Palubný počítač - dokumentácia

1 ZADANIE

Dôveryhodný softvér

Všadeprítomnosť softvéru zviditeľňuje dôležitosť jeho dôveryhodnosti. Tá sa prejavuje cez ochranu, spoľahlivosť, dostupnosť, odolnosť a bezpečnosť. Ochrana predpokladá manažment rizík, spoľahlivosť a dostupnosť sa odzrkadľujú najmä v očakávanom správaní systému v priebehu času, odolnosť znamená udržanie funkčnosti systému za sťažených podmienok, kým bezpečnosť predstavuje cielené aktivity proti zámerným útokom.

Dôveryhodnosť možno vnímať ako mimofuknčný (nefunkcionálny) aspekt, ale môže byť realizovaná aj vyhradenými riadiacimi prvkami. Možné aplikácie zahŕňajú systémy na správu citlivých údajov, budov, dopravy, organizácií, spojení atď.

2 IMPLEMENTÁCIA

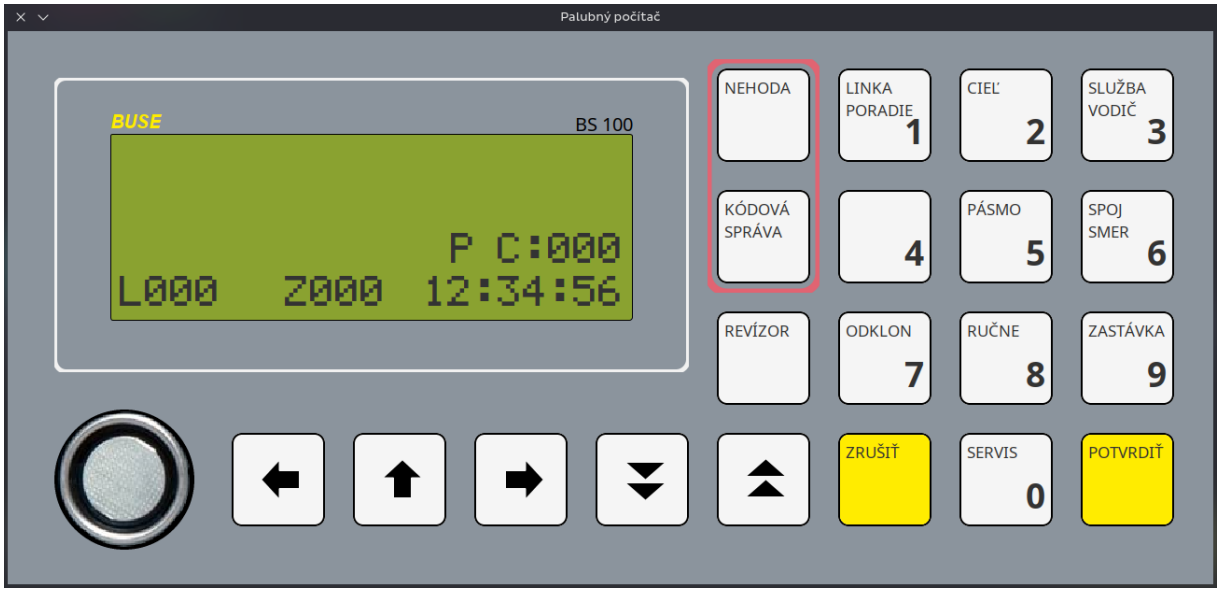
Ako spresnenie rámcového zadania som si zvolil vytvorenie palubného počítača Buse BS100. V jazyku Java som cez semester preto naprogramoval program, ktorý čiastočne simuluje svojou funkcionalitou aj vizuálnou formou časť informačného systému používaného vo vozidlách mestskej hromadnej dopravy. Pre reprezentáciu použiteľných liniek som si vybral električkové linky s cestným poriadkom aktuálnym ku dňu odovzdania (podľa plánu priloženom v prílohe).

Viaceré funkcie boli už priblížené v spresnení zadania v prvom odovzdávaní, táto časť len doplňuje dané údaje.

