



Semestrální práce z Teoretické Informatiky (KIV/TI)

Vypracovali:

Vojtěch Danišík A16B0019P

danisik@students.zcu.cz

Jakub Černý A16B0015P

cernyj@students.zcu.cz

Michaela Janková A16B0044P

jankovam@students.zcu.cz

Obsah

1	Zadání	1
1.1	Obecné zadání	1
1.2	Konkrétní zadání	1
2	Analýza problému	1
3	Řešení problému	2
3.1	Návrh konečného automatu	2
3.2	Popis řešení	4
4	Popis implementace	4
4.1	Implementace	4
4.2	Popis stavů	5
4.2.1	Stav START	5
4.2.2	Stav SERVIS	5
4.2.3	Stav PENEZENKA	5
4.2.4	Stav NAPOJ_ZVOLEN	5
4.2.5	Stav ROZMENENI	6
4.2.6	Stav PLATIT_VIC	6
4.2.7	Stav VYDEJ_KELIMEK	6
4.2.8	Stav OHREV_VODY	6
4.2.9	Stav ODSYP_KAVY	6
4.2.10	Stav ODLITI_NAPOJE	6
4.2.11	Stav VRATIT_ZBYTEK	6
4.2.12	Stav VYDEJ_KAFE	7
4.2.13	Stav VRATIT_PENIZE	7
4.3	Popis akcí	7
4.3.1	Akce PRIDEJ_MINCI	7
4.3.2	Akce VYPIS_MINCE	7
4.3.3	Akce VYPIS_DOSTUPNOST	7
4.3.4	Akce KUP_KAFE	7
4.3.5	Akce VYPIS_POUZIVANI	7
4.3.6	Akce ODEBER_VODU, ODEBER_CUKR, ODEBER_KELIMEK	8
4.3.7	Akce PRIDEJ_VODU, PRIDEJ_CUKR, PRIDEJ_KELIMEK	8
4.3.8	Akce ODEBER_KAFE	8
4.3.9	Akce PRIDEJ_KAFE	8
4.3.10	Akce VRAT_CHYBOVY_STAV	8
4.3.11	Akce NEPLATNA_AKCE	8
4.3.12	Akce PRAZDNA_PENEZENKA	8
4.3.13	Akce VRAT_VSECHNY_PENIZE	8
4.3.14	Akce VRAT_ZBYTEK	9
4.3.15	Akce MAM_ROZMENENI	9
4.3.16	Akce CHCI_PLATIT_VIC	9
4.3.17	Akce VYDANI_KELIMKU	9

4.3.18	Akce OHREVANI_VODY	9
4.3.19	Akce ODSYPANI_KAVY	9
4.3.20	Akce ODLEVANI	10
4.3.21	Akce VYDANI_KAFE	10
5	Struktura archivu	10
5.1	složka app	10
5.2	složka doc	10
5.3	složka src	10
6	Uživatelská dokumentace	11
7	Závěr	14

1 Zadání

1.1 Obecné zadání

Vytvořte konečný automat (resp. síť konečných automatů) pro řízení zvoleného prodejního automatu (nápojového, na jízdenky apod.).

Automat bude mít tři typy vstupů: stisky tlačítek, indikaci hodnoty vhozené mince a signály z různých čidel (např. "došla káva", "došly kelímky" aj.). Svými výstupy bude automat řídit motory, elektromagnety apod., které zajistí vrácení mincí z mincovníku, přípravu kelímku, zapnutí/vypnutí ohřívání vody, puštění/zavření přívodu vody apod. Konkrétní vnitřní prvky automatu si realisticky navrhnete sami.

Vytvořte dále aplikaci, která vhodným způsobem automat vizualizuje a umožňuje simulovat mačkání tlačítek, vhazování mincí a sledování vrácení přeplatku; rovněž by měla umožnit simulovat výjimečné stavy, např. "neteče voda" apod.

Hlavním smyslem zadání je vyzkoušet si návrh implementace pomocí konečného automatu. Implementace proto musí striktně dodržovat princip konečného automatu; jiné implementace budou vráceny k přepracování.

Při implementaci začněte od konečného automatu, případné uživatelské rozhraní je doplněk, který doděláte nakonec. Rozhodně se nesnažte automat simulovat nástroji, které vám poskytuje knihovna pro tvorbu GUI, např. JavaFX apod.!

