## Žilinská univerzita v Žiline Fakulta riadenia a informatiky

**DISKRÉTNA SIMULÁCIA**

Semestrálna práca č. 2

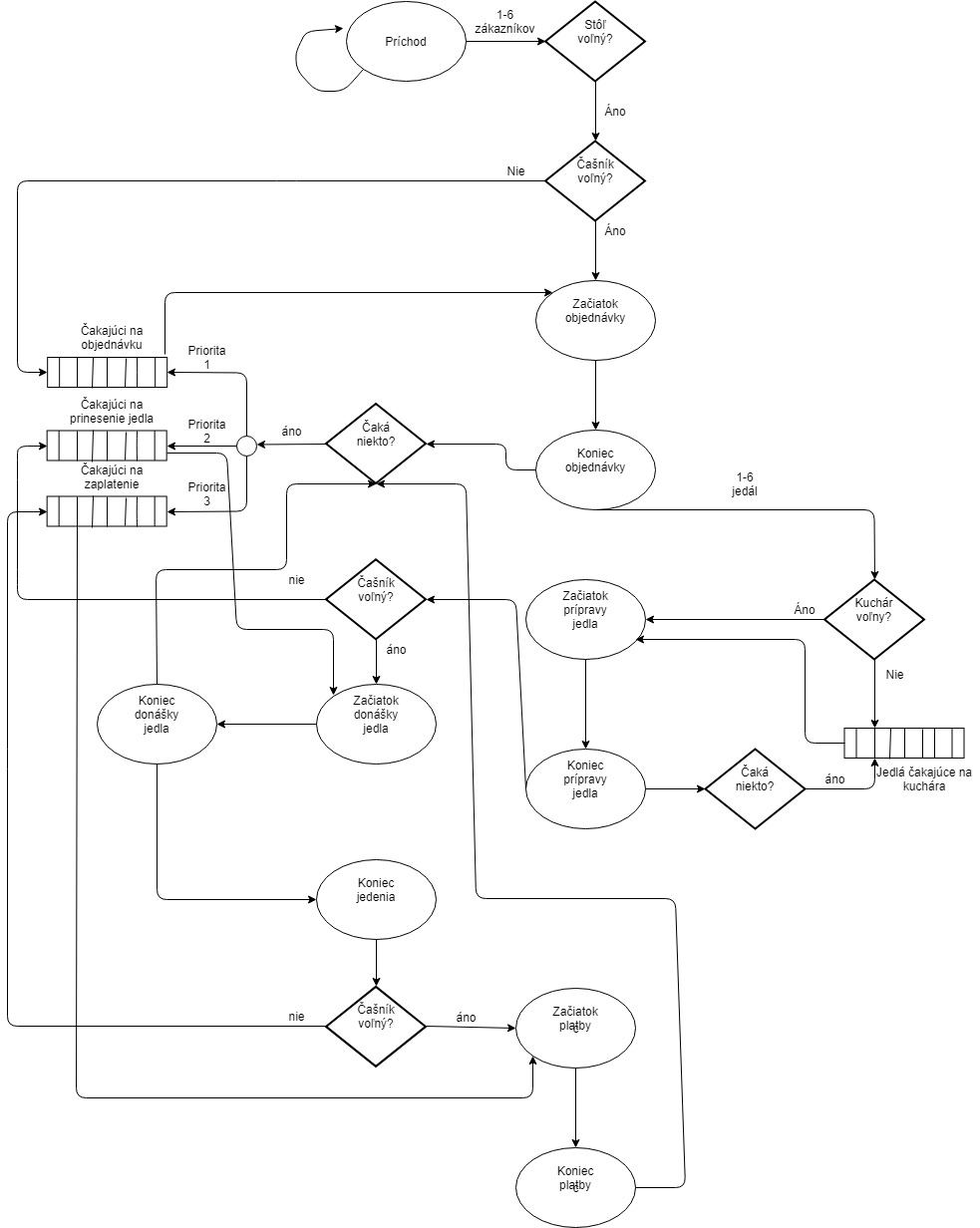
Marek Zaťko 2018/2019

# Popis problému

Zadaním bolo vytvorenie simulačnej štúdie pre potreby novo otvorenej prevádzky reštaurácie. V simulačnej štúdií bolo potrebné odporučiť vhodný počet kuchárov a čašníkov na základe vstupných informácií, ktoré sú v zadaní.

