

Projekt OOP – Zdravotnícky systém – Dokumentácia

Obsah dokumentu

- Pôvodný zámer projektu (z 13.3.2020)
- Návod na spustenie a ovládanie programu
- Aktuálna štruktúra systému vo forme diagramu
- Plnenie hlavných kritérií
- Plnenie ďalších kritérií
- Zoznam hlavných verzií programu odovzdaných do GitHub

Pôvodný zámer projektu (z 13.3.2020)

Navrhovaný softvér disponuje možnosťou prihlásiť sa v niekoľkých roliach, použitím prihlasovacieho mena, hesla a zadaním SMS kľúča, čo zabezpečuje jeho dôveryhodnosť vzhľadom na spracovávanie citlivých údajov. Lekári, lekárnik a zdravotná poisťovňa dokážu pristupovať k niektorým informáciám o pacientovi na základe jeho rodného čísla.

Pacient v systéme vidí predpísané lieky, nárok na čerpanie doplatku na lieky a existenciu výmenných lístkov. Môže meniť svoje osobné údaje, ako adresu, či telefónne číslo.

Všeobecný lekár má možnosť predpisovať pacientovi lieky na **predpis**. Môže predpísať bežné lieky, alebo, na základe odporúčania, môže predpísať aj špecializované lieky. Taktiež vydáva pacientovi výmenný lístok pre špecializovaného lekára. Dokáže pacientovi meniť údaje o krvnej skupine a alergiách na lieky. Pridáva do evidencie nového pacienta.

Lekár – špecialista má rovnaké možnosti ako všeobecný lekár, no podľa špecializácie má oprávnenie vydávať predpisy a odporúčania na predpis špecializovaných liekov a zdravotných pomôcok. Môže vyšetriť len pacienta s platným výmenným lístkom.

Lekárnik vidí len dosiaľ nevydané recepty na lieky pacienta a nárok na čerpanie doplatku od zdravotnej poisťovne. Liek môže vydať.

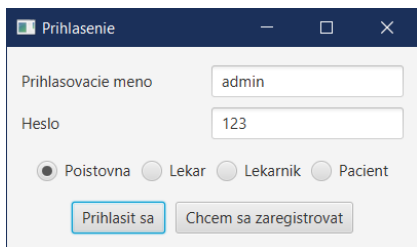
Zdravotná poisťovňa pacientovi eviduje informáciu o platbe zdravotného poistenia, na základe ktorej môže čerpať doplatok na lieky. Prepláca zdravotnú starostlivosť o pacientov lekárom a doplatky na lieky lekárnikom. Pridáva do evidencie nového lekára.

Návod na spustenie a ovládanie programu

Program sa spúšťa súborom PrihlasovacieOknoGUI. Po spustení programu sa objaví možnosť zaregistrovať sa ako nový pacient, alebo prihlásiť sa ako existujúci používateľ. To je možné uskutočniť v niekoľkých roliach.

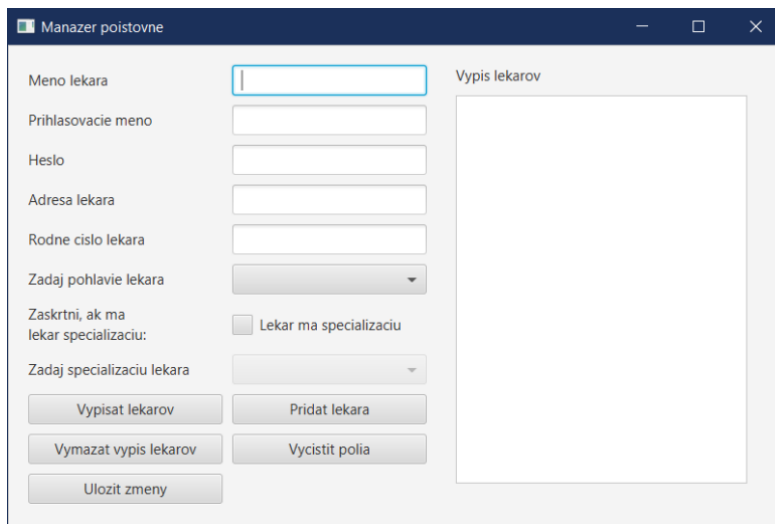
Prihlasovacie údaje pre rôznych prednastavených používateľov sú uvedené v tabuľke na konci tohto odseku.

Program podporuje iba jedného prihláseného používateľa súčasne. Prihlasovanie prebieha cez prihlasovacie okno, odhlasovanie je implementované iba tak, že sa okno daného používateľa zatvorí.



Obrázok 1 Prihlasovanie manažéra poisťovne

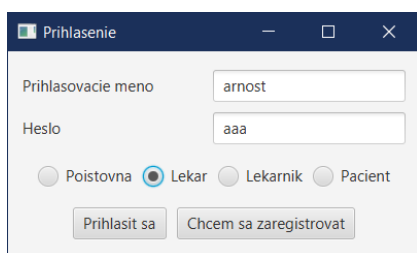
Manažér poisťovne (Obrázok 1, 2) je len jeden a prihlási sa vyplnením príslušných údajov, zaškrtnutie radiobuttonu „poistovna“ a kliknutím na tlačidlo „Prihlásiť sa“.



