

Řízení výtahu

KIV/TI – Semestrální práce

studenti: *Marek Kubů a Josef Kalivoda*

studijní čísla: A15B0072P, A15B0055P

Zadání

Vytvořte konečný automat (resp. síť konečných automatů) pro řízení nejrůznějších zařízení nutných k činnosti výtahu. Automat tedy bude zapínat a vypínat motory (motor výtahu, otvírání dveří apod.) a světla (výtah jede, indikace zvoleného podlaží apod.), reagovat bude jak na vstupy externí (přivolání výtahu, volba podlaží, zastavení/opětovné rozjetí výtahu apod.), tak na vstupy interní (indikátor polohy výtahu, indikátor otevření/zavření dveří, bezpečnostní čidlo na dveřích apod.). Výtah by měl být pro alespoň čtyřpodlažní budovu. Dále vytvořte program, který činnost výtahu vizualizuje.

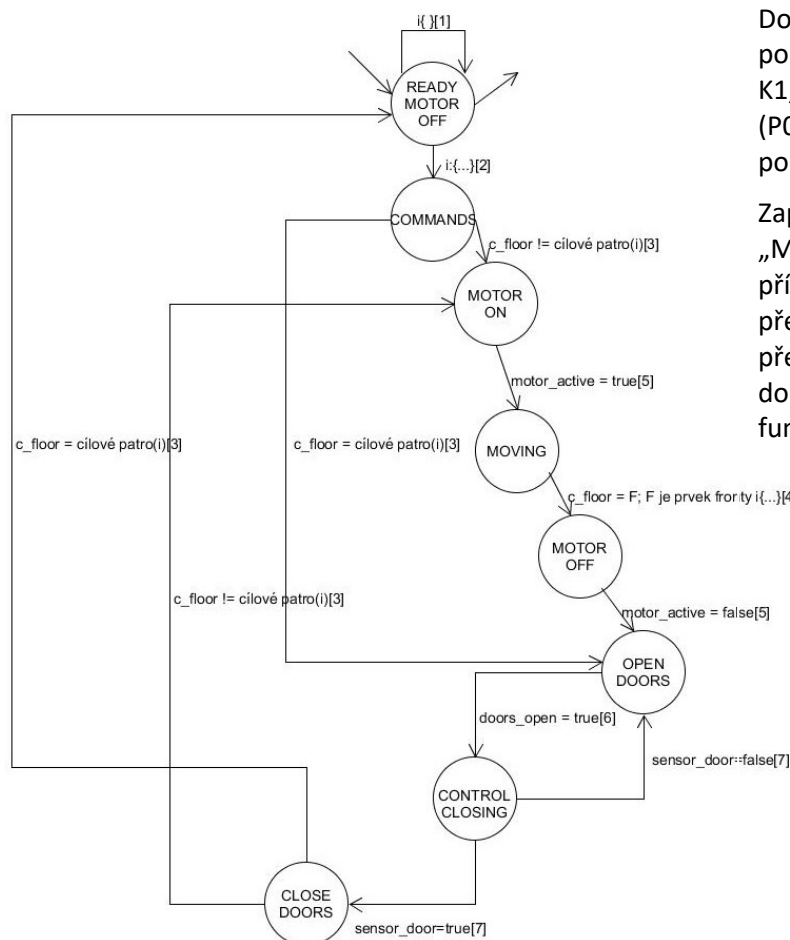
Teoretický rozbor

Zvolili jsme výtah pro čtyřpodlažní budovu. Program bude popisovat stavový automat, kterým projde výtah od zavolání po vykonání příkazu. Průběh lze ovlivnit vytvořením problému, který zachytí senzory na výtahu.

Pro realizaci projektu zvolíme programovací jazyk Java.

Automatový model

Přechodový graf



Do fronty i jsou vkládány hodnoty pater po stisku daného tlačítka čili z kabiny (K0, K1, K2, K3) nebo přivolávacím tlačítkem (P0, P1, P2, P3) tak, že se hodnota vloží pouze jako číslo patra.

Zapnutí a vypnutí motoru: Ve stavu „Motor on“ výtah motoru pouze vyšle příkaz na zapnutí, ale zapne se až na přechodu z tohoto stavu. Je vypnut až na přechodu mezi stavy „Motor off“ a „Open doors“. Používáme Mooreovu výstupní funkci.

- [1] fronta i je prázdná
- [2] fronta i obsahuje data
- [3] aktuální patro(c_floor) se porovnává s cílovým patrem, které si program vzal z fronty i , buď se shoduje (=) nebo ne(!=)
- [4] aktuální patro se shoduje s patrem, které je prvkem fronty i a zároveň hodnotu odebrá z fronty
- [5] motor výtahu je zapnut ($motor_active = true$) nebo je motor výtahu vypnut ($motor_active = false$)
- [6] dveře výtahu otevřeny
- [7] kontrola senzorů dveří, který zjistí jestli jsou dveře zavřeny nebo ne

Scénář

Kabina výtahu se nachází v přízemí (`c_floor = 0`). Ve třetím patře si cestující přivolali kabinu výtahu stisknutím přivolávacího tlačítka. Po stisknutí tohoto tlačítka se do zobecněné fronty (datová struktura, kterou využívá výtah) přidá hodnota odpovídající stisknutému tlačítku.

Hodnota je okamžitě vyjmuta z fronty a je nastavena jako cílové patro. Jelikož je výtah v jiném patře, než je patro cílové, zapne se motor a kabina se dá do pohybu, směrem do cílového patra. Další cestující zmáchl v druhém podlaží přivolávací tlačítko ještě než kabina projela jeho patrem. Hodnota se uložila do fronty. Jakmile byla kabina ve druhém patře, zjistila, že tam chce nastoupit cestující a vypnula motor. Poté se otevřely dveře a cestující nastoupil, zvolil v kabině patro č.0. Při zavírání dveří tam měl stále cestující ruku a dveře se opět otevřely, protože senzor dveří zahlásil chybu. Poté se dveře začaly opět zavírat a tentokrát to proběhlo bez problému. Kabina pokračuje dále do cílového patra (3. patro).

