**ŽILINSKÁ UNIVERZITA V ŽILINE**

**FAKULTA RIADENIA A INFORMATIKY**

DISKRÉTNA SIMULÁCIA

Semestrálna práca č.2

Udalostne orientovaná simulácia

Adrián Václavek

5ZIS11

**2021**

Obsah

[Úvod 3](#_Toc68567111)

[Analýza zadania 4](#_Toc68567112)

[Udalostný diagram 5](#_Toc68567113)

[Generátory náhodných čísel 7](#_Toc68567114)

[Overenie generátora náhodných čísel rovnomerného rozdelenia 7](#_Toc68567115)

[Overenie generátora náhodných čísel exponenciálneho rozdelenia 8](#_Toc68567116)

[Overenie generátora náhodných čísel trojuholníkového rozdelenia 9](#_Toc68567117)

[Popis balíčkov 10](#_Toc68567118)

[Simulation 10](#_Toc68567119)

[EventSimulation 10](#_Toc68567120)

[Generators 10](#_Toc68567121)

[VaccinationCentreSimulation 10](#_Toc68567122)

[Entities 10](#_Toc68567123)

[Events 10](#_Toc68567124)

[GUI 10](#_Toc68567125)

[Popis simulačného jadra 11](#_Toc68567126)

[Experimenty 13](#_Toc68567127)

[Experiment číslo 1: 13](#_Toc68567128)

[Experiment číslo 2: 13](#_Toc68567129)

[Experiment číslo 3: 16](#_Toc68567130)

[Záver 19](#_Toc68567131)

# Úvod

Cieľom práce bola implementácia udalostnej orientovanej simulácie vakcinačného centra. Pacient sa po príchode do systému musí zaregistrovať, pričom si náhodne vyberie jedného z dostupných administratívnych pracovníkov. Ak taký neexistuje, postaví sa do radu a čaká kým sa nejaký pracovník uvoľní. Po registrácii ide na lekárske vyšetrenie, kde si taktiež vyberie jedného z dostupných doktorov, alebo sa postaví do radu a čaká kým bude niektorý doktor dostupný. Doktor pacientovi na základe výsledku vyšetrenia povie, koľko minút má pacient po vakcinácii čakať v centre. Po vyšetrení pacient ide na samotnú vakcináciu. Náhodne si vyberie dostupnú sestru, alebo sa postaví do radu a čaká. Po vakcinácii sa presunie do čakacej miestnosti, kde strávi čas, ktorý mu doctor nakázal. Následne pacient odíde zo systému.

# Analýza zadania

Ako známe parametre programu boli zadané:

* Intenzita vstupného toku pacientov v rámci času obsluhy centra
* Centrum je v prevádzke od 8:00 do 17:00
* Doba obsluhy administratívnym pracovníkom je modelovaná rovnomerným rozdelením na intervale <140,220>
* Doba obsluhy doktorom je modelovaná exponenciálnym rozdelením so strednou hodnotou 260 sekúnd
* Doba obsluhy sestrou je modelovaná trojuholníkovým rozdelením (20 sekúnd, 100 sekúnd, 70 sekúnd)
* Doba v čakárni pre jednotlivých pacientov na základe pokynov doktora je pre 95% pacientov 15 minút a pre 5% pacientov 30 minút

V aplikácii musíme pozorovať tieto štatistiky:

Priemerný počet ľudí:

* v rade na registráciu
* v rade na vyšetrenie doktorom
* v rade na vakcináciu
* v čakacej miestnosti

Priemerné vyťaženie personálu:

* pri registrácii
* pri vyšetrení
* pri vakcinácii

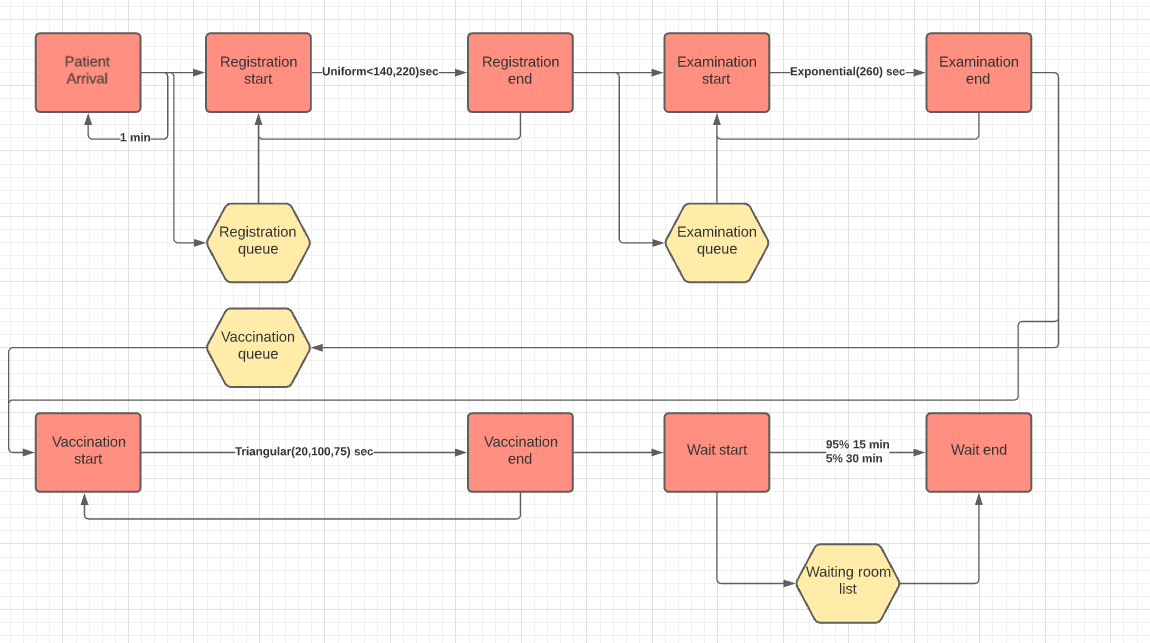
Priemerný čas čakania :

* v rade na registráciu
* v rade na vyšetrenie doktorom
* v rade na vakcináciu
* v čakacej miestnosti

Zároveň je nutné poskytovať aj 95% interval spoľahlivosti pre priemerný počet ľudí v čakárni.

# Udalostný diagram

Základným predpokladom dobre naimplementovanej udalostne orientovanej simulácie je udalostný diagram. Tento diagram môžeme vidieť na nasledujúcom obrázku:



Obdlžníky ružovej farby predstavujú jednotlivé udalosti a šesťuholníky svetložltej farby štruktúry, v ktorých sa pacient nachádza. Každá z týchto udalostí je naplánovaná na konkrétny čas. Udalosti nastávajú v poradí, ktoré je vyznačené šípkami. Ak šipka smeruje do frontu, znamená to, že pre nasledujúcu udalosť nie je splnená podmienka voľného personálu, alebo sa pacient presúva do čakacej miestnosti.