1.2 Konkrétní zadání

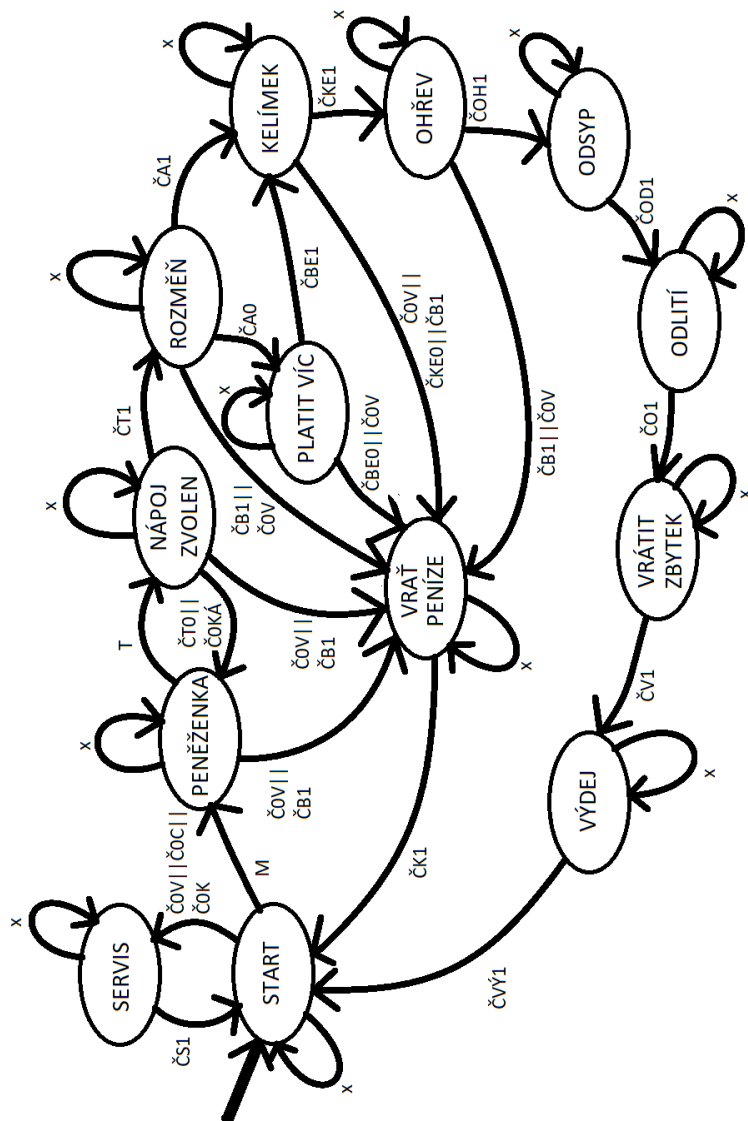
Pro naši práci jsme zvolili nápojový automat, jenž obsahuje 10 druhů teplých nápojů (kávy a čaj). Do automatu uživatel vloží požadovanou částku v mincích, zvolí příslušné tlačítko nápoje a nápoj se začne vyrábět - odsype se část zvolené směsi, ohřeje se dané množství vody a spustí se kelímek, do kterého se následně vše namíchá. Po dokončení tohoto procesu se uživateli vrátí zbytek peněz a čeká se, než si nápoj odebere.

2 Analýza problému

Problém může nastat v situacích, kdy dojde voda, cukr, nápojová směs nebo kelímky, nebo selže vydání kelímku, ohřev vody či odsyp směsi. Dále může dojít k tomu, že v automatu nebude dost mincí k vrácení.

3 Řešení problému

3.1 Návrh konečného automatu



Obrázek 1 - Návrh konečného automatu

Legenda:

ČS1 - Servisní tlačítko pro vrácení do počátečního stavu po doplnění komponent

Č0V - Čidlo značící nedostatek vody

Č0C - Čidlo značící nedostatek cukru

Č0K - Čidlo značící nedostatek kelímků

M - Vhozená mince

ČB1 - Tlačítko pro zrušení objednávky a vrácení peněz

T - Tlačítko pro volbu požadovaného nápoje

ČT0/ČT1 - Porovnání ceny nápoje s obsahem uživatelské peněženky

(0 - málo, 1 - dost)

Č0KÁ - Čidlo značící nedostatek požadované nápojové směsi

ČA0/ČA1 - Porovnání zbytku k vrácení s obsahem interní peněženky

(0 - nemá, 1 - má na vrácení)

