Diskrétní simulace 2

přednáška NPRG031 Programování 2 11. 3. 2020

Minule jsme viděli, jak pomocí **modelu** a **diskrétní simulace** můžeme získat odpovědi na některé otázky.

Návrh modelu

Co kdybychom místo stěhování písku chtěli modelovat něco jiného?

- hlavní cyklus zůstane stejný
- změní se
    - reprezentace a inicializace modelu (PisekA, PisekB...)
    - jaké **procesy** budou v modelu
      (proces říkáme tomu, co se v našem modelu pohybuje)
    - jaké **události** budou v modelu
    - jak budou jednotlivé procesy zpracovávat události
      třeba AUTO a událost ZAČNI_NAKLÁDAT:
        - *zjisti kolik můžeš naložit – min(**PisekA**, **nosnost**)*
        - *pokud je to <=0, tak skonči*
        - *jinak nastav vlastnost **kolikVezu***
        - ***PisekA** zmenši o **kolikVezu***
        - *naplánuj si událost NALOŽENO na okamžik **Čas+dobaNakládání***

Když máme tohle vymyšleno, máme určitým způsobem hotovo, protože pak už zbývá to jen (mechanicky) přepsat do zdrojového kódu (a je vcelku jedno, v jakém jazyku) – na začátku roku jsem říkal, že potkáte různé významy slova „programování" a tohle je jeden z nich.

„Návrh" naznačuje, že neexistuje jedno správné řešení, které máme najít, ale spíše (nekonečně) mnoho možností, z nichž si můžeme vybírat a porovnávat klady a zápory, které má ta která možnost.
Proto je někdy těžké jednoznačně určit, který návrh je dobrý a který špatný.

Podívejme se na příklad.

Úloha

Obchodní dům sestává z několika **oddělení** ležících v různých **patrech**, mezi nimiž jezdí **výtah.** Do obchodního domu přicházejí **zákazníci** se svými **seznamy oddělení**, která chtějí navštívit. V každém oddělení obsluhuje jedna **prodavačka,** která má svou **rychlost** udávající, za jak dlouho obslouží jednoho zákazníka, zákazníci čekají ve **frontě** a pokud zákazník čeká déle, než je jeho **trpělivost**, odejde z fronty a jde jinam, pokud to už není jeho poslední nákup.
Na výtah zákazníci také čekají ve frontě, ale tady se trpělivost neuplatňuje. Výtah se řídí **algoritmem výtahu** (směr jízdy mění jen

tehdy, když neveze žádné pasažéry a zároveň na něj nikdo nečeká ve směru jízdy a zároveň na něj někdo čeká ve směru, odkud přijel).

Cíl: Chceme spočítat, v kolik hodin odejde poslední zákazník.

Příklad vstupních dat

```
Vstupní data pro simulaci obchodního domu
 ========================================

 Oddělení:
    jméno          patro     rychlost obsluhy
    ---------------------------------------
O   papírnictví      0          5
O   potraviny        0         15
O   drogerie         0          6
O   textil           1         15
O   nábytek          2         20
O   elektronika      3          5
O   CD-DVD           3          4

 Výtah
    jméno          kapacita  doba nastoupení    doba vystoupení   doba jízdy z patra do patra
    ------------------------------------------------------------------------------------------
V   elefant          10            1                   1                   1

 Zákazníci
    jméno          příchod  trpělivost seznam oddělení...
    ------------------------------------------------------------------------------------------
Z   Adam            10        1        CD-DVD
Z   Alžběta         10        5        textil drogerie nábytek
Z   Bořík           10        1        papírnictví elektronika
Z   Barborka        10        1        CD-DVD papírnictví
Z   Cyril           10        2        potraviny papírnictví
Z   Cecilka         10        5        potraviny textil drogerie nábytek
Z   David           11        3        CD-DVD elektronika
Z   Dana            11        2        potraviny textil drogerie nábytek
```

Příklad návrhu

Začneme **seznamem procesů**, v tomto případě navrhujeme, aby model obsahoval procesy