# Popis implementácie

Podľa zadania bola vytvorená udalostne orientovaná simulácia, naprogramovaná v jazyku Java, grafické rozhranie je implementované pomocou JavaFX. Pre správne namodelovanie simuláciu sme vytvorili 10 eventov.



Všetky udalosti majú predka podľa toho o aký typ udalosti sa jedná. Máme 3 typy eventov:

1. **CustomerEvent** – predok pre eventy, týkajúce sa len zákazníkov, dedia ho:

* ArrivalEvent - udalosť príchodu zákazníckej skupiny
* FoodFinishedEvent – koniec jedenia zákazníckej skupiny

2. **CustomerWaiterEvent** – predok pre eventy, týkajúce sa zákazníkov a čašníkov:

* StartOfOrderEvent – začiatok objednávky
* EndOfOrderEven – koniec objedávky
* StartOfFoodDeliverEvent – začiatok donášky jedla
* EndOfFoodDeliverEven – koniec donášky jedla
* StartPayingEvent – začiatok platenia
* EndOfPayingEvent – koniec platenia

3. **CustomerCookEvent** – predok pre eventy, týkajúce sa zákazníkov a kuchárov:

* StartCookingEvent – začiatok prípravy jedla
* FoodPreparedEvent – koniec prípravy jedla

Všetky typy eventov majú predka základný event simulačného jadra Event. Simulačné jadro obsahuje aj SystemEvent, ktorý slúži na to aby sa simulácia tvárila ako spojitá a takisto na jej spomaľovanie. Samotné simulačné jadro udalostne orientovanej simulácie je implementované všeobecne a to v **balíčku event\_sim\_core**, balíček obsahuje generátory, triedy na podporu štatisík a samotnú všeobecnú implementáciu jadra. Samotná konkrétna implementácia simulácie reštaurácie sa nachádza v **balíčku restauracia**.

Aplikácia podporuje 3 režimy:

1. Sledovanie simulačného behu – sledovanie behu simulácie, zobrazuje rady zákazníkov, momentálne činnosti zamestnancov, zákazníkov, možnosť spomalenia simulácie
2. Replikačný mód – vykoná stanovený počet replikácií maximálnej rýchlosti, a zobrazí priemerné výsledky
3. Zobrazenie grafov zo zadania – ukazuje graf závislosti času čakania zákazníka od počtu čašníkov/kuchárov.

# Výsledky simulačnej štúdie

## Popis chladenia

V práci bolo implementované chladenie a to z dôvodu, že reštaurácia reálne nezatvorí prevádzku presne o 22:00, ale obslúži všetkých zákazníkov, ktorí sa už nachádzajú v reštaurácií v dobe zatvorenia. Implementované to je tak, že udalosť príchodu neplánujeme po 22:00 a simuáciu necháme dobežať dokým nebude kalendár udalostí prázdny. Možnosť či chceme chladenia je v aplikácií vstupom do simulácie.

## Popis výsledkov

Zadanie požaduje aby čas čakania zákazníka bol menší ako 12 minút a aby neodišlo viac ako 17% zákazníkov(zákazník odíde ak sú všetky stoly vyhovujúce kapacite zákazníckej skupine obsadené). Pre tieto požadované hodnoty máme stanoviť čo najmenší počet čašníkov a kuchárov s tým, že prioritne sa snažíme znížiť počet čašníkov. Keďže simulácia obsahuje prvky náhody, pre každú výslednú hodnotu stanovíme aj 90% spoľahlivosti pri **10 000 replikáciách**. **Všetky výsledky a štatistiky je možné si pozrieť v priložených excelovských súboroch, v dokumentácií budú uvedené len najzaujímavejšie hodnoty času čakania a počtu odídených zákazníkov**.

## Výsledky bez chladenia

|  |  |  |  |
| --- | --- | --- | --- |
| Počet čašníkov | Počet kuchárov | Priemerný čas čakania zákazníka | Priemerný počet odídených zákazníkov  (%) |
| 5 | 14 | 718.21622  IS (90%): <717.82074 : 718.61171> | 17.08818  IS (90%): <17.02678 : 17.14958> |
| 6 | 14 | 717.44877 I  IS (90%): <717.05331 : 717.84423> | 17.04286  IS (90%): <16.98244: 17.103291> |
| 7 | 14 | 717.54877  IS (90%): <717.14973 : 717.94781> | 17.08254  IS (90%): <17.02154 : 17.14354> |