ČK1 - Čidlo spouštící se při úspěšném vrácení drobných

ČBE0/ČBE1 - Volba uživatele, když automat nemá na vrácení (1 - zaplatit víc,

0 - vrátit peníze)

ČKE0/ČKE1 - Kontrola spouštění kelímku (0 - chyba, 1 - kelímek připraven)

ČOH1 - Čidlo ohřevu, které se spustí při dostatečné teplotě vody

ČOD1 - Čidlo odsypu, které se spustí při odsypání dostatečného množství směsi

ČO1 - Čidlo, které se spustí při úspěšném odlití do kelímku

ČV1 - Čidlo spouštící se při úspěšném vrácení drobných

ČVÝ1 - Čidlo odebrání kelímku uživatelem

x - zastupuje všechny vstupní signály, které ponechávají automat v aktuálním stavu

3.2 Popis řešení

Dostupnost vody, cukru a kelímků kontrolujeme na začátku chodu automatu, kdy se při chybějících komponentách přepne do chybového stavu, odkud se dostane pouze stiskem servisního tlačítka po doplnění pověřeným pracovníkem. Dostupnost nápojové směsi se řeší až při její volbě. Pokud v průběhu výroby jakákoliv z potřebných komponent dojde, automat na to uživatele upozorní, vrátí vhozené mince a přejde do počátečního stavu. Toto neplatí po skončení ohřevu vody, kdy již všechny potřebné komponenty máme, a jejich dostupnost pro další nápoj tedy stačí zkontrolovat až před počátkem jeho výroby.

V případě, že v automatu již není dost mincí k vrácení přebývajících částky uživateli, bude mu nabídnuta možnost si koupit požadovaný nápoj za vyšší cenu. Pokud bude uživatel souhlasit, bude mu s výdejem nápoje vrácena maximální možná částka. Pokud souhlasit nebude, vrátí se uživateli celý vklad a automat se přesune do počátečního stavu.

4 Popis implementace

4.1 Implementace

Semestrální práci jsme vytvářeli v programovacím jazyku Java. Pro ukládání množin stavů a akcí jsme si vytvořili výčetové typy `STAV` a `ACTION` a při práci pouze upravujeme jejich příslušné instance `STAV stav` a `ACTION action` v hlavní třídě `Automat`.

Informace o nápojích uchováváme ve třídě `Kafe` - my zvolili `int id` (pro indexování v mapě nápojů `Map<Integer, Kafe>` výrobky hlavní třídy `Automat`), `int cena`, `String nazev`, `boolean dostupne` a `int hmotnost`.

Pro práci s mincemi využíváme výčetový typ `MINCE`, který obsahuje hodnoty mincí - `JEDNA`, `DVA`, `PET`, `DESET`, `DVACET`, `PADESAT` - u nichž si uchovává i jejich číselnou hodnotu a počet v interní peněženice automatu.

Pro uchovávání hodnot v peněženice použijeme celočíselné proměnné `penezenka`, která se bude průběžně měnit například při počítání hodnoty k vrácení, a `penezenkaBezOdecitani`, která uchovává celou částku, kterou uživatel vložil, pro její případné vrácení při zrušení objednávky, nebo chybě při výrobě.

Dále zde máme třídu `ScannerThread`, která nám poskytne možnost program pozastavit na dobu nutnou například pro spadnutí kelímku po jeho uvolnění.

Hodnoty z čidel uchováváme pomocí proměnných typu `boolean` - `jeVoda`, `jeCukr`, `jeKelimek`, `jeKafe`, `zmacknuteTlacitko`, `mamNaRozmenit`, `beruRozmenit` (volba uživatele koupit si nápoj za vyšší cenu, nemá-li automat na rozměnění), `kelimekNaMiste`, `ohrevVody`, `odsypKavy`, `odliti`, `vratitZbytek`, `vydejKafe`.

Celý program se inicializuje - do mapy `vyrobky` se vloží nápoje a jejich ceny, `stav` se nastaví na `START`, `action` na `DEFAULT`, čidla se nastaví na hodnoty po doplnění a spustí se hlavní smyčka, která příkazem `switch` zareaguje na stav z proměnné `stav` - vypíše příslušné informace pro uživatele/personál, vyzve uživatele ke zvolení požadované akce zadáním příslušného řetězce, ze kterého se poté pomocí metody `zjistiAkci()` zjistí požadovaná akce a nastaví se do proměnné `action`. Poté metoda `provedAkci()` požadovanou (aktuálně nastavenou) akci provede.