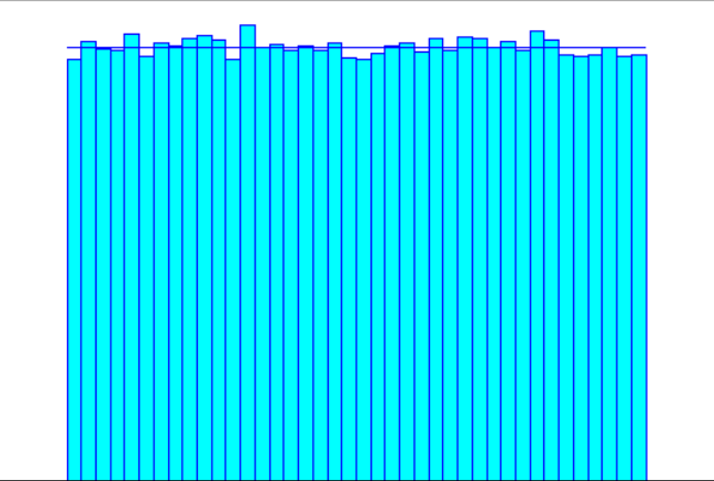
Na diagrame môžeme vidieť celkovo 9 udalostí. Simulácia začína príchodom pacienta do systému. Ak v systéme existuje voľný administratívny pracovník a rad je prázdny, naplánuje sa udalosť Registration start, v opačnom prípade sa pacient presunie do frontu, kde bude čakať na naplánovanie novej udalosti Registration start z udalosti Registration end. Udalosť Patient arrival zároveň plánuje novú udalosť Patient arrival, ktorá naplánuje nový príchod zákazníka podľa intenzity toku príchodov. Udalosť Registration start naplánuje udalosť Registration end na čas, ktorý pochádza z rovnomerného rozdelenia pravdepodobnosti s parametrami <140,220) v sekundách plus aktuálny čas. Po uplynutí tohto času sa v udalosti Registration end podľa toho, či je rad prázdny a je dostupný doktor buď naplánuje Examination start, alebo sa pacient presunie do frontu na vyšetrenie. Následne prebieha rovnaký proces ako pri registračných udalostiach, s tým rozdielom, že čas udalosti Examination end pochádza z exponenciálneho rozdelenia pravdepodobnosti so strednou hodnotou 260. Zároveň sa v tejto udalosti pacientovi určí, či má čakať 900 alebo 1800 sekúnd po vakcinácii. Udalosť Examination end ďalej rozhoduje o tom či sa pacient presunie do frontu pre vakcináciu, alebo sa rovno naplánuje udalosť Vaccination start, na základe prázdneho vakcinačného radu a dostupných zdravotných sestier. Zároveň plánuje začiatok novej udalosti Examination start. Ak nastane udalosť Vaccination start, naplánuje sa udalosť Vaccination end na čas, ktorý pochádza z trojuholníkového rozdelenia s minimom 20 sekúnd, maximom 100 sekúnd a módusom 70 sekúnd, plus aktuálny simulačný čas. Pri vykonaní udalosti Vaccination End sa naplánuje udalosť Vaccination Start na rovnaký čas aký má Vaccination End, pričom sa taktiež na rovnaký čas naplánuje udalosť Wait start, pri ktorej sa pacient presunie do čakacieho listu. Pri vykonaní Wait start sa zároveň naplánuje udalosť Wait End na aktuálny čas plus čas, ktorý doktor predpísal pacientovi na čakanie. V udalosti Wait End sa pacient vyberie z čakacieho listu, čím odchádza zo systému.

# Generátory náhodných čísel

V tejto semestrálnej práci využívame čísla, ktoré pochádzajú z určitých rozdelení pravdepodobností. Konkrétne sa jedná o rovnomerné, exponenciálne a trojuholníkové rozdelenie. Z tohto dôvodu vyplýva nutnosť implementácie generátorov náhodných čísel, ktoré budú disponovať možnosťou poskytnutia takýchto potrebných dát. Keďže je požadované, aby tieto čísla boli kvalitné, je potrebné ich overiť pomocou štatistických testov, či implementované generátory naozaj produkujú čísla z daného rozdelenia. Na účely kontroly bol použitý softvér Arena Input Analyzer.

## Overenie generátora náhodných čísel rovnomerného rozdelenia

Nasledujúci obrázok obsahuje 100 000 hodnôt vygenerovaných generátorom náhodných čísel rovnomerného rozdelenia s parametrami <140,220) , ktorý je použitý v aplikácii.

