**Stručný (é) popis (y) projektu: Smarthome**

Autoři: Kozhemiakin Viktor, Belokon Nikita

Projekt smarthome je simulace domu, s použitím funkcionality “smarthome”. Simulace obsahuje chování osob v domě, použití spotřebiče a změnu jejich stavu podle náhodných eventů.

**Jak jsme realizovali tu ulohu?**

Krátká odpověď: Vytvoříme program v Javě, a pro testování budeme používat standartní konzole, pak budeme očekávat reporty z toho a kontrolovat jejich odpovědi.

Podrobnější popis:

* Popis modelu - Diagram tříd - je popis základních tříd, jejich chování a smyslu
* Konfigurace old - Podle úlohy máme 2 konfigurace. V tom soubory jsou popsané, jako příklad
* Hard conf – konfigurace pro >10 člověk
* Easy conf – konfigurace pro 6 člověk
* Dům.png je projekt našeho domu (pokoje)
* UC.png je Use Case diagram
* ShortDescription obsahuje scénáře a seznam pokojů a spotřebičů, které jsou v našem řešení
* Readme.md – popis splnění požádavků a pattern
* java-Doc - java doc
* example.txt – příklad standartního reportu

Pro fungování systému budeme používat “Design patterns” – Observer, Visitor a Singleton (jsou v Readme.md), které umožnují nám vhodnou a adekvátní práci s tým systémem.

Jak spustit aplikaci

V Main.java máme selectHouse, když změníte ten atribut, tak změníte jeden z 2 domů a pak program vypíše všechny reporty do kozole

Naše seebohodnocení

**Nikita Belokon:** Nejtěžší částí naší práce byl proces přemýšlení o struktuře budoucího projektu. Na základě Use Case diagramy a UML diagramy jsme sestavili přibližný vzhled práce a pustili se do procesu psaní struktury kódu. Zpočátku Projekt na papíře vypadal mnohem objemnější než na konci práce. Naše časové možnosti se neshodovaly s našimi plány na projekt, což způsobilo, že jsme zjednodušili některé části kódu. Projekt se ale nezhoršil. V důsledku práce jsme neměli konfliktní situace a velmi často se naše vize projektu shodovala. Snažili jsme se rozdělit práci rovnoměrně, ale protože píšeme kód různými způsoby, museli jsme si navzájem pomáhat se spojením částí projektu a snažit se spojit neslučitelné.

**Viktor Kozhemiakin:** Ta práce byla velmi zajímavá tým, že použili jsme patterny pro program, to je známka dobrého a solidního programu. Samozřejmě nebylo to použití ideální, ale mám začátek pro dalšího studia. Jsem jako dítě, které už umí číst, a může přečíst libovolnou pohádku samostatně, tak můžu napsat složitější a modernější program samostatně.

Týmová práce je velmi důležitá, ale měli jsme problém s tím, že to byl pro nás nový projekt a proto ne vždy chápali: “Co máme dělat a kde”. Ale doufám, že v budoucnosti ten problém už mít nebudeme.

Ještě jsem zopakoval nějaké věci ze Javy, například, když jsem měl otázku v 2. semestru: “Proč potřebujeme privátní konstruktor”, tak nechápal smysl toho, teď chápu (pro Singleton).