4.2 Popis stavů

4.2.1 Stav START

Ve stavu `START` se nejdříve zkontrolují hodnoty proměnných `jeVoda`, `jeKelimek` a `jeCukr`. Pokud je některá z nich `false`, přepne se automat do chybového stavu `SERVIS`. Pokud proběhne kontrola komponent úspěšně, může se uživatel pomocí akce `PRIDEJ_MINCI` přesunout do stavu `PENEZENKA`.

4.2.2 Stav SERVIS

Ve stavu `SERVIS` se vypíší informace o chybách, zjistí se akce z podmnožiny - `PRIDEJ_VODU`, `PRIDEJ_CUKR`, `PRIDEJ_KELIMEK` a `PRIDEJ_KAFE` pro doplnění komponent, `ODEBER_VODU`, `ODEBER_CUKR`, `ODEBER_KAFE` a `ODEBER_KELIMEK` pro odebrání komponent a `VRAT_CHYBOVY_STAV`, která vrátí automat z chybového stavu zpět do počátečního, jsou-li všechny potřebné komponenty doplněny.

4.2.3 Stav PENEZENKA

Stav `PENEZENKA` načítá vhozené mince a počítá jejich celkovou hodnotu. Poté se pomocí volby příslušného nápoje přesune do stavu `NAPOJ_ZVOLEN`.

4.2.4 Stav NAPOJ_ZVOLEN

Ve stavu `NAPOJ_ZVOLEN` se zvolí a provede akce `KUP_KAFE`, kterou se při chybě (chybí-li nějaká komponenta potřebná pro výrobu nápoje) automat přesune do chybového stavu `SERVIS`, při nedostačující vložené částce do stavu `PENEZENKA`, při stisknutí tlačítka pro vrácení peněz do stavu `VRATIT_PENIZE` a v případě dostačující částky do stavu `ROZMENENI`.

4.2.5 Stav ROZMENENI

Ve stavu ROZMENENI se pomocí akce MAM_ROZMENIT postupným dělením a odečítáním z dostupného počtu jednotlivých mincí zjistí, zda jich máme dostatek k vrácení přebytku. Pokud ano, přesune se automat do stavu VYDEJ_KELIMEK, v opačném případě se přesune do stavu PLATIT_VIC.

4.2.6 Stav PLATIT_VIC

Ve stavu PLATIT_VIC je pomocí akce CHCI_PLATIT_VIC uživateli nabídnuta možnost si nápoj koupit na vyšší cenu (automat vrátí maximální možnou částku ze zbytku). Pokud uživatel souhlasí, přesune se automat do stavu VYDEJ_KELIMEK, v opačném případě se přesune do VRATIT_PENIZE a uživateli je vrácen celý vklad.

4.2.7 Stav VYDEJ_KELIMEK

Ve stavu VYDEJ_KELIMEK se využitím akce VYDANI_KELIMKU zkontroluje dostupnost potřebných komponent a, jsou-li všechny dostupné, spustí se vydávání kelímku. Po krátké pauze (aby měl kelímek čas propadnout), se ověří, že propadl, a automat se přesune do stavu OHREV_VODY.

4.2.8 Stav OHREV_VODY

Ve stavu OHREV_VODY se spustí akce OHREVANI_VODY, při které má uživatel poslední možnost objednávku zrušit a nechat si vrátit peníze. Pokud je voda dostupná a dosáhne požadované teploty, pokračuje se ve výrobě do stavu ODSYP_KAVY.

4.2.9 Stav ODSYP_KAVY

Ve stavu ODSYP_KAVY se spustí akce ODSYPANI_KAVY a po úspěšném odsypání se pokračuje do ODLITI_NAPOJE.

4.2.10 Stav ODLITI_NAPOJE

Ve stavu ODLITI_NAPOJE se spustí akce ODLEVANI a po úspěšném odlití se pokračuje do VRATIT_ZBYTEK.

4.2.11 Stav VRATIT_ZBYTEK

Ve stavu VRATIT_ZBYTEK se pomocí akce VRAT_ZBYTEK uživateli vrátí maximální částka ze zbytku vložených peněz.

4.2.12 Stav VYDEJ_KAFE

Ve stavu VYDEJ_KAFE je pouze akcí VYDANI_KAFE uživatel vyzván k odebrání nápoje. Po zaznamenání jeho odebrání se automat přesouvá zpět do stavu START.