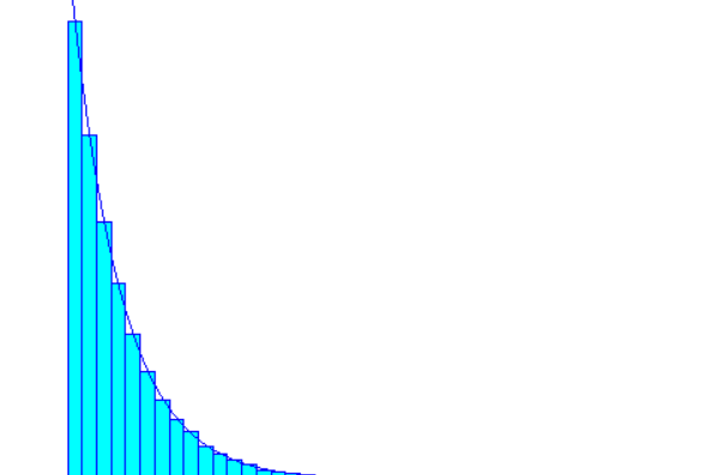
****

|  |  |  |  |
| --- | --- | --- | --- |
| Počet intervalov | Stupne voľnosti | Testovacia štatistika | p-hodnota |
| 40 | 39 | 35.5 | 0.628 |

Po použití funkcie FIT Input Analyzer napasoval vygenerované hodnoty tohto rozdelenia na rovnomerné rozdelenie s požadovanými parametrami. Generátor teda môžeme považovať za valídny.

## Overenie generátora náhodných čísel exponenciálneho rozdelenia

Nasledujúci obrázok obsahuje 100 000 hodnôt vygenerovaných generátorom náhodných čísel exponenciálneho rozdelenia s parametrom (260) , ktorý je použitý v aplikácii.

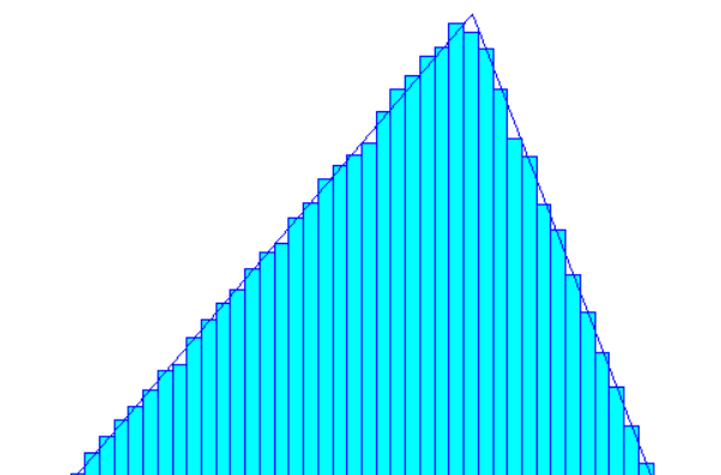
****

|  |  |  |  |
| --- | --- | --- | --- |
| Počet intervalov | Stupne voľnosti | Testovacia štatistika | p-hodnota |
| 30 | 28 | 49.6 | 0.00756 |

Po použití funkcie FIT Input Analyzer napasoval vygenerované hodnoty tohto rozdelenia na exponenciálne rozdelenie s požadovanými parametrami. Generátor teda môžeme považovať za valídny.

## Overenie generátora náhodných čísel trojuholníkového rozdelenia

Nasledujúci obrázok obsahuje 100 000 hodnôt vygenerovaných generátorom náhodných čísel trojuholníkového rozdelenia s parametrami (20, 75, 100) , ktorý je použitý v aplikácii.

****

|  |  |  |  |
| --- | --- | --- | --- |
| Počet intervalov | Stupne voľnosti | Testovacia štatistika | p-hodnota |
| 40 | 38 | 48.2 | 0.132 |

Po použití funkcie FIT Input Analyzer napasoval vygenerované hodnoty tohto rozdelenia na trojuholníkové rozdelenie s požadovanými parametrami. Generátor teda môžeme považovať za valídny.