Obrázok 2 Prihlásený manažér poisťovne

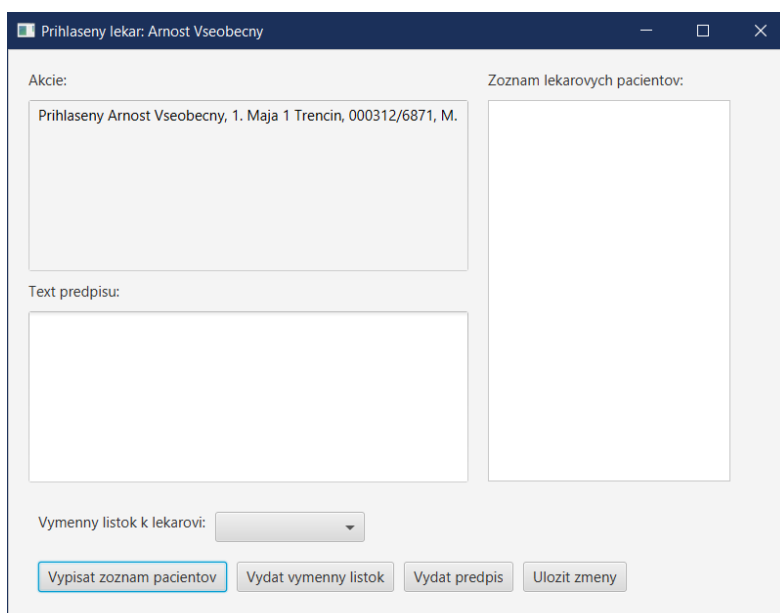
Po kliknutí na príslušné tlačidlo sa otvorí okno manažéra poisťovne. Ten môže vypísať všetkých lekárov do zoznamu napravo. Môže pridať nového lekára vyplnením polí a kliknutím na príslušné tlačidlo. Ak zvolí v checkboxe, že má lekár špecializáciu, vytvorí a pridá sa do evidencie nový špecializovaný lekár, inak sa pridá do evidencie nový všeobecný lekár.

Aby sa zmeny serializovali, je nutné kliknúť na tlačidlo „Uložiť zmeny“. Toto je nutné urobiť aj v ostatných oknách ostatných prihlásených používateľov a viac na to nebude v tomto návode upozorňované.



Obrázok 3 Prihlasovanie všeobecného lekára

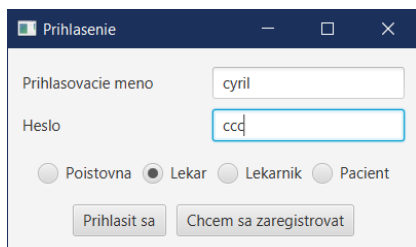
Analogicky k manažérovi poisťovne sa dá prihlásiť ako **lekár** (Obrázok 3, 4). V tomto prípade je uvedený príklad pre prihlásenie všeobecného lekára.



Obrázok 4 Prihlásený všeobecný lekár

Po prihlásení sa otvorí okno lekárskeho záznamu. Lekár môže vypísať zoznam svojich pacientov, ktorých má v evidencii. Po vypísaní evidovaných pacientov môže napísať text predpisu, vybrať pacienta zo zoznamu a tlačidlom „Vydať predpis“ mu môže vydať predpis so zadávaným textom. O tejto akcii bude napísané hlásenie do poľa „Akcie“.

Lekár môže vybrať špecializáciu lekára z choiceboxu a vybranému pacientovi zo zoznamu môže kliknutím na tlačidlo „Vydať výmenný listok“ vydať výmenný listok ku konkrétnemu špecialistovi.



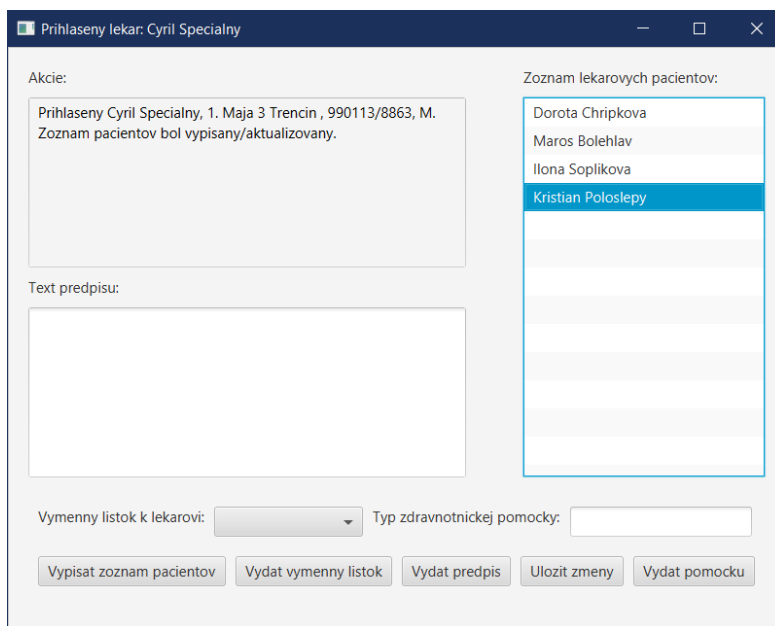
Obrázok 5 Prihlasovanie špecializovaného lekára

Ak sa prihlási **špecializovaný lekár** (Obrázok 5, 6), môže robiť všetko to, čo bežný lekár, a okrem toho môže vydať zdravotnícku pomôcku.