4.2.13 Stav VRATIT_PENIZE

Ve stavu VRATIT_PENIZE se pomocí akce VRAT_VSECHNY_PENIZE vrátí uživateli celá vložená částka a automat se přesune zpět na začátek do stavu START.

4.3 Popis akcí

4.3.1 Akce PRIDEJ_MINCI

Akce PRIDEJ_MINCI zkontroluje, zda je hodnota mince platná (1, 2, 5, 10, 20, nebo 50) a přidá ji do konta uživatele.

4.3.2 Akce VYPIS_MINCE

Akce VYPIS_MINCE vypíše počty jednotlivých mincí v interní peněženke automatu.

4.3.3 Akce VYPIS_DOSTUPNOST

Akce VYPIS_DOSTUPNOST vypíše seznam všech nápojů, jejich aktuální dostupnost a hmotnost odpovídajících směsí.

4.3.4 Akce KUP_KAFE

Akce KUP_KAFE zkontroluje, zda je ID nápoje validní (existuje nápoj s tímto ID), zda je nápoj dostupný a zda uživatel vhodil dost peněz pro jeho koupi. Pokud nedojde k chybě v předchozích kontrolách, pokračuje se k simulaci výroby nápoje do stavu ROZMENENI.

4.3.5 Akce VYPIS_POUZIVANI

Akce VYPIS_POUZIVANI vypisuje info, jaké vstupy má uživatel zadávat pro zvolení požadovaných akcí.

4.3.6 Akce ODEBER_VODU, ODEBER_CUKR, ODEBER_KELIMEK

Akce ODEBER_VODU, ODEBER_CUKR a ODEBER_KELIMEK nastaví odpovídající proměnné `jeVoda`, `jeCukr`, nebo `jeKelimek` na `false`.

4.3.7 Akce PRIDEJ_VODU, PRIDEJ_CUKR, PRIDEJ_KELIMEK

Akce PRIDEJ_VODU, PRIDEJ_CUKR a PRIDEJ_KELIMEK zajistí nastavení příslušných proměnných `jeVoda`, `jeCukr`, nebo `jeKelimek` na `true`.

4.3.8 Akce ODEBER_KAFE

Akce ODEBER_KAFE nastaví dostupnost `dostupne` nápoje zvoleného pomocí ID na `false` a `hmotnost` na 0.

4.3.9 Akce PRIDEJ_KAFE

Akce PRIDEJ_KAFE nastaví dostupnost `dostupne` nápoje zvoleného pomocí ID na `true` a `hmotnost` na 200.

4.3.10 Akce VRAT_CHYBOVY_STAV

Akce VRAT_CHYBOVY_STAV zkontroluje dostupnost vody, cukru a kelímků. Pokud je vše dostupné, přesune automat do stavu `START`, v opačném případě vyzve pracovníka k doplnění chybějících komponent.

4.3.11 Akce NEPLATNA_AKCE

Akce NEPLATNA_AKCE pouze vypíše informaci, že akce nebyla platná, a ponechá automat v aktuálním stavu.

4.3.12 Akce PRAZDNA_PENEZENKA

Akce PRAZDNA_PENEZENKA vypíše uživateli informaci, že doposud nevložil žádnou minci, pokusí-li se si nechat vrátit vklad.

4.3.13 Akce VRAT_VSECHNY_PENIZE

Akce VRAT_VSECHNY_PENIZE vrátí uživateli celý vklad a přesune automat zpět do stavu `START`.

4.3.14 Akce VRAT_ZBYTEK

Akce VRAT_ZBYTEK zajistí vrácení maximální možné částky z vkladu uživatele po odečtení ceny nápoje.

4.3.15 Akce MAM_ROZMENENI

Akce MAM_ROZMENENI pomocí postupného dělení a odečítání maximálního dostupného počtu požadovaných mincí (např. pro padesátikoruny se celkový zůstatek v peněžence celočíselně vydělí hodnotou 50, výsledek se porovná s počtem padesátikorun v automatu a, je-li požadován větší počet padesátikorun než dostupný, využije se jejich maximální dostupný počet, jejichž hodnota se odečte od zůstatku, nebo, je-li dostupný požadovaný počet, využije se k vrácení a jeho hodnota se odečte od zůstatku. V obou případech se poté pokračuje ve výpočtech s dvacetikorunou atd.) zjistí, zda lze uživateli vrátit zůstatek v dostupných mincích. Pokud ano, pokračuje přes akci VRAT_ZBYTEK do stavu VYDEJ_KELIMEK. V opačném případě pokračuje akcí CHCI_PLATIT_VIC .