# Popis balíčkov

Aplikácia disponuje určitým počtom balíčkov, ktoré slúžia na logické členenie tried do skupín. Medzi tieto balíčky patria:

## Simulation

Balíček, ktorý udržiava všeobecné časti programu, teda generické simulačné jadro a generátor náhodných násad. Ďalej sa v tomto balíčku nachádzajú balíčky:

### EventSimulation

Balíček, v ktorom sa nachádzajú všeobecné triedy pre udalostnú simuláciu, teda udalostné jadro, udalosť a vlastná časová výnimka.

### Generators

Balíček, ktorý obsahuje generátory náhodných čísel.

## VaccinationCentreSimulation

Balíček, v ktorom sa nachádzajú triedy, ktoré predstavujú konkrétne využitie prvkov udalostnej simulácie. Nachádza sa v ňom trieda Main, konkrétne simulačné jadro pre simuláciu vakcinačného centra, controller pre GUI a interface delegátov. Ďalej disponuje balíčkami:

### Entities

V tomto balíčku nájdeme entity, ktoré vystupujú v simulácii. Konkrétne sa jedná o pacienta a personál. Zároveň obsahuje enum Location, ktoré poskytuje hodnoty aktuálneho umiestnenia pacienta v systéme.

### Events

Balíček, ktorý obsahuje udalosti obsiahnuté v udalostnom diagrame. Ich spoločným predkom je VaccinationCentreEvent, ktorý je potomkom triedy Event.

### GUI

Balíček, v ktorom sú umiestnené GUI triedy.

# Popis simulačného jadra

V aplikácii existuje niekoľko simulačných jadier, medzi ktorými je použitý vzťah dedičnosti. Najvšeobecnejším z nich je jadro SimulationCore, ktoré disponuje metódou simulate, kde sa vykonávajú základné simulačné metódy:

* beforeSimulation()
* beforeReplication()
* doReplication()
* afterReplication()
* afterSimulation()

Táto konštrukcia nám umožňuje veľmi výhodne konkretizovať riešenie udalostnej simulácie, čím sa dostávame k potomkovi tejto triedy a to EventSimulationCore, ktorý prekrýva metódu doReplication(), pričom poskytuje možnosti nastavenia pauzy a turbo režimu simulácie. V tejto prekrytej metóde sa z kalendára udalostí v cykle vyberie event, nastaví sa aktuálny simulačný čas na čas tohto eventu, vykoná sa event a taktiež poskytuje metódu afterExecute(), ktorá nastane po vykonaní daného eventu. Kalendár udalostí je implementovaný PriorityQueue v jazyku Java.

Potomkom EventSimulationCore je trieda VaccinationCentreSimulationCore, ktorá tvorí konkrétne jadro samotnej aplikácie. Táto trieda prekrýva vyššie spomenuté metódy SimulationCore, ale aj metódy EventSimulationCore. Obsahuje generátory náhodných čísel, ktoré sú inicializované násadou z triedy RandomSeedGenerator. Zároveň je použitý upravený návrhový vzor Observer pre potreby použitia GUI. Rozoberieme si podstatné metódy tejto triedy:

beforeSimulation:

V tejto metóde sa inicializujú všetky atribúty potrebné pre beh a sledovanie simulácie.

beforeReplication:

V tejto metóde sa resetujú atribúty, ktoré služia na sledovanie behu jednej replikácie v simulácii, pričom sa premažú a nanovo vytvoria fronty a listy zamestnancov a pacientov. Zároveň sa vyčistí udalostný kalendár. Simulácia sa teda dostáva do inicializačného stavu.

afterReplication:

Táto metóda slúži na akumulovanie simulačných premenných hodnotami ktoré pochádzajú z aktuálnej replikácie.

afterSimulation:

V tejto metóde sa posiela správa delegátovi na obnovu GUI po ukončení simulačného behu.

refreshGUI:

Metóda, ktorá slúži na obnovu GUI na základe toho, aký event sa práve vykonal.

Na ovládanie tohto jadra slúži trieda AppController, ktorá umožňuje komunikáciu smerom od GUI k jadru. Opačnú komunikáciu umožňuje už spomínaný návrhový vzor Observer cez interface SimDelegate.