Z technických príčin sa mi nepodarilo implementovať všetky funkcie, ktoré som spomenul, no základnú funkcionalitu program zahŕňa. Pomocou tohto programu sa dajú pochopiť základné princípy ovládania tohto palubného počítača, pre ďalšie funkcie, ktoré nie sú obsiahnuté ďalej treba vyhľadať pôvodný prístroj.

Pre vytvorenie používateľského prostredia som použil java Swing toolkit, a všetky elementy som vyskladal pomocou tried, ktoré Swing poskytuje. Viaceré moje triedy dedia z tried poskytnutých, ide hlavne o JButton, JLabel, JFrame a JPanel.

3 PREHĽAD FUNKCIÍ PROGRAMU



3.1 PRÁCA S PROGRAMOM

Práca s programom je veľmi jednoduchá, ide hlavne o stláčanie tlačidiel. Pri ich stláčaní sa vykonávajú rôzne funkcie, ktorých výsledok je možné vidieť na zelenom display-i v ľavej časti aplikačného okna.

Po spustení je systém v móde, kde fungujú tlačidlá cez ich funkcie – čísla na tlačidlách môžeme zanedbať. Po spustení niektorej z týchto funkcií, je systém v móde náhľadu – vidíme aktuálne údaje, ktoré sú zadané v pamäti. Ak znova stlačíme niektoré z numerických tlačidiel, tak môžeme prepisovať dané údaje. Svoju voľbu potom potvrdíme, alebo zrušíme žltými tlačidlami.

Tento princíp funguje naprieč celým systémom. Po prihlásení na linku (vysvetlené v kapitole 3.3) sú na display-i zobrazené ďalšie údaje, a to po riadkoch v nasledovnom formáte:

1. Prázdny riadok, slúži iba na interakciu
2. Aktuálna zastávka, Z: kód zastávky
3. Cieľová zastávka, P (stav rádiostanice, v simulácii trvalo P - prihlásená), C: kód cieľu
4. Kód linky, aktuálna zóna, aktuálny čas

3.2 FUNKČNÉ TLAČIDLÁ

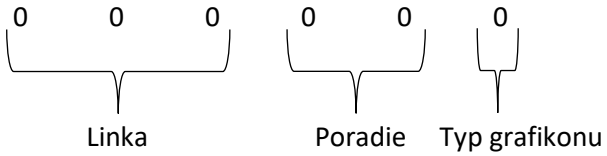
Implementácia pozostáva z tlačidiel s funkciami:

- 1 - Linka, Poradie
- 2 - Cieľ
- 3 - Služba, vodič
- 6 - Spoj, Smer
- 9 - Zastávka
- Zrušiť
- Potvrdiť
- Tlačidlá dvojitých šípok
- Numerické tlačidlá

3.2.1 Linka, Poradie
Tlačidlo slúži na priame zadanie trojmiestneho čísla linky, poradia na linke, a jej následného spustenia.

3.2.2 Cieľ
Tlačidlo slúži na priame zadanie cieľu, vo forme trojmiestneho čísla.

3.2.3 Služba, vodič
Tlačidlo slúži na zadanie služby – šesť miestneho čísla, zadania unikátneho päťmiestneho čísla vodiča, a ich následnej akceptácie. Číslo služby pozostáva z viacerých údajov, a to vo formáte:



Po zadaní týchto údajov a ich akceptácií sa spustí linka, ktorá bola zadaná.

Typ grafikonu je číslo vybraté podľa legendy:

- 1 - pondelok až piatok
- 2 - Sobota

- Nedeľa a sviatok

3.2.4 Spoj, smer
Tlačilo slúži na zmenu smeru aktívnej linky – číselné kódy 00 alebo 01.

3.2.5 Zastávka
Tlačidlo slúži na potvrdenie a odchod z aktuálnej zastávky. Pre jeho funkčnosť je potrebné byť zaregistrovaný na platnej linke a nachádzať sa na niektorej zo zastávok.