4.3.16 Akce CHCI_PLATIT_VIC

Akce CHCI_PLATIT_VIC přesune uživatele k volbě zaplacení vyšší částky ve stavu PLATIT_VIC.

4.3.17 Akce VYDANI_KELIMKU

Akce VYDANI_KELIMKU zkontroluje dostupnost potřebných komponent, vydá kelímek, chvíli počká, aby mohl uživatel případně zrušit objednávku, a poté zkontroluje, zda kelímek úspěšně dopadl. Pokud ano, pokračuje do stavu OHREV_VODY. Pokud ne, vrátí uživateli vklad a přesune se zpět do stavu START.

4.3.18 Akce OHREVANI_VODY

Akce OHREVANI_VODY zkontroluje dostupnost potřebných komponent a začne ohřívat vodu. Pokud do dosažení požadované teploty uživatel nezruší objednávku, přesune automat do stavu ODSYP_KAVY.

4.3.19 Akce ODSYPANI_KAVY

Akce ODSYPANI_KAVY začne odsypávat kávu a po chvíli automat přesune do stavu ODLITI_NAPOJE.

4.3.20 Akce ODLEVANI

Akce ODLEVANI odleje kávu do kelímku a po chvíli přesune automat do stavu VRATIT_ZBYTEK.

4.3.21 Akce VYDANI_KAFE

Akce VYDANI_KAFE čeká, než uživatel odebere kelímek, a přesune automat zpátky na START.

5 Struktura archivu

5.1 složka app

- soubor `Uno_Trio.jar`, v němž jsou přeložené zdrojové kódy
- dávkovací soubor `unotrio.bat` sloužící pro spuštění programu

5.2 složka doc

- soubor `Dokumentace.pdf` s touto dokumentací

5.3 složka src

- obsahuje zdrojové soubory (v podsložce `resources`)

6 Uživatelská dokumentace

Pro spuštění programu spusťte dávkovací soubor „unotrio.bat”.

Po spuštění programu se vypíší všechny teplé nápoje, které lze v automatu objednat, jejich cena i dostupnost. Pod tímto výpisem následuje další výpis, a to výpis všech akcí, které lze provést.

```
-----
0 - Espresso, cena - 10 Kc, dostupnost = dostupne
1 - Ristretto, cena - 15 Kc, dostupnost = dostupne
2 - Lungo, cena - 18 Kc, dostupnost = dostupne
3 - Espresso doppio, cena - 13 Kc, dostupnost = dostupne
4 - Cappuccino, cena - 25 Kc, dostupnost = dostupne
5 - Latte, cena - 20 Kc, dostupnost = dostupne
6 - Latte macchiato, cena - 22 Kc, dostupnost = dostupne
7 - Americano, cena - 30 Kc, dostupnost = dostupne
8 - Irská káva, cena - 25 Kc, dostupnost = dostupne
9 - Turecká káva, cena - 8 Kc, dostupnost = dostupne
10 - Čaj citrónový, cena - 5 Kc, dostupnost = dostupne

Vstup je ve formátu P=akce
P=M-x : Vhození mince, kde X značí sumu
P=D : Vypis poctu minci
P=H : Vypis dostupnosti kafe
P=T-x : Zmáčknutí tlačítka, kde X značí id kafe
P=B : Vrácení všech mincí, ukončení nákupu
P=I : Vypsání používání
P=V : Simulace dosla voda
P=S : Simulace dosel cukr
P=K : Simulace dosly kelímky
P=E : Chybovy stav (pro testovací ucely)
P=C-x : Simulace doslo kafe, kde X značí id kafe
```

Obrázek 2 - výpis informací při spuštění programu

Dalším krokem může být vhození mincí. K tomuto účelu slouží akce P=M-x, kde x značí sumu, kterou do automatu vhodíme.

```
Aktualni stav: START
P=M-20
Aktuální stav peněženky = 20
Aktualni stav: PENEZENKA
```