# Experimenty

Keďže konečným cieľom každej simulácie je dozvedieť sa niečo nové, poprípade vyvodiť nejaké zistené štatistické výsledky, tak aj v tejto práci vyplýva potreba preukázateľne vykonávať experimenty. Zadané boli tri experimenty.

## Experiment číslo 1:

Zadanie:

*„V súčasnosti pracuje vo vakcinačnom centre 5 administratívnych pracovníkov, 6 lekárov a 3 zdravotné sestry. Namodelujte súčasné fungovanie centra a stanovte tiež priemerné vyťaženie administratívnych pracovníkov, priemerné vyťaženie lekárov a priemerné vyťaženie zdravotných sestier.“*

Pri adekvátnych vstupných údajoch boli po vykonaní 10 000 replikácií a vstupnom toku 540 pacientov namerané nasledujúce výsledky:

|  |  |
| --- | --- |
| Priemerné vyťaženie administratívnych pracovníkov | 55.1298% |
| Priemerné vyťaženie lekárov | 65.1304% |
| Priemerné vyťaženie zdravotných sestier | 33.1813% |

## Experiment číslo 2:

Zadanie:

*„Následne upravte model tak, aby vakcinačné centrum obsluhovalo denne 2500 ľudí. Stanovte také počty jednotlivých typov personálu, aby priemerné vyťaženie personálu bolo čo možno najbližšie súčasnému stavu. Teda aby sa záťaž na pracovníkov nezvýšila.“*

Pre úspešné vykonanie tohto experimentu sme zvolili metódu postupného zvyšovania personálu, pri zvýšenej intenzite toku príchodu pacientov. Z dôvodu mnohých kombinácií vstupných parametrov však nebudeme uvádzať výsledky všetkých pokusov, ale výber zúžime len na množinu signifikantných a pre nás zaujímavých pozorovaní simulačného behu.  
V zásade sme pri vykonávaní tohto experimentu postupovali tak, že sme najprv zvyšovali počet administratívnych pracovníkov, následne počet doktorov a potom počet zdravotných sestier. Tento postup je rozumný z hľadiska toho, že jednotlivé priemerné vyťaženia personálu sú od seba závislé v tomto poradí.

V nasledujúcej tabuľke je uvedené namerané priemerné vyťaženie administratívnych pracovníkov v závislosti od ich počtu po vykonaní 10 000 replikácií a vstupnom toku 2500 pacientov.

|  |  |
| --- | --- |
| Počet administratívnych pracovníkov | Priemerné vyťaženie |
| 20 | 62.6402% |
| 22 | 56.9456% |
| 23 | 54.4697% |
| 24 | 52.2002% |

Z nameraných výsledkov vidíme, že počet administratívnych pracovníkov, pri ktorom je priemerné vyťaženie menšie a zároveň najbližšie k výsledku predchádzajúceho experimentu je **23**. Pri hľadaní požadovaného počtu doktorov teda budeme ďalej simulovať analogicky, pri tomto počte administratívnych pracovníkov.

V nasledujúcej tabuľke je uvedené namerané priemerné vyťaženie lekárov v závislosti od ich počtu po vykonaní 10 000 replikácií a vstupnom toku 2500 pacientov.

|  |  |
| --- | --- |
| Počet lekárov | Priemerné vyťaženie |
| 20 | 90.3628% |
| 23 | 79.0539% |
| 25 | 72.7423% |
| 27 | 67.3563% |
| 28 | 64.9510% |
| 29 | 62.7115% |

Z nameraných výsledkov vidíme, že počet lekárov, pri ktorom je priemerné vyťaženie menšie a zároveň najbližšie k výsledku predchádzajúceho experimentu je **28**. Pri hľadaní požadovaného počtu zdravotných sestier teda budeme ďalej simulovať analogicky, pri tomto počte lekárov a 23 administratívnych pracovníkov.