3.2.6 Zrušiť
Tlačidlo slúži na rušenie spustených funkcií tlačidiel.

3.2.7 Potvrdiť
Tlačidlo slúži na akceptáciu zadaných údajov.

3.2.8 Tlačidlá dvojitých šípok
Funkčné sú dve tlačidlá, obe sú schopné inkrementovať alebo dekrementovať vstupné čísla.

3.2.9 Numerické tlačidlá
Všetky tlačidlá na ktorých je vyobrazená číslica slúžia na zadávanie číslic na display. Ich interakcia je ale možná iba pri spustenej funkcií, ktorá podporuje zadávanie vstupných údajov.

3.3 POSTUP PRE PRÁCU SO SYSTÉMOM

Vytvorme scenár:

Chceme zadať električkovú linku číslo **4**, poradie *12*. Linka dnes jazdí v prázdninovom režime (typ grafikonu číslo 4).

- Naše šesť-miestne číslo bude vyzerať nasledovne: "**004***144*".
- Číslo vodiča má každý vodič svoje unikátne, pre naše testovanie to môže byť kľudne "12345".

Zadávanie týchto údajov sa realizuje nasledovne:

1. Zatlačím tlačidlo "3 - Služba, vodič".
2. Na displayi sa objaví aktuálny stav zadanej služby.
3. Môžem zadať číslo služby pomocou numerických kláves, a potvrdím tlačidlom "Potvrdiť".
4. Display následne vypíše číslo vodiča, ktoré je tiež potrebné zadať a potvrdiť.
5. Po zadaní údajov display vypíše údaje o linke, východziu a koncovú zastávku, a otázku, či je zadaná služba správna.
6. Po potvrdení sa spustí linka ktorá bola zadaná.

Zadané údaje sa po poslednom kroku zapísali do pamäte. Môžeme si všimnúť, že číslo linky ostáva vypásané na poslednom riadku displaya - za písmenom L (v našom prípade "L004"). Týmto krokom sa spustila linka číslo 4, a môžeme pomocou tlačidla “9 – zastávka” postupne prechádzať zastávkami.

Pri ostatných tlačidlách možno niektoré z aktuálnych údajov aj samostatne meniť, rovnakým alebo podobným postupom ako bol opísaný vyššie. Čo tlačidlá menia je vypísané na nich.

4 SPLNENÉ KRITÉRIA

Od informačného systému sa očakáva spoľahlivosť a hlavne funkčnosť komunikácie medzi ďalšími prvkami tarifno-technických zariadení. Je potrebné aby tento systém bezpečne fungoval za každých okolností a požiadavok svojej obsluhy, aby aj uľahčoval prácu pri mimoriadnych udalostiach taktiež aby zreteľne ukazoval údaje potrebné pre fungovanie spojov. V mojom programe som preto dbal zvýšenú pozornosť na jeho funkcionalitu, a ošetrovanie špeciálnych prípadov, ktoré by mohli oslabiť jeho funkčnosť a spoľahlivosť.

4.1 HLAVNÉ KRITÉRIA

1. **miera splnenia zadania** – moje vypracovanie sa dotýka viacerých objektovo orientovaných princípov, ktoré sú využité naprieč celým programom. Počas písania programu boli postupne pridávané nové funkcie a aplikačná logika, a tak sa vystaval výsledný program.
2. **uplatnenie objektovo-orientovaných mechanizmov v programe**
    a. **dedenie** – Väčšina tried v programe je obsiahnutých v aspoň jednom dediacom strome. Najviac je dedenie vidieť pri triedach BuseButton, BuseScript, TextLine, a všetkých ďalších triedach ktoré z nich dedia.
    b. **polymorfizmus** – polymorfizmus je tiež vidieť všade naprieč programom. Viditeľný je hlavne pri triedach liniek v balíku pp.data.lines.
    c. **zapuzdrenie** – všetky triedy v programe využívajú zapuzdrenie svojich premenných. Pre prácu naprieč triedami sú použité getter-y a setter-y.

