

Company Manager

Projekt OOP LS 2020/2021

Vypracoval: Peter Koszorús

AIS ID: 110826

Dátum: 15.5.2021

Obsah

Zámer projektu	3
Miera splnenia zámeru projektu	3
Diagram tried a ich vzťahov	4
Popis diagramu	4
Splnenie kritérií.....	6
Zoznam verzií.....	6
Vyhodnotenie	6

Zámer projektu

Company manager

Tento program bude uľahčovať zamestnávateľovi riadenie firmy. Zamestnávateľ bude pomocou tohto programu schopný jednoducho monitorovať prácu svojich zamestnancov a následne ju aj vyhodnocovať.

Program bude schopný zaznamenávať pracovnú dobu zamestnanca na základe ktorej bude potom zamestnancovi vyplatená mzda. Systém bude ponúkať aj prostredie kde zamestnávateľ bude vidieť, že ktorý zamestnanec je momentálne dostupný poprípadе ak nie je tak na ktorom projekte pracuje. Taktiež pri každom priečinku projektu na ktorom bude pracovať zamestnanec bude možné uploadovať dokumenty potrebné k realizácii projektu (server na ktorom by mali byť v reálnom behu programu uložené tieto údaje bude v mojom prípade s najväčšou pravdepodobnosťou simulovaný). Každý zamestnanec bude mať vytvorené konto pomocou ktorého sa bude prihlasovať. Admin teda správcom systému bude zamestnávateľ. Taktiež bude v priečinku s projektom kalendár v ktorom bude možné nastavovať dôležité termíny týkajúce sa projektu.

Miera splnenia zámeru projektu

Zámer projektu sa mi podarilo naplniť skoro úplne. Jedine čo sa mi nepodarilo naplniť je kalendár ktorý by ukazoval deadlines a taktiež to uplodovanie súborov na server.

Projekt som sa snažil vypracovať tak aby pri jeho implementácii bolo využitých čo najviac objektovo orientovaných postupov.

Diagram tried a ich vzťahov

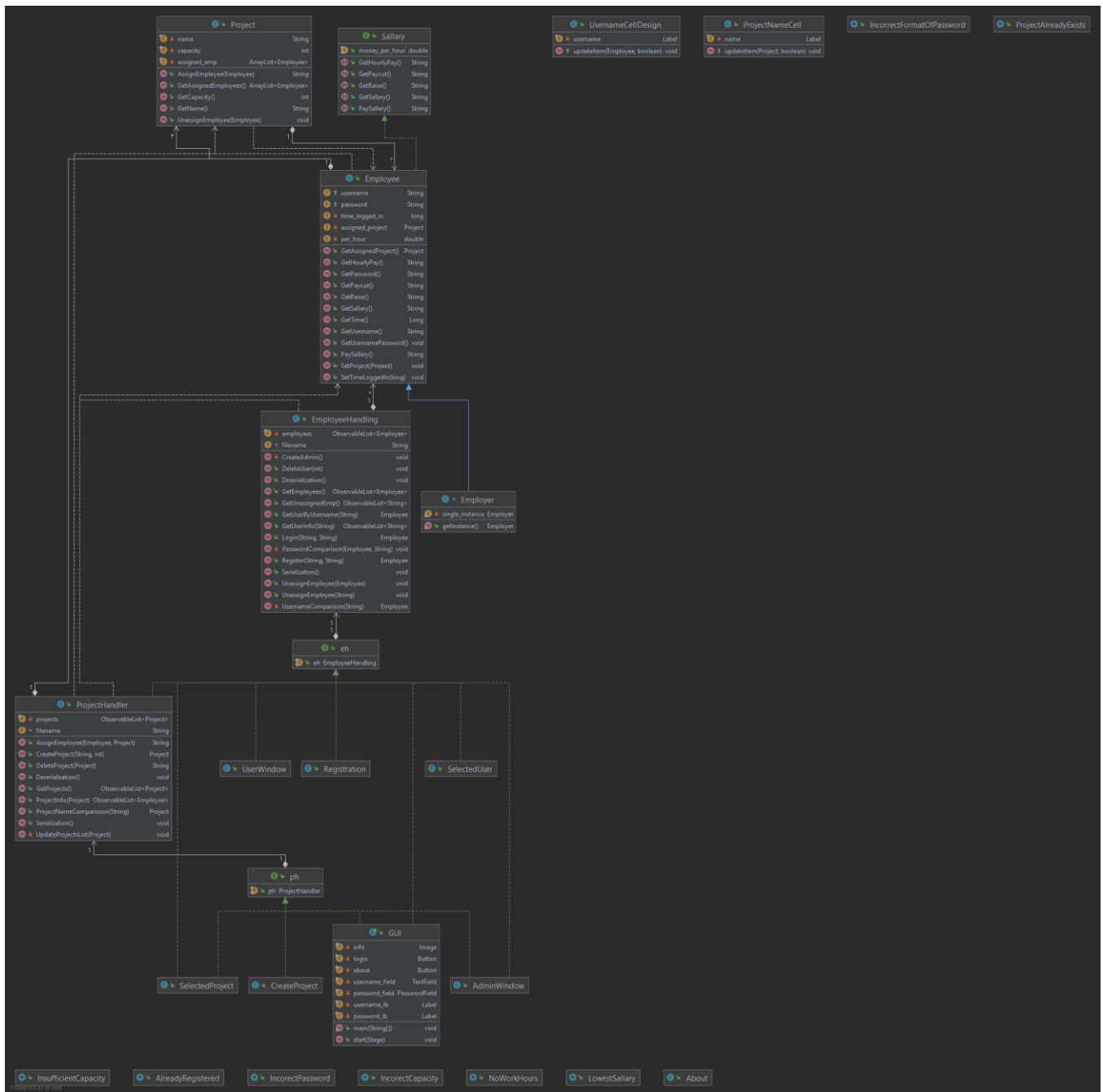


Diagram bude dostupný aj na GitHube v plnej veľkosti pre zjednodušenie čítania (UML_diagram.png).