Jakmile se kabina dostavila do třetího patra, vykonala již zmiňovaný proces, kdy nechá nastoupit/vystoupit cestujícího.

V kabině jsou nyní dva cestující, první, který přivolal výtah do 3. patra a druhý cestující, který přistoupil v 2. patře. Ve frontě je zatím pouze hodnota 0, kterou zadal druhý cestující. Tato hodnota se v tomto momentě nastaví jako další cílové patro. První cestující chce do jiného patra a to konkrétně do 1. Při cestě do cílového patra (0) kabina zastaví v 1. Patře a nechá vystoupit pasažéra.

Dále nikdo nepřistoupil a kabina se přesunula do cílového patra, kde vystoupil poslední pasažér a výtah zůstává prázdný v posledním cílovém patře a čeká na další pokyny.

Popis implementace

Na vizualizaci výtahu jsme použili java knihovnu swing. Všechna vstupní data jsou ukládána pomocí tlačítek v uživatelském rozhraní, kde také lze nastavit stavy senzorů. Datovou strukturu pro ukládání a načítání hodnot pater využíváme zobecněné fronty, protože vstupujeme dovnitř a nejen na konec. Fronta je implementována třídou `ArrayDeque` z knihovny `java.util`.

Interní vstupy si automat generuje v závislosti na senzorech, které kontroluje před zavřením dveří. V naší implementaci není možné, aby se dveře otevřely za jízdy, protože při otvírání dveří musí být motor vypnut.

Struktura archivu

- `Sem_TI.jar` – jar soubor obsahující program i s java soubory
- `Dokumentace.pdf`
- `Doc` – složka s java dokumentací
- `Img` – složka s obrázky

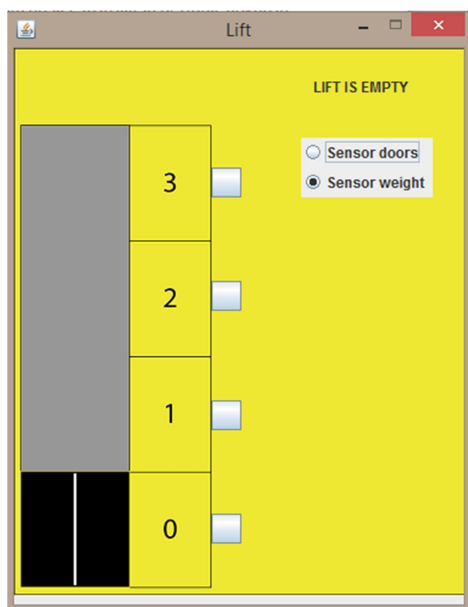
Spuštění programu

Program spustíte přes jar soubor nebo z příkazové řádky `java -jar sem_Tl.jar`. Jar soubor musíte mít vždy v jedné složce se složkou `img`. Při spuštění programu z příkazové řádky vypisuje program data také do konzole.

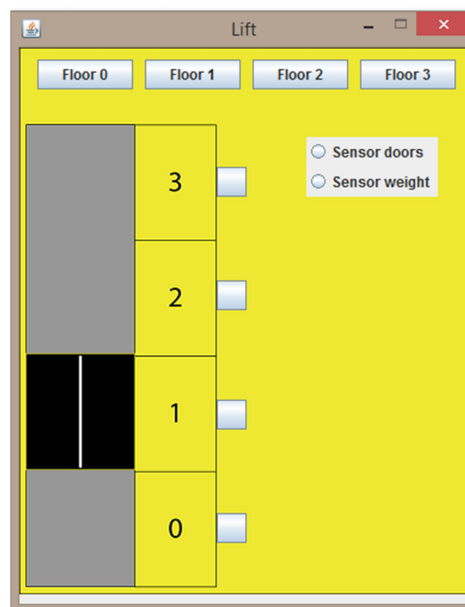
Uživatelská příručka

V aplikaci jsou použity následující komponenty.

- Radiobutton **Sensor doors** – pokud je tento radiobutton aktivní, zaškrtnutý, tak nám udává informaci o tom, že senzor mezi dveřmi hlásí chybu. Např. někdo tam má ruku apod.. Výtah se tedy nezavře, dokud uživatel neklikne znovu na radiobutton.
- Radiobutton **Sensor weight** – Pokud je aktivní, tak nám udává, že je výtah prázdný. Pokud je výtah prázdný, tak je zablokováno ovládání zevnitř výtahu. Změny se projeví až po otevření výtahu.
- U každého podlaží je jedno **přivolávací tlačítko** – po kliknutí přivoláte výtah do podlaží, u kterého jste to právě zmáčkli. Po zmáčknutí musíte počkat, až výtah přijede a otevřou se dveře, poté bude znovu možné použít dané tlačítko.
- Pokud jsou ve výtahu lidé, je umožněno ovládat zevnitř. K tomu slouží panel **Floor** tlačítek v horní části aplikace. Po stisknutí tlačítka se vykoná pokyn, nebo se přidá do fronty.



Obrázek 1 Prázdný výtah v patře 0



Obrázek 2 Výtah, ve kterém jsou osoby v patře 1

Závěr

Tento program lze volně rozšiřovat ve smyslu zvyšování počtu pater nebo přidávání více senzorů a indikátorů stavu výtahu. Při implementování jsme snažili dodržet námi navrženou síť konečných automatů pro zvolené téma. Příště musíme lépe návrh, který budeme posléze implementovat.