

FAKULTA RIADENIA A INFORMATIKY  
ŽILINSKÁ UNIVERZITA V ŽILINE

## UDALOSTNÁ SIMULÁCIA

SEMESTRÁLNA PRÁCA Č. 2 Z PREDMETU DISKRÉTNÁ SIMULÁCIA

Autor: **Bc. Matej Poljak**

Cvičiaci: **Ing. Andrea Galadíková, PhD.**

Akademický rok: **2024/2025**

## Obsah

Validácia generátorov .....	5
Generátor pre exponenciálne rozdelenie pravdepodobnosti.....	5
Generátor pre trojuholníkové rozdelenie pravdepodobnosti.....	6
Diagram udalostnej simulácie .....	9
Architektúra riešenia .....	10
Logika.....	10
Grafické užívateľské rozhranie .....	12
Experimenty .....	13
Experiment A.....	13
Experiment B .....	13
Experiment C .....	13
Experiment D.....	13
Experiment E, F, G .....	13
Vyhodnotenie experimentálnej časti .....	14

## Zoznam obrázkov

Obrázok 1 – Input analyzer - vyhodnotenie vzoriek generátora exponenciálneho rozdelenia .....	5
Obrázok 2 - vygenerovanie vzoriek pre exponenciálne rozdelenie programom Input analyzer .....	6
Obrázok 3 - porovnanie výsledkov vzoriek pre exponenciálne rozdelenie pravdepodobnosti .....	6
Obrázok 4 - Input analyzer - vyhodnotenie vzoriek generátora trojuholníkového rozdelenia.....	7
Obrázok 5 - porovnanie výsledkov vzoriek pre trojuholníkové rozdelenie pravdepodobnosti .....	7
Obrázok 6 – diagram udalostnej simulácie výroby nábytku .....	9
Obrázok 7 - udalostná simulácia .....	10
Obrázok 8 - spolupráca tried logiky.....	11
Obrázok 9 - udalosti simulácie výroby nábytku .....	11
Obrázok 10 – simulačné výsledky.....	12
Obrázok 11 - GUI s implementáciou aktualizácie simulačných výsledkov .....	12

## Zoznam tabuliek

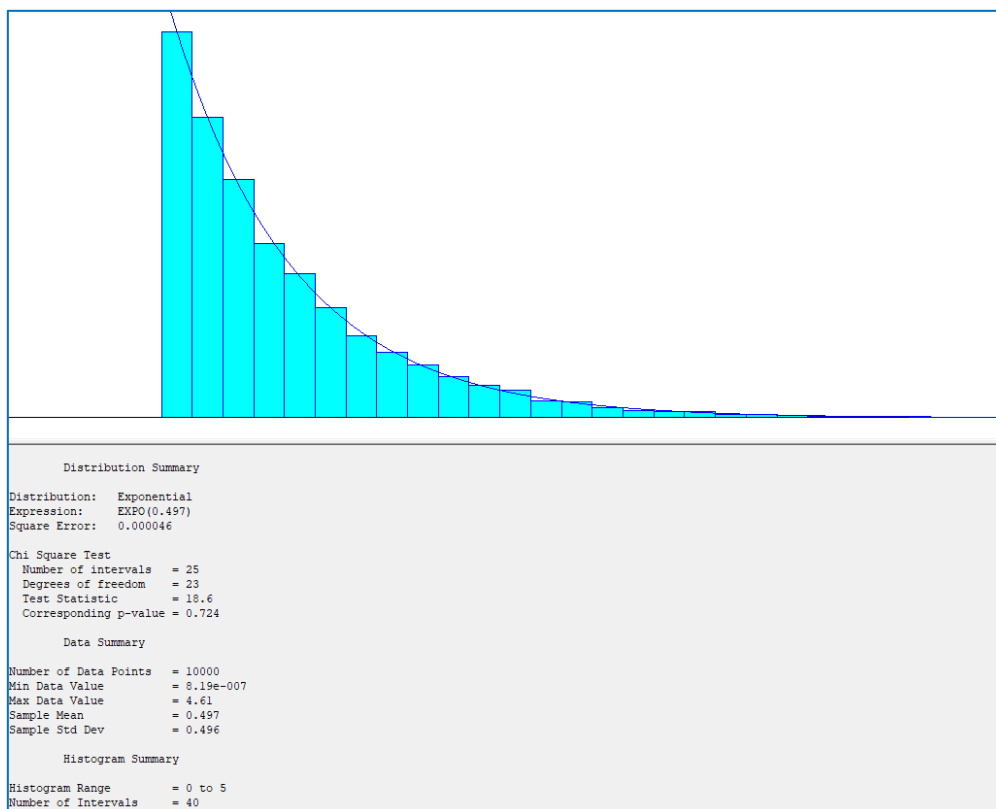
Tabuľka 1- výsledky experimentov simulácie s 95% intervalmi spoľahlivosti pre priemernú dobu vyhotovenia .....	13
---	----

# Validácia generátorov

V tejto časti si ukážeme spôsob overenia rozdelení pravdepodobností pre nami implementované generátory pre exponenciálne a trojuholníkové rozdelenie pravdepodobnosti.

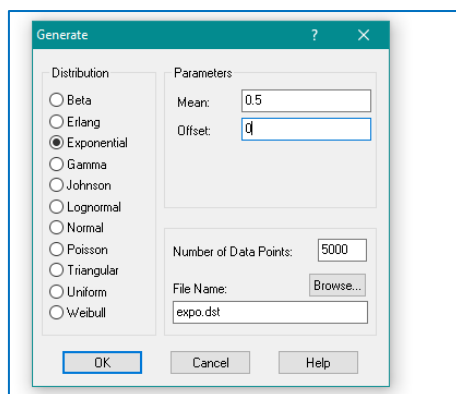
## Generátor pre exponenciálne rozdelenie pravdepodobnosti

Pre zvolenú hodnotu parametra  $\lambda = 2$  (priemerne 2 výskyty udalosti za jednotku času) sme si vygenerovali 10\_000 hodnôt, na ktoré, keď sme v programe Input analyzer aplikovali funkcionality *Fit all*, sme dostali výsledky zobrazené na obrázku 1.