Popis diagramu

V projekte som vytvoril triedu zamestnancov ktorá má child triedu zamestnávateľ (Singleton keďže zamestnávateľ je len jeden). Operácie nad týmito triedami vykonáva ďalšia trieda ktorú som nazval EmployeeHandler. Má na starosti vykonávanie operácii nad zamestnancami ako je napríklad ich

vymazávanie, prihlasovanie, registrovanie a ešte ďalšie pomocné metódy ktoré slúžia pre správne vykresľovanie v grafickom rozhraní. Tým, že tieto metódy mám v EmployeeHandleri som si zabezpečil hneď niekoľko vecí: dokážem pridávať hocikolko typov zamestnancov a program bude stále úplne v poriadku a taktiež som si týmto oddelil aplikačnú od logickej časti programu.

Rovnaký postup ako pri zamestnancoch som si zvolil aj pri projektoch metódy zodpovedné za operácie nad projektom sú v triede ProjectHandler. Každá trieda projektu obsahuje zoznam zamestnancov ktorý sú pridelený k danému projektu a taktiež každá inštancia triedy Employee obsahuje adresu inštancie projektu ku ktorému je pridelený.

Aby aplikačná časť dostala prístup k metódam na spravovanie zamestnancov alebo projektov je potrebné vytvoriť inštanciu triedy EmployeeHandler a ProjectHandler toto som si zabezpečil pomocou interfacu, toto mi zabezpečí to, že každá jedna trieda stagu ktorá využíva buď EmployeeHandler alebo ProjectHandler dedí z týchto interfacov tu istú inštanciu triedy ak by tomu tak nebolo a vytváral by si za každým novu inštanciu triedy program by bol nefunkčný. Moje riešenie sa dalo ešte nahradiť spôsobom, že by som z tried EmployeeHandler a ProjectHandler spravil triedy typu singleton.

Triedy ako SelectedUser, SelectedProject, AdminWindow, UserWindow, Registration, CreateProject a About sú triedy ktoré dedia triedu stage a obsahujú iba konštruktor kde je nadizajnovaný vzhľad a funkčnosť okna. Ak je vytvorená inštancia takejto triedy tak dôjde k otvoreniu nového okna (stagu).

Dalej projekt obsahuje aj triedy vlastných výnimiek ako aj dve triedy ktoré slúžia na dizajn jednotlivých riadkov v triede ListView.

Splnenie kritérií

Podľa môjho názoru program spĺňa všetky hlavné kritéria keďže využíva polymorfizmus, agregáciu, dedenie, zapuzdrenie a mnoho ďalších konceptov objektovo orientovaného programovania.

Môj program by mal taktiež spĺňať aj niektoré vedľajšie kritériá ako sú:

- 1, Využitie návrhového vzoru observer ktorý slúži na automatické obnovenie objektu ListView v grafickom rozhraní
- 2, Ošetrenie mimoriadnych stavov pomocou vlastných výnimiek (napr. pri registrácii, prihlasovaní atď..)
- 3, Oddelenie aplikačnej logiky od grafického rozhrania
- 4, Použitie RTTI pre prihlasovaní používateľa zisťujem o aký typ objektu sa jedná
- 5, Využitie lambda výrazu na prehľadanie celého zoznamu
- 6, Využitie sterilizácie (sterilizujem zoznam všetkých zamestnancov a zoznam všetkých projektov)

Zoznam verzií

Verzia 1.0.0 – Priebežná verzia kde som mal približne načrtnutú kostru programu a boli funkčne niektoré základne funkcie ako napríklad prihlasovanie

Verzia 1.1.0 – Bola funkčná časť programu ktorá mala na starosti spravovanie zamestnancov

Verzia 2.0.0 – Finálna verzia v tejto verzii je už aj implementácia projektov a ich prepájanie medzi zamestnancami

Vyhodnotenie

Tento projekt ma naučil veľa konceptov a princípov objektovo orientovaného programovania. Z výsledkom projektu som celkom spokojný a považujem predmet OOP za dôležitý a určite mám v pláne sa venovať objektovo orientovanému programovaniu aj v budúcnosti.

Keďže som projekt začal robiť dosť skoro tak ako bežal čas som zisťoval nové veci a niektoré veci vo svojom projekte by som pravdepodobne prerobil. Taktiež do projektu by sa dali ešte určite implementovať rôzne funkcie. Oblasť projektu ktorú považujem za slabšiu je grafické používateľské rozhranie.

Taktiež do budúca som si uvedomil, že je veľmi dôležité robiť častejšie a kvalitnejšie commity a tiež pracovať s dokumentáciou postupne ako developujem projekt. Predchádza to rôznym nedorozumeniam.