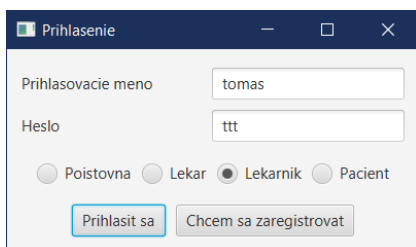
Vypíše si zoznam pacientov, vyberie pacienta zo zoznamu, napíše text predpisu, napríklad veľkosť

špecifikáciu zdravotníckej pomôcky, vyplní pole s typom zdravotníckej pomôcky, napríklad „okuliare“ a klikne na tlačidlo „Vydať pomôcku“. Do poľa „Akcie“ sa vypíše hlásenie.

Na okne prihláseného špecializovaného lekára je demonštrovaný výber pacienta z vopred vypísaného zoznamu pacientov.



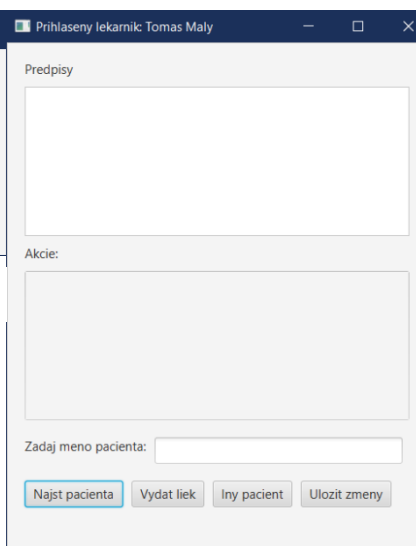
Obrázok 6 Prihlásený špecializovaný lekár s vypísaným zoznamom pacientov a vybraným pacientom zo zoznamu



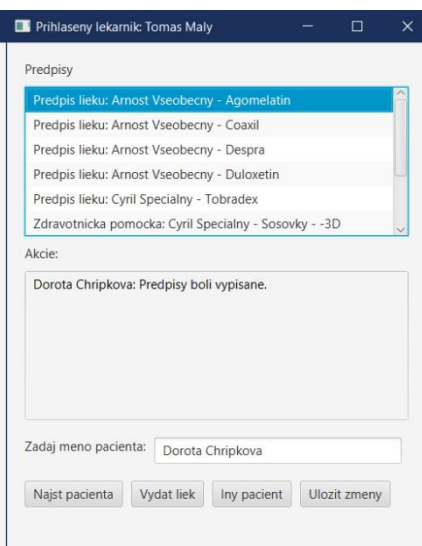
Obrázok 7 Prihlasovanie lekárničky

Po prihlásení lekárničky sa otvorí príslušné okno. Obrázok 7, 8.

Lekárnik zadá do poľa meno pacienta, ktorému chce vydať predpis a klikne na tlačidlo „Najst pacienta“. Ak



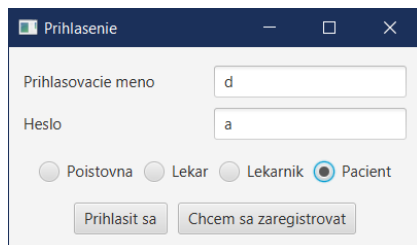
Obrázok 8 Prihlásený lekárnik



Obrázok 9 Prihlásený lekárnik s vyhladaným pacientom a vybraným receptom zo zoznamu