V nasledujúcej tabuľke je uvedené namerané priemerné vyťaženie zdravotných sestier v závislosti od ich počtu po vykonaní 10 000 replikácií a vstupnom toku 2500 pacientov.

|  |  |
| --- | --- |
| Počet zdravotných sestier | Priemerné vyťaženie |
| 12 | 37.9355% |
| 13 | 35.0174% |
| 14 | 32.5161% |
| 15 | 30.3484% |

Z nameraných výsledkov je viditeľné , že počet zdravotných sestier, pri ktorom je priemerné vyťaženie menšie a zároveň najbližšie k výsledku predchádzajúceho experimentu je **14**.

**Závery experimentu:**

Na základe pozorovania výsledkov simulácie pri viacerých postupných zmenách môžeme skonštatovať, že pri vstupnom toku 2500 vieme najlepšie modelovať jednotlivé vyťaženosti personálu pri počte 23 administratívnych pracovníkov, 28 lekárov a 14 zdravotných sestier. Porovnanie výsledku experimentu s výsledkom predošlého experimentu môžeme vidieť v nasledujúcej tabuľke:

|  |  |  |
| --- | --- | --- |
| Vstupný tok | 540 | 2500 |
| Počet administratívnych pracovníkov | 5 | 23 |
| Počet lekárov | 6 | 28 |
| Počet zdravotných sestier | 3 | 14 |
| Priemerné vyťaženie administratívnych pracovníkov | 55.1298% | 54.4697% |
| Priemerné vyťaženie lekárov | 65.1304% | 64.9510% |
| Priemerné vyťaženie zdravotných sestier | 33.1813% | 32.5161% |

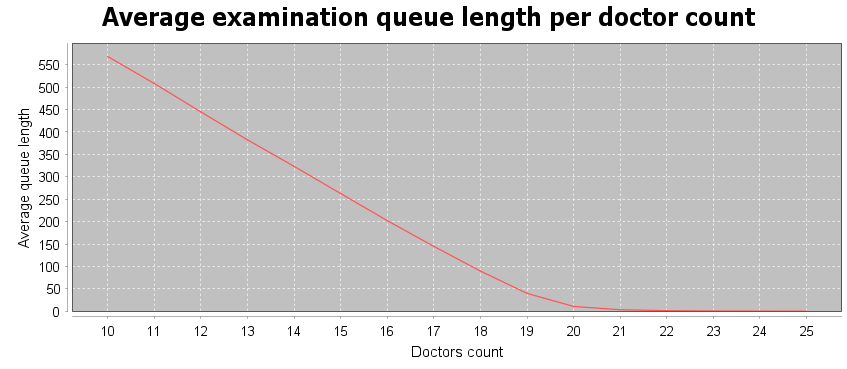
## Experiment číslo 3:

Zadanie:

*„Keďže lekárov je nedostatok, určite aký minimálny počet lekárov musí po navýšení kapacity v centre pracovať, aby priemerný počet osôb čakajúcich na lekárske vyšetrenie neprekročil 12 ľudí a priemerný čas čakania na vyšetrenie neprekročil 15 minút, pričom systém bol stabilný (v priebehu dňa nemá trvalo narastať rad čakajúcich osôb na vyšetrenie). Ďalej graficky (na grafe v programe) dokumentujte závislosť priemerného počtu osôb čakajúcich na lekárske vyšetrenie na počte lekárov (počet replikácií potrebných pre pridanie jedného bodu do grafu ako aj minimálny a maximálny počet lekárov si nastaví užívateľ).”*

Na účely vykonania tohto experiment najprv využijeme grafické znázornenie závislosti priemerného počtu osôb čakajúcich na lekárske vyšetrenie, ktoré máme naimplementované v aplikácii. Ako vstupné parameter uvedieme minimum 10 a maximum 25 lekárov pričom body do grafu budeme pridávať po vykonaní 1000 replikácií s daným počtom lekárov.