Vidíme, že počet odídených zákazníkov nie je menej ako 17% a s pribúdajúcim počtom čašníkov pri 14 kuchároch sa sledované hodnoty neznižujú, preto pridáme jedného kuchára.

|  |  |  |  |
| --- | --- | --- | --- |
| Počet čašníkov | Počet kuchárov | Priemerný čas čakania zákazníka | Priemerný počet odídených zákazníkov  (%) |
| 3 | 15 | 720.181037  IS (90%): <719.83720 : 720.5248699> | 17.2964273  IS (90%): <17.23590 : 17.35694> |
| **4** | **15** | **701.14448**  **IS (90%): <700.80169 : 701.48726>** | **16.7540916**  **IS (90%): <16.694479 : 16.813703>** |
| 5 | 15 | 697.242427 I  S (90%): <696.902280 : 697.58257> | 16.646398  IS (90%): <16.586512 : 16.706285> |
| 6 | 15 | 696.567273  IS (90%): <696.22568 : 696.90886> | 16.5732351  IS (90%): <16.51341 : 16.633056> |
| 3 | 16 | 705.87546  IS (90%): <705.573184 : 706.17774> | 16.90321  IS (90%): <16.84305 : 16.96335> |
| 4 | 16 | 686.4811474  IS (90%): <686.181256 : 686.78103> | 16.432434  IS (90%): <16.372606: 16.49226> |

## Optimálny počet

Z výsledkov vidíme, že požadované hodnoty je možné dosiahnuť najmenším počtom zamestnancov 19 a to konkrétne:

1. 4 čašníci a 15 kuchári
2. 5 čašníci a 14 kuchári

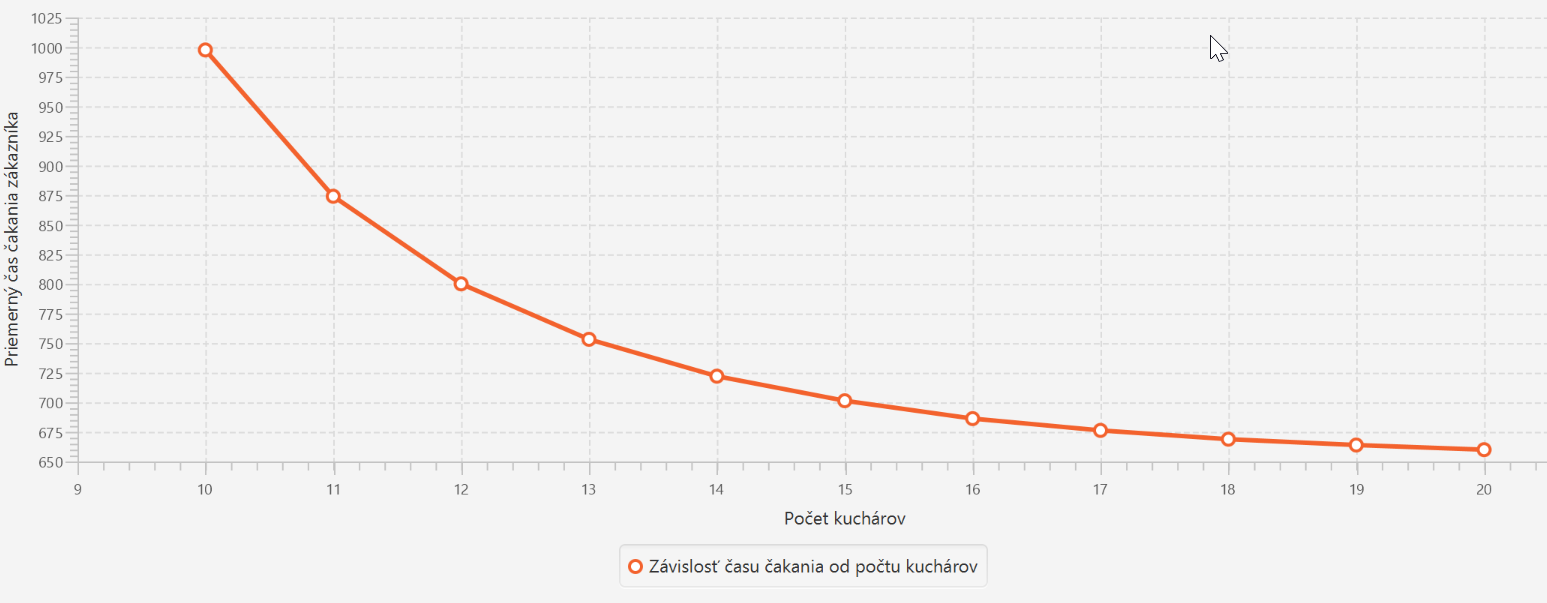
Keďže zadanie hovorí, že prioritne sa snažíme znížiť počet čašníkov tak ako optimálny počet odporúčame **4 čašníkov a 15 kuchárov**.