Obrázek 3 - vhození dvaceti korun do automatu

Následně si můžeme nechat vypsát počet mincí pomocí akce $P=D$.

```
P=D
1: 50
2: 30
5: 10
10: 20
20: 6
50: 30
Aktuální stav peněženky = 20
Aktuální stav: PENEZENKA
```

Obrázek 4 - vypsání počtu mincí v automatu

V této chvíli můžeme buď zvolit teplý nápoj, který si chceme koupit nebo si nákup rozmyslet a nechat si vrátit peníze. V případě rozmyšlení nákupu použijeme akci $P=B$. V takovém případě se nám vrátí peníze a znovu vypíše nabídka jako při spuštění.

```
P=B
Aktuální stav peněženky = 20
Aktuální stav: VRATIT_PENIZE
Vracím peníze: 20
0x 50
1x 20
0x 10
0x 5
0x 2
0x 1
```

Obrázek 5 - vypsání počtu mincí k vrácení

V opačném případě si vybereme teplý nápoj, který má automat připravit. Například když si vybereme citronový čaj, provedeme akci $P=T-10$, protože citronový čaj má číslo 10.

```
P=T-10
Aktuální stav peněženky = 20
Aktuální stav: NAPOJ_ZVOLEN
Simulace vyberu kafe:
P=V : Odeber vodu
P=C-x : Simulace doslo kafe, kde X značí id kafe
P=B : Vrat peníze
P=P : Pokračovat ve výrobě
```

Obrázek 6 - zvolení čaje

Pomocí akcí P=P budeme pokračovat ve výrobě čaje.

```
Aktualni stav: ROZMENENI
Simulace vyberu kafe:
P=V   : Odeber vodu
P=C-x  : Simulace doslo kafe, kde X značí id kafe
P=B   : Vrat penize
P=P   : Pokračovat ve výrobě
```

Obrázek 7 - rozměnění

```
Aktualni stav: VYDEJ_KELIMEK
Simulace vyberu kafe:
P=V   : Odeber vodu
P=K   : Odeber kelímek
P=B   : Vrat penize
P=P   : Pokračovat ve výrobě
```

Obrázek 8 - výdej kelímku

```
Aktualni stav: OHREV_VODY
Simulace vyberu kafe:
P=V   : Odeber vodu
P=B   : Vrat penize
P=P   : Pokračovat ve výrobě
```

Obrázek 9 - ohřev vody

```
P=P
Aktualni stav: ODSYP_KAVY
Odsyp kavy
Aktualni stav: ODLITI_NAPOJE
Odlevani kavy
Aktualni stav: VRATIT_ZBYTEK
Vraceni penez
Vracim peníze: 15
0x 50
0x 20
1x 10
1x 5
0x 2
0x 1
```

Obrázek 10 - příprava nápoje a vrácení peněz


```
Aktualni stav: VYDEJ_KAFE
Prosim odeberte kafe
```

Obrázek 11 - výdej nápoje

V případě, že v automatu není vhozeno dost peněz pro nákup požadovaného nápoje, vám automat vypíše chybovou hlášku (obr. 12) a příkazem $P=M-x$ můžete vhodit další mince do automatu.

```
Aktualni stav: START
P=M-1
Aktuální stav peněženky = 1
Aktualni stav: PENEZENKA
P=T-3
Aktuální stav peněženky = 1
Aktualni stav: NAPOJ_ZVOLEN
Nemate dostatek penez na nakup tohoto kafe
Aktualni stav: PENEZENKA
```

Obrázek 12 - nedostatek peněz

V programu se dají nasimulovat i reálné problémy, které mohou nastat. K tomu slouží akce $P=V$ (došla voda), $P=S$ (došel cukr), $P=K$ (došly kelímky), $P=C-x$ (došel teplý nápoj číslo x) a $P=E$ pro rychlý přechod do chybového stavu pro testování.

Pokud zvolíme některou z těchto akcí, vypíše se informace o tom, co je špatně a jaké akce je možné provést. Po vyřešení problému můžeme automat uvést zpět do provozu akcí $P=A$.

Pomocí akce $P=I$ lze vždy vypsat všechny akce, které jsou při používání automatu možné.

7 Závěr

Ve výsledném programu jsme neimplementovali možnost zvolit si množství cukru, což by byl vhodný příklad případného rozšíření.

Dále by bylo možné vytvořit uživatelské rozhraní pro jednodušší ovládání a větší přehlednost.

Rozdělení práce:

- Vojtěch Danišík - program
- Jakub Černý - program
- Michaela Janková - dokumentace