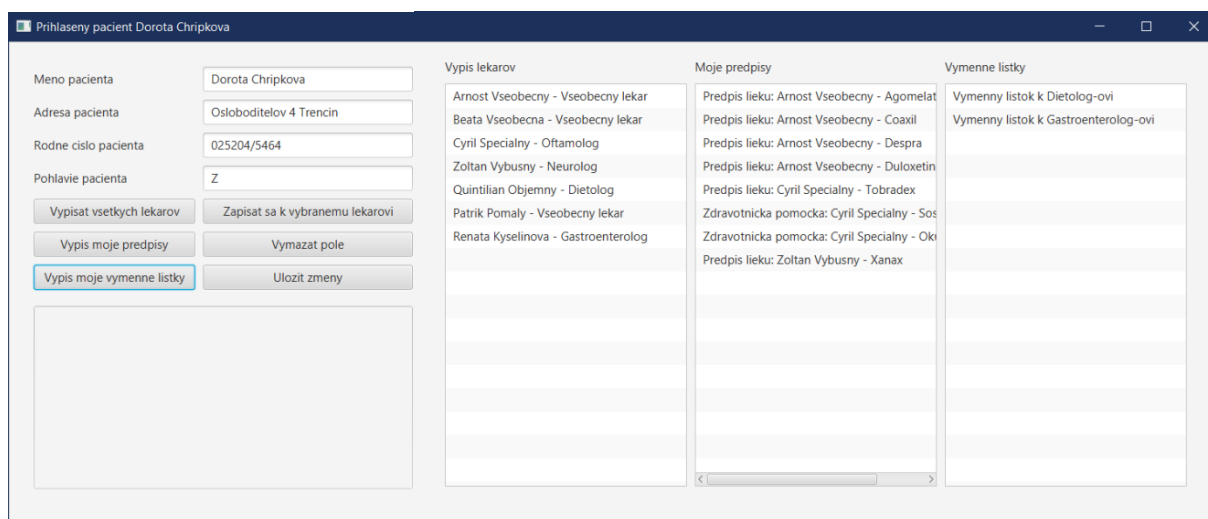
je pacient nájdený, vypíšu sa predpisy a do poľa „Akcie“ sa vypíše oznam. Ak sa pacient v evidencii nenachádza, vypíše sa do poľa „Akcie“ hlásenie.

Následne zo zoznamu pacientových predpisov vyberie jeden predpis, ako je znázornené na Obrázku 9, a klikne na „Vydat liek“. Po vydaní lieku sa vypíše hlásenie. Lekárnik môže vydať tomu istému pacientovi aj ďalší liek, prípadne vyčistí polia kliknutím na tlačidlo „Nový pacient“ a postupuje rovnako ako je popísané vyššie pre nového pacienta.



Obrázok 10 Prihlasovanie pacienta

Prihlásený **pacient** (Obrázok 10, 11), ako je zobrazené na obrázku nižšie, vidí svoje osobné údaje a môže vypísať všetkých lekárov poisťovne, všetky svoje dosiaľ lekárnikom nevydané predpisy a všetky svoje dosiaľ nepoužívané výmenné lístky. Do poľa pod tlačidlami sa píše hlásenie o aktuálnej vykonanej funkcionalite. Vypisovanie predpisov je jednorazová akcia.



Obrázok 11 Prihlásený pacient s vypísanými zoznamami lekárov, predpisov a výmenných lístkov

Ak pacient ešte nie je registrovaný u všeobecného lekára, zaregistruje sa k ľubovoľnému nich tak, že klikne na jeho meno v zozname v ktorom sú lekári vypísaní a klikne na tlačidlo „Zapísať sa k vybranému lekárovi“. Ak už všeobecného lekára má a pokúsi sa zapísať k inému všeobecnému lekárovi, hlásenie oznámi, že to nie je možné, pretože pacient môže navštevovať iba jedného všeobecného lekára.

Ak pacient má výmenný lístok k špecialistovi, vydaný obvodným lekárom, alebo iným špecialistom, môže sa zapísať k špecialistovi podľa špecializácie uvedenej na výmennom lístku. Zápis k vybranému špecializovanému lekárovi sa podarí iba ak má pacient príslušný výmenný lístok. Ak ho nemá, zápis sa nepodarí a vypíše sa hlásenie.

The screenshot shows a window titled "Prihlasenie". It contains two input fields: "Prihlasovacie meno" and "Heslo". Below these fields are four radio buttons for role selection: "Poistovna" (selected), "Lekar", "Lekarnik", and "Pacient". At the bottom, there are two buttons: "Prihlasiť sa" and "Chcem sa zaregistrovať".

Obrázok 12 Otváranie okna registrácie

Tlačidlom „Chcem sa zaregistrovať“ je možné otvoriť okno registrácie pre nového pacienta. Obrázok

Po vyplnení všetkých políček je možné sa zaregistrovať. Akonáhle je pacient registrovaný, môže sa prihlásiť v role pacienta, aby si vybral všeobecného lekára.

The screenshot shows a window titled "Registracia pacienta". It contains several input fields: "Meno pacienta", "Prihlasovacie meno", "Heslo", "Adresa pacienta", "Rodne cislo pacienta", and a dropdown menu for "Pohlavie pacienta". At the bottom, there are three buttons: "Zaregistrovať sa", "Vycistiť polia", and "Uložiť zmeny".

Obrázok 13 Okno registrácie nového pacienta

Rola	Meno	Prihlasovacie meno	Prihlasovacie heslo
Manažér poisťovne		admin	123
Pacient	Dorota Chripkova	d	a
Pacient	Emilia Chripkova	e	a
Pacient	Frederika Chripkova	f	a
Pacient	Gertruda Soplikova	g	a
Pacient	Hyacinta Soplikova	h	a
Pacient	Ilona Soplikova	i	a
Pacient	Jan Poloslepy	j	a
Pacient	Kristian Poloslepy	k	a
Pacient	Ladislav Poloslepy	l	a
Pacient	Maros Bolehlav	m	a
Pacient	Narcisa Bolehlavova	n	a
Pacient	Oktavian Bolehlav	o	a
Všeobecný lekár		arnost	aaa
Všeobecný lekár		beata	bbb
Všeobecný lekár		patrik	ppp
Špec. lekár		cyril	ccc
Špec. lekár		zoltan	zzz
Špec. lekár		quintilian	qqq
Špec. lekár		renata	rrr
Lekárnik		tomas	ttt
Lekárnik		ulrich	uuu

Tabuľka 1 Prihlasovacie údaje do systému pre prednastavených používateľov. Nie sú tam uvedené mená iných používateľov, ako pacientov z dôvodu ich nepotrebnosti. Mená pacientov sú potrebné pre lekárnika, aby ich mohol vyhľadať v evidencii.