   d. **agregácia** – tento OOP princíp je podstatne využitý v mnohých oblastiach. Ako príklad sa dá uviesť ArrayList pre uloženie buttonov alebo ukladanie rôznych scriptov v buttonoch, ale aj v ďalších častiach programu.
3. **organizácia kódu** – program je organizovaný do viacerých balíkov, podľa funkcie jednotlivých tried v programe. Je tu zvlášť balík pre hlavné triedy, na ktorých je program postavený, svoj balík majú aj triedy databázy, triedy implementovaných liniek a triedy vizuálnych prvkov. V týchto balíkoch sú triedy organizované do ďalších balíkov podľa svojej ďalšej špecifikácie. Ako posledný balík je tu aj balík obsahujúci všetky súbory, ktoré neobsahujú kód programu – balík resources s obrázkami a zvukovými súbormi.
4. **kvalita dokumentácie** – celý program je zdokumentovaný v sebe samotnom pomocou dokumentácie Javadoc, a taktiež v tomto dokumente. Program bol písaný tak, aby sa dalo čo najrýchlejšie pochopiť kúskom kódu, a to pomocou komentárov a formátovania do blokov.

4.2 ĎALŠIE KRITÉRIA

1. **použitie návrhových vzorov** – v celom projekte bol voľne implementovaný návrhový vzor MVC, ktorý bol mierne pozmenený pre funkčnosť a málo potrebné obnovovanie obrazovky. V podstate teda ide o pozmenenie v časti ovládania, kde view nie je naplno oddelený od controler-u pri riadení zobrazovania údajov na hlavnom display-i.
2. **ošetrenie mimoriadnych stavov prostredníctvom vlastných výnimiek** – vlastnú výnimku som naimplementoval pri načítavaní zvukového súboru click.wav v triede BuseButton. Výnimka je tu nazvaná ako NoAudioLoadedException, a vyhadzuje ju práve vtedy, ak sa nepodarí na začiatku programu načítať zvukový súbor.
3. **poskytnutie grafického používateľského rozhrania oddelene od aplikačnej logiky** – program je vytvorený s grafickým používateľským rozhraním, ktoré je oddelené od logiky. Aplikačná logika funguje oddelene, a to pomocou rôznych ActionListener-och na button-och, ktoré majú každá vlastné funkcie.
4. **explicitné použitie viacniťovosti** – pre toto kritérium som naimplementoval tzv. debug mode, ktorý sa dá aktivovať cez System.in. Ide tu o nekonečný while() cyklus, ktorý beží popri hlavnom programe.
5. **explicitné použitie RTTI** – tento princíp som použil pri update display-a v metóde setStop() v triede CurrentData, a to tak, že si zistím či je vybratá inštancia podtriedy JLabel taká, akú potrebujem.
6. **použitie lambda výrazov alebo referencií na metódy** – lambda výrazy mám v projekte implementované na viacerých miestach, najviac asi pri priraďovaní funkcií ku actionEvent v button-och. Tieto výrazy sa ale určite budú dať nájsť aj inde.
7. **použitie implicitnej implementácie metód v rozhraniach** – implicitnú metódu som využil pri rozhraní Line, ktorú implementuje trieda TramLine. Implicitné metódy tu sú getCode() a getDestination(), ktoré vracajú základné hodnoty.

5 DIAGRAM TRIED

UML diagram bol vytvorený pomocou programu PlantUML a nachádza sa v adresári projektu. Obsahuje všetky triedy a zobrazenie dedenia naprieč celým stromom tried.

6 HLAVNÉ VERZIE PROGRAMU

Počas vytvárania programu boli priebežné verzie odovzdávané do repozitára na github.com. Ako hlavnejšie verzie programu by sa dali zhrnúť do pár bodov, a to:

1. 9dbd3fe – verzia obsahujúca aplikačné okno spolu s grafickým rozhraním
2. f717cd – verzia s prvým skriptom pre tlačidlá
3. e1a287e – verzia programu použitá pri odovzdaní
4. ddd5091 – verzia obsahujúca prvotnú implementáciu liniek
5. be4aef2 – finálna verzia po opravách

7 ZHODNOTENIE

Finálna verzia bola spolu s touto dokumentáciou a JavaDoc dokumentáciou push-nutá na github určený pre účely tohto zadania. Stav tried a balíkov obsiahnutých v projekte zodpovedá stavu opísanému v dokumentácií, všetky kritéria tu opísané by sa mali dať s ľahkosťou nájsť v programe.

8 PRÍLOHA