Výsledok tohto zobrazenia je uvedený v tomto grafe:



Zároveň údaje pre väčšiu špecificitu budeme reprezentovať formou tabuľky:

|  |  |
| --- | --- |
| Počet lekárov | Priemerná dĺžka radu |
| 10 | |  |  | | --- | --- | |  | 568.2791 | |
| 11 | |  |  | | --- | --- | |  | 507.9988 | |
| 12 | |  |  | | --- | --- | |  | 444.6344 | |
| 13 | |  | | --- | | 382.3417 | |
| 14 | |  |  | | --- | --- | |  | 323.2900 | |
| 15 | |  |  | | --- | --- | |  | 262.5750 | |
| 16 | |  |  | | --- | --- | |  | 202.1517 | |
| 17 | |  |  | | --- | --- | |  | 144.7289 | |
| 18 | |  |  | | --- | --- | |  | 89.6218 | |
| 19 | |  |  | | --- | --- | |  | 39.8950 | |
| 20 | |  |  | | --- | --- | |  | 10.7341 | |
| 21 | |  |  | | --- | --- | |  | 3.3878 | |
| 22 | |  | | --- | | 1.2587 | |
| 23 | |  |  | | --- | --- | |  | 0.5533 | |
| 24 | |  |  | | --- | --- | |  | 0.2565 | |
| 25 | |  |  | | --- | --- | |  | 0.1122 | |

Keďže priemerný počet ľudí v rade úzko súvisí s priemernou dobou čakania v rade, môžeme využiť údaje ktoré nám poskytol graf na radikálne zúženie okruhu skúmania zmeny počtu lekárov v závislosti od požiadaviek. V nasledujúcej tabuľke teda vidíme výsledky nameraných hodnôt, pričom sme menili iba počet lekárov. 3.3878

|  |  |  |
| --- | --- | --- |
| Počet lekárov | Priemerný čas strávený čakaním v rade | Priemerná dĺžka radu |
| 20 | |  | | --- | | 2 min |   41.478 sec | 10.7341 |
| 21 | |  | | --- | | 47.2320 sec | | 3.3878 |

Na základe týchto výsledkov však ešte nevieme určiť, či spĺňame stanovené podmienky, keďže musíme overiť stabilitu systému. Toto overenie vykonáme na základe jednej replikácie pri nezmenených parametroch, pričom budeme sledovať priemernú dĺžku radu počas simulačného behu.

Nasledujúca tabuľka zodpovedá nameraným údajom pri 23 administratívnych pracovníkoch, 20 lekárov, 14 sestier, pričom intenzita vstupného toku je 2500.

|  |  |  |
| --- | --- | --- |
| Simulačný čas | Počet ľudí v rade | Priemerná dĺžka radu |
| 12:00 | 5 | 1.8651 |
| 15:00 | 2 | 3.1255 |
| 17:00 | 5 | 2.5662 |

Keďže v priebehu simulačného behu počet ľudí v rade kolísa okolo priemernej dĺžky radu, môžeme skonštatovať, že systém je stabilný. Nemá preto zmysel pokračovať v skúmaní štatistiky pri 21 lekároch, keďže cieľom tohto experimentu je čo najviac znížiť počet lekárov.

**Závery experimentu:**

Na základe odpozorovaných výsledkov môžeme skonštatovať, že 20 lekárov postačujúco napĺňa ciele tohto experimentu, pričom priemerná dĺžka radu po 1000 vykonaných replikáciách je 10.7341 a priemerný čas čakania v rade je 2 minúty a 41.478 sekúnd.

# Záver

Z naimplementovaného simulačného modelu sme boli schopní úspešne vyprodukovať viacero štatistických výsledkov. Vykonané experimenty splnili svoj účel a priniesli zaujímavé zistenia. Zistili sme aký približne by mal byť počet personálu v jednotlivých častiach vakcinačného centra na to, aby bol systém stabilný a vyťaženosť personálu, priemerné dĺžky radov a priemerný čas strávený v rade zodpovedal požadovaným hodnotám. Zároveň je aplikácia pripravená na vykonávanie ďalších rozličných experimentov.