřebiče bude stejný manuál, takže pokud se pro daný druh spotřebiče jednou načte, tak bude uložen do Object Pool.

Použité deisgn patterny

- State machine - DeviceState
- Iterator – InhabitantIterator, DeviceIterator
- Factory/Factory method – Inicialisace domu
- Builder – Inicialisace domu
- Dependency injection – Simulation dam HouseFactory v konstruktoru. Je to DI?
- Singleton – House, OutsideWorld, Configuration
- Visitor – Generování reportů
- Observer/Event listener
- Object Pool – Manual
- Lazy Initialization – Manual, Warranty