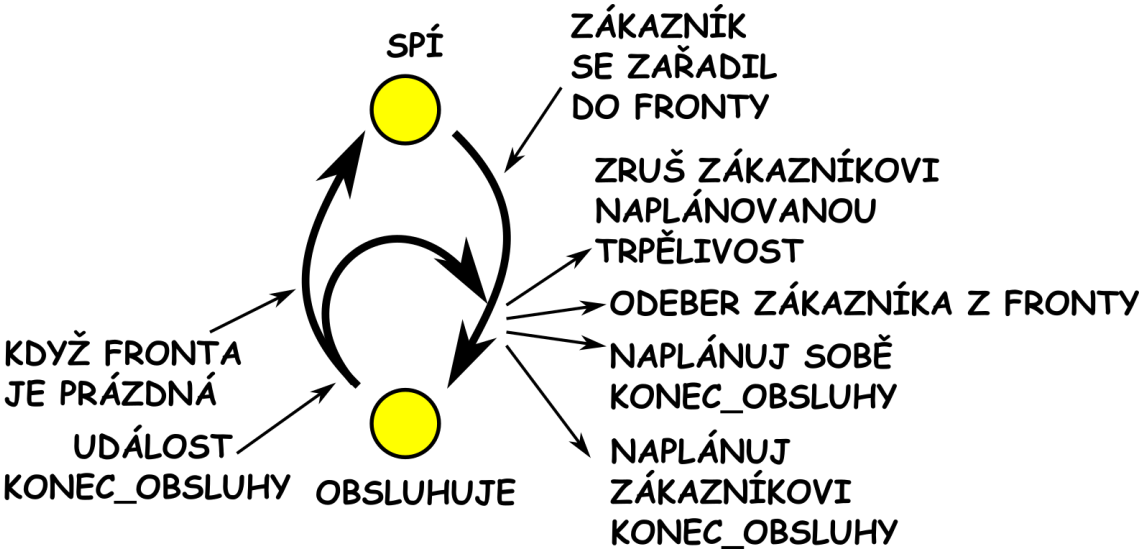
- **zákazník**
- **oddělení**
- **výtah**

Pro návrh událostí můžeme jako pomůcku použít **diagram stavů** (též „stavový diagram“) znázorňující **stavy**, ve kterých se proces může nacházet a **přechody** mezi nimi.

U každého přechodu nás potom zajímá

- kdo ho spustí
- co je při něm potřeba udělat

Například oddělení se může nacházet ve stavech **obsluhuje** a **spí.**



Takhle může vypadat jedna z mnoha možností.
Jiná by byla, že zákazník nebude mít konec obsluhy (ale kdy si označí vyřízený nákup?) nebo... nebo...

Protože máme několik typů procesů, bude se nám hodit abstraktní třída **proces,** která bude společným předkem všech tříd procesů a která nám dovolí (polymorfismus!) obsluhovat události tak, že je vybereme z kalendáře a necháme je zpracovat ten objekt, který je v události uveden.

Protože třída proces slouží jenom jako společný jmenovatel a nikdy od ní nebudeme vytvářet instance/objekty, bude **abstraktní.**

Použití společného předka a polymorfismu má tu výhodu, že pokud bychom v budoucnosti chtěli přidat další typ procesu, nebudeme kvůli tomu muset měnit hlavní cyklus zpracování události.

Implementace a cvičení

1. úkol
Stáhněte si zdrojový kód
https://ksvi.mff.cuni.cz/~holan/py/obchod.txt
a vstupní data
https://ksvi.mff.cuni.cz/~holan/py/obchod_data.txt ,
vytvořte si nový projekt, nakopírujte zdrojový kód a soubor se vstupními daty přidejte do projektu

(ve VisualStudiu v Průzkumníku řešení (Solution Explorer) stačí soubor přetáhnout na název **projektu** a ve Vlastnostech (Properties) tomuto souboru nastavte vlastnost **Kopírovat do výstupního adresáře** na **Kopírovat, pokud je novější** (dtto anglicky) – tím zařídíte,

a) že vstupní soubor máte mezi zdrojovými texty
b) že se při překladu nakopíruje tam, kde bude přeložený a spouštěný program.

Spusťte program a přesvědčte se, že počítá.

2. úkol
Prozkoumejte zdrojový kód tak, abyste si byli jisti, že všemu rozumíte. Pokud ne, tak se ptejte (viz dále)

3. úkol
Zkuste si rozmyslet návrh jiného modelu, třeba lyžařské středisko, kde lidé stojí fronty u vleků, u kiosků s občerstvením, toalet a půjčoven lyží a sjíždějí po různých sjezdovkách.

Otázky a odpovědi

Pokud máte nějaké dotazy k přednášce, podívejte se na stránku
https://ksvi.mff.cuni.cz/~holan/py/otazky_200311.html
a jestli tam vaše otázka už není zodpovězená, pošlete mi mail a předmět zprávy nastavte na **Pgm-2: 200311**.

Hodně zdraví!

Tomáš Holan