Aktuálna štruktúra systému vo forme diagramu

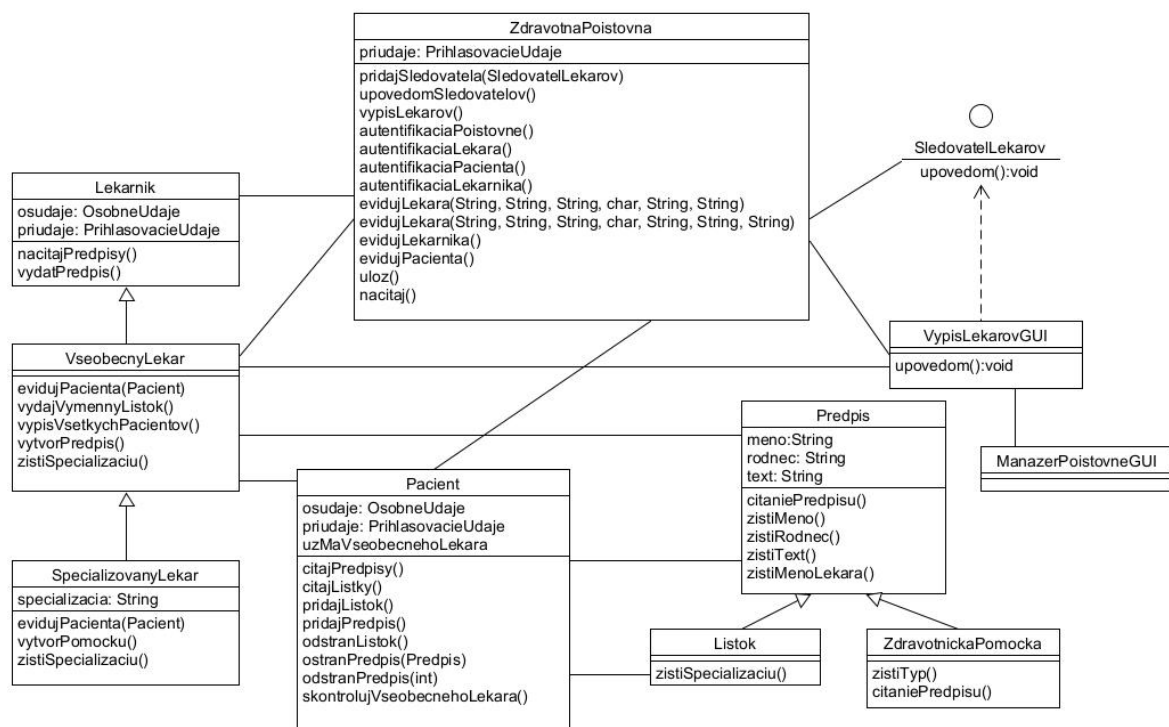
Keďže GUI neobsahuje metódy, nie je do diagramu zahrnuté celé, iba nutné časti pre demonštrovanie návrhového vzoru Observer.

Taktiež triedy PrihlasovacieUdaje a OsobneUdaje a ich rozhrania nie sú do diagramu zahrnuté, pretože sa jedná o jednoduchú agregáciu, ktorá len diagram zneprehľadňuje, miesto toho je ich vzťah popísaný: PrihlasovacieUdaje sú agregované triedami Lekarnik, Pacient, ZdravotnaPoistovna. Tieto triedy taktiež implementujú rozhranie ZistiPrihlasovacieUdaje. OsobneUdaje sú agregované triedami Lekarnik a Pacient. Tieto triedy taktiež implementujú rozhranie ZistiOsobneUdaje.

Výnimka NenajdenyUzivatelException a rozhranie Sprava, používané na generovanie správ lambda výrazmi tiež nie sú v diagrame pre nízku dôležitosť zahrnuté.

Triedy ZdravotnaPoistovna a Pacient obsahujú preťaženú metódu. Triedy ZdravotnickaPomocka a SpecializovanyLekar obsahujú prekované metódy.

Návrhový vzor Observer je implementovaný rozhraním SledovatelLekarov a triedami Poistovna a VypisLekarovGUI.



UML diagram tried programu

V diagrame sú uvedené len najdôležitejšie triedy aplikačnej logiky s ich najdôležitejšími metódami. Hierarchie dedičnosti aj vzťahy medzi triedami sú v diagrame označené. Funkcionalita programu je opísaná v časti Návod na spustenie a ovládanie programu.