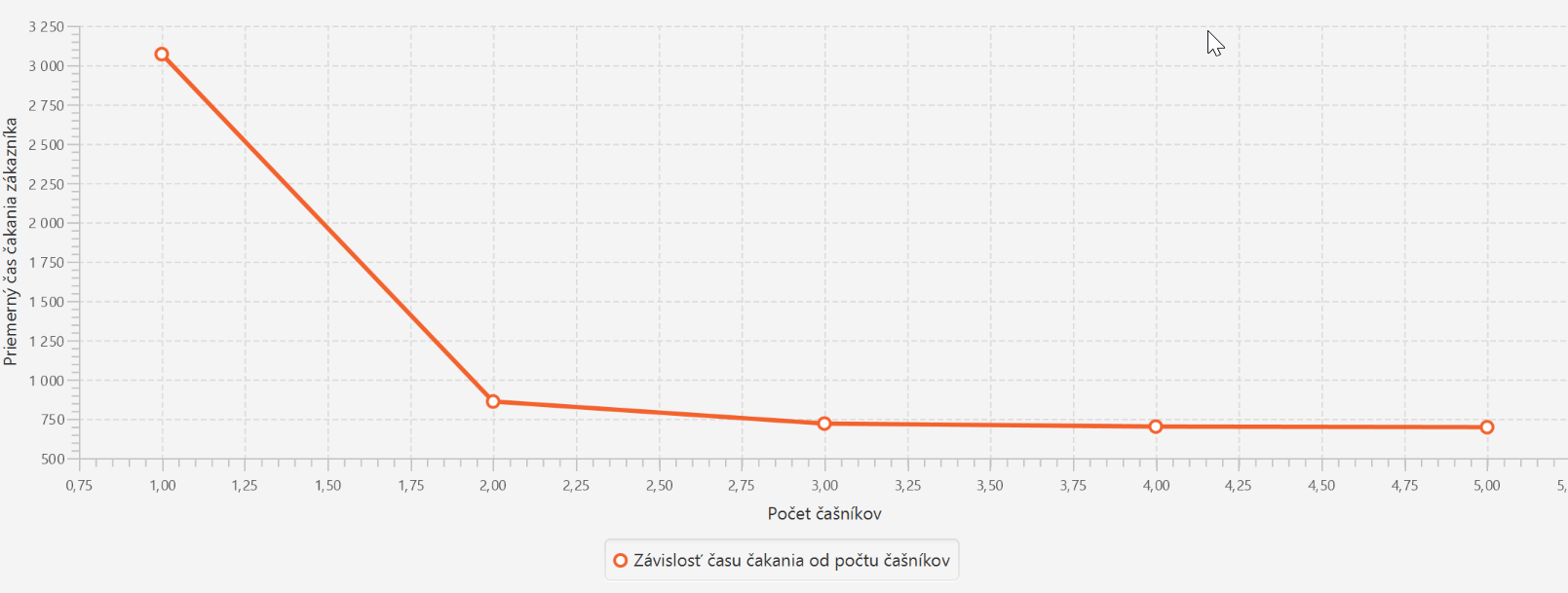
Tento počet je optimálny aj v prípade chladenia, nami popísané chladenie sa na výsledkoch prejaví len v prípade čakania zákazníka a to zvýšením priemernej hodnoty o priblížne 2 sekundy, čo je stále menej ako stanovená horná hranica, teda optimálny počet zostáva rovnaký.

## Grafy časovej závislosti

Pre nami odporúčaný počet **4 čašníkov a 15 kuchárov**, sme zostavili aj grafy závislosti priemernej doby čakania zákazníka od variabilného počtu zamestnancov:

Pre variabilný počet kuchárov, je rozsah <10;20>, na každý bod v grafe bolo vykonaných 10 000 replikácií.



Pre variabilný počet čašníkov je rozsah <1;5> :