Hlavnou triedou programu je ZdravotnaPoistovna. Sú tu uložené zoznamy lekárov, lekárnikov a pacientov. Všeobecný lekári a Špecializovaní lekári sú uložení spolu v jednom

zozname, čím dochádza k polymorfnému správaniu, napríklad pri prihlasovaní lekára. Prihlasovanie v okne prebieha práve pomocou autentifikačných metód v tejto triede. Taktiež sa pomocou metód týchto tried evidujú používatelia systému. Trieda obsahuje metódy na serializáciu a deserializáciu.

Trieda Pacient zodpovedá používateľovi pacient. Dokáže prečítať svoje predpisy, lístky a lekár dokáže pacientovi vydať Predpis a Lístok pomocou setterov.

Triedy Lekárnik, VseobecnyLekar, SpecializovanyLekar, Predpis, Listok a ZdravotnickaPomocka sú opísané v časti Plnenie hlavných kritérií, pri popise hierarchií dedenia.

Plnenie hlavných kritérií

Program je funkčný a vypracovaný pre verziu Java 8, tak ako bolo deklarované na webovej stránke predmetu. Funkčnosť programu bola niekoľkokrát prezentovaná aj na cvičeniach.

Pre spustenie programu je nevyhnuté mať ku kódu priložený aj súbor evidencia.out ktorý obsahuje prednastavené údaje o používateľoch.

Program je oproti **zámeru projektu** zredukovaný, z dôvodu nepotrebnosti implementovať všetku opísanú funkčnosť pre úspešné splnenie kritérií. SMS autentifikácia, vydávanie odporúčaní, či možnosť upravovať svoje osobné údaje boli z finálnej verzie projektu vypustené, rovnako ako doplatky pacientov a preplácanie poisťovňou. Projekt v súčasnej podobe naplňa hlavné kritériá hodnotenia a niekoľko vedľajších, preto nebolo potrebné implementovať všetko opísané v zámere projektu.

Program obsahuje **2 oddelené hierarchie dedenia**. Všeobecný lekár je podtriedou triedy Lekárnik, pretože dedí základnú štruktúru agregovaných údajov. Všeobecný lekár má oproti Lekárnikovi pridanú funkčnosť a dokáže evidovať pacienta, vydávať výmenné lístky a predpisovať lieky. Špecializovaný lekár je podtriedou Všeobecný lekár, dedí všetky jeho metódy, prekonáva metódu evidencie pacientov, pretože zatiaľ čo na navštívenie Všeobecného lekára nie je potreba mať výmenný lístok, pre navštívenie Špecializovaného lekára je nevyhnutné disponovať výmenným lístkom ku konkrétnemu špecialistovi. Špecializovaný lekár tiež pridáva metódu na vytvorenie predpisu pre zdravotnícku pomôcku, ktorú nemôže predpísať Všeobecný lekár. Všeobecní lekári a Špecializovaní lekári sú evidovaní v zozname Lekárov v Zdravotnej Poisťovni. Tu dochádza k polymorfizmu. Pri evidovaní nového pacienta pre nejakého Lekára zo zoznamu lekárov je volaná podľa typu lekára prekonaná, alebo pôvodná metóda evidencie. Špecializovaný lekár taktiež volá metódy svojho nadtypu, najmä v GUI pri vypisovaní pacientov. Opísaný príklad sa nachádza v súboroch *Lekarnik.java*, *VseobecnyLekar.java* a *SpecializovanyLekar.java*.

Ďalšia **hierarchia dedenia** je dedenie Predpisu. Výmenný Lístok a Zdravotnícka pomôcka sú 2 na sebe nezávislé podtriedy Predpisu. Výmenný lístok pridáva oproti predpisu metódu na zistenie typu lekára, pre ktorého je určený. Zdravotnícka pomôcka v sebe okrem zdedených atribútov Predpisu drží aj typ zdravotníckej pomôcky a taktiež má metódu na jej zistenie. Pacient eviduje Predpisy a Zdravotnícke pomôcky v jednom poli, z ktorého číta tieto

záznamy, teda dochádza k rozdielnemu spôsobu výpisu predpisu pre volanie prekonanej metódy, hoci sú držané spolu, v jednom poli, teda dochádza k polymorfnému správaniu. Opísaný príklad sa nachádza v súboroch *Predpis.java*, *Listok.java* a *ZdravotnickaPomocka.java*.

V projekte nachádza niekoľko **rozhraní**. Všetky typy osôb používajú rozhrania predpisujúce metódy na zápis a vrátenie osobných a prihlasovacích údajov. Údaje o osobách, a osoby v rámci poisťovne sú **zapúzdrené** a agregované. Príklad rozhrania je napríklad uvedený v súbore *ZistiOsobneUdaje.java*. Príklad zapúzdrenia a agregácie je uvedený v triede *Lekarnik.java*.

V projekte sú používané výstižné názvy premenných, z ktorých implicitne vyplýva, čo zastávajú. Každá trieda a každá jej metóda je okomentovaná JavaDoc **komentárom**.

Aplikačná logika oddelená od GUI – metódy aplikačnej logiky sa nachádzajú v triedach aplikačnej logiky. V event hadleroch v triedach GUI sa nachádza iba potrebný kód pre GUI a prácu s jeho elementami.

Kód je organizovaný do 3 **balíkov**, podľa príslušnosti ku GUI, osobám alebo poisťovni.

Dokumentácia – tento dokument.

JavaDoc – Priložený v priečinku JavaDoc.

Plnenie ďalších kritérií

Celkovo bolo splnených 9 z ďalších kritérií. Vzhľadom na ich počet poskytujem ku každému uplatnenému kritériu odkaz na súbor, v ktorom je možno nájsť implementáciu.

Použitie návrhových vzorov – Použitý návrhový vzor Observer pri výpise zoznamu Lekárov. Implementácia je analogická k implementácii vzoru Observer v Hre s Obrami a Rytiermi. V *ZdravotnaPoistovna.java* sa nachádza zoznam lekárov a zoznam sledovateľov lekárov. Sledovatelia sú odtiaľto aj aktualizovaní. Sledovateľom zodpovedá rozhranie *SledovatelLekarov.java*. Element GUI profitujúci z implementácie návrhového vzoru Observer je vo *VypisLekarovGUI.java*. V tomto ListView elemente sa udržiava aktuálny zoznam lekárov.

Ošetrovanie mimoriadnych stavov prostredníctvom vlastných výnimiek – Poisťovňa overuje prihlasovacie údaje, ak sa nenájde zhoda, metóda vyhodí výnimku, ktorá sa prejaví ako informačné okno. Výnimka sa nachádza v *NenajdenyUzivatelException.java*. Je vyhadzovaná napríklad metódou *autentifikaciaLekara* v *ZdravotnaPoistovna.java*. Zachytávaná je v *PrihlasovacieOknoGUI.java*, kde sa jej zachytenie prejaví vyhodnotením hlásenia s informáciou.

Poskytnutie grafického používateľského rozhrania oddelene od aplikačnej logiky a s aspoň časťou spracovateľov udalostí (handlers) vytvorenou manuálne – handlers sú vytvorené manuálne všetky, GUI je oddelené od AL. Ráta sa ako splnenie 2 kritérií. GUI zodpovedajú všetky súbory v balíku nazvanom GUI. Názvy tried AI korešpondujú s názvami

GUI a preto je možné ich ľahko nájsť. Napríklad *PrihlasenyLekarnikGUI.java* je GUI zodpovedajúce *Lekarnik.java* z balíka osoby.

Použitie serializácie – Objekt poisťovne a jej agregované objekty sú serializované. Serializácia je spúšťaná tlačidlom „Uložit zmeny“, ktoré sa nachádza v oknách všetkých prihlásených používateľov. Serializácia je implementovaná v *ZdravotnaPoistovna.java*.

Použitie lambda výrazov alebo referencií na metódy (method references) – vytvorené funkčné rozhranie *Sprava.java*, ktorým sa v *Lekarnik.java* generujú správy pomocou lambda výrazov.

Použitie implicitnej implementácie metód v rozhraniach (default method implementation) – rozhranie *ZistiOsobneUdaje.java* obsahuje default metódu na výpis všetkých údajov o prihlásenom používateľovi. Použitie je vidno v GUI, pri prihlásení Lekára, kedy sa do poľa „Akcie“ vypíšu informácie o lekárovi. Metóda je volaná z *LekarskeZaznamyGUI.java*.

Explicitné použitie viacnitévosti (multithreading) – spustenie vlastnej nite priamo alebo prostredníctvom API vyššej úrovne (trieda *Task* a pod.) – použité na vypisovanie Predpisov v triede *Pacient*. V súbore *Pacient.java* v metóde *citajPredpisy* sa nachádza vnorená trieda ktorá beží v samostatnej niti a vypisuje predpisy. Zabezpečuje plynulý beh programu, ak je predpisov na vypísanie príliš veľa. Ak by v programe nebola a predpisov na vypísanie by bolo príliš veľa, GUI by zamrzlo, kým by sa všetky nevypísali. Takto sa dá s GUI plynule pracovať ďalej aj pri náročnom výpise.

Použitie vnhniezených tried a rozhraní – vnhniezdenie triedy s novou nitou, ktorá je popísaná v predchádzajúcom kritériu. Vnhniezená v súbore *Pacient.java*.

Zoznam hlavných verzií programu odovzdaných do GitHub

Vzhľadom na to, že som commitoval po veľkých dávkach, commitov je pomerne málo, no rozsah vypracovaného projektu je adekvátny zadaniu aj Zámeru projektu. Takmer každá z verzií v GitHube je preto dôležitá a uvádzam ich tu všetky.

Text commitu je zároveň dostatočne výstižným krátkym popisom verzie programu. V niektorých prípadoch udávam aj ďalší popis, ten je uvedený v zátvorkách (). Commity nezaradené do tohto zoznamu sa týkajú len úpravy diagramu a dokumentácie.

commit 02fac0a37bee2d775c2822edf1c1e6189f2e7bdc

Author: Peter Smrecek <petersmrecek@gmail.com>

Date: Sun May 24 14:34:53 2020 +0200

Pridane komentare v kode a vypracovana dokumentacia.

commit 975e0ee5aeaa7ed3d4561ba4a9a5d4dd2064da48

Author: Peter Smrecek <petersmrecek@gmail.com>

Date: Sat May 23 18:02:45 2020 +0200

Opravný commit v dôsledku nezaradenia JavaDocu do commitovaných suborov.

commit 2104ac81818a8f690921bac38aabc5eee60bb4ed

Author: Peter Smrecek <petersmrecek@gmail.com>

Date: Sat May 23 18:01:07 2020 +0200

Pridany celý Javadoc, opravený bug pri multithreadingu, drobne zmeny v kóde v podobe zmien názvov tlačidiel. Pridaní užívatelia do serializovaného výstupu.

(Nešťastná formulácia, kód sa nemenil, do programu boli pridaní používatelia a boli uložené v serializovanom výstupe tak, aby sa s nimi dalo pracovať hneď po spustení programu a nebolo ich treba manuálne nahadzovať do programu.)

commit f4a3ed3fdd7eed5380a557b5e4fe1a1129f7af53

Author: Peter Smrecek <petersmrecek@gmail.com>

Date: Fri May 22 18:16:44 2020 +0200

Opravený bug v okne Lekarníka a prerobené vnorené triedy s vlastnými nitami.
Finalná verzia 1

(Úprava kódu, funkcionality sa od tohto momentu nemení.)

commit 0e6966ed4287a860b2d68bb20507c98a84650471

Author: Peter Smrecek <petersmrecek@gmail.com>

Date: Tue May 5 15:28:55 2020 +0200

Pridané vnorené triedy, thready, default metódy a lambda výrazov
(Pridanie vedľajších kritérií.)

commit 93ef2334e8ace57f590322a95275f67ed7639c8c

Author: Peter Smrecek <petersmrecek@gmail.com>

Date: Tue May 5 12:11:56 2020 +0200

Dokončenie hlavných kritérií + pridaná výnimka a jej ošetrenie

(Upravené dedenie a kontrola hlavných kritérií. Úprava GUI v dôsledku pridanej výnimky.)

commit e71df7258936016f8c2c969547cb8052986b0409

Author: Peter Smrecek <petersmrecek@gmail.com>

Date: Mon May 4 22:13:09 2020 +0200

Upravene GUI, pridany Lekarnik

(Pridaná trieda Lekárnik a jej GUI. Upravené prihlasovanie do programu.)

commit 81895f8b9d1f6fa66fe52dfaff9fa7e218b983d6

Author: Peter Smrecek <petersmrecek@gmail.com>

Date: Fri May 1 16:52:27 2020 +0200

Pridany predpis

(Pridaná funkcionálnosť vydávania predpisov, trieda Predpis a GUI bolo upravené, aby zodpovedalo pridanej funkcionálnosti.)

commit 7ff3b911032a6187317e2036f307585fdb4d2ee8

Author: Peter Smrecek <petersmrecek@gmail.com>

Date: Mon Apr 13 19:14:53 2020 +0200

Pridany diagram tried pre priebežné odovzdanie

(Kód programu sa výrazne nemenil.)

commit d5ac96e74da2369f818ef789bb6054ecf9957486

Author: Peter Smrecek <petersmrecek@gmail.com>

Date: Mon Apr 13 18:12:42 2020 +0200

Opravené chyby v readme.txt

(Kód programu sa výrazne nemenil.)

commit 47766171cc69bfedb6113cb37d7a4c8bbb46b253

Author: Peter Smrecek <petersmrecek@gmail.com>

Date: Mon Apr 13 15:55:07 2020 +0200

Uprava GUI a pridanie readme.txt pre potreby priebežného odovzdania.

commit 4c85d4c1dcfecdba8568228791308ab290eeb247

Author: Peter Smrecek <petersmrecek@gmail.com>

Date: Thu Apr 9 21:18:33 2020 +0200

Dokončenie gui

(GUI zodpovedá vtedy aktuálnej verzii programu a odvtedy boli robené už len úpravy.)

commit 6c71f423f1aad3d1c51e69249f193627b4b5c137

Author: Peter Smrecek <petersmrecek@gmail.com>

Date: Thu Apr 9 10:48:21 2020 +0200

Pridanie GUI

(Začiatok práce na GUI a pridanie hlavnej kostry GUI.)

commit 4bf954807e1073575d5b0a90d76043980a0f7e6a

Author: Peter Smrecek <petersmrecek@gmail.com>

Date: Mon Mar 30 22:28:51 2020 +0200

Pridanie príklady na dedenie, polymorfizmus, pretazenie, konstruktory, zapuzdrenie, prekonanie, serializáciu a agregáciu

commit d8fa09b3d0353911cde9598db9c016ea9e967d38

Author: Peter Smrecek <petersmrecek@gmail.com>

Date: Mon Mar 30 12:03:42 2020 +0200

Prvý commit - základné delenie do tried

(Vytvorenie základnej štruktúry programu v triedach a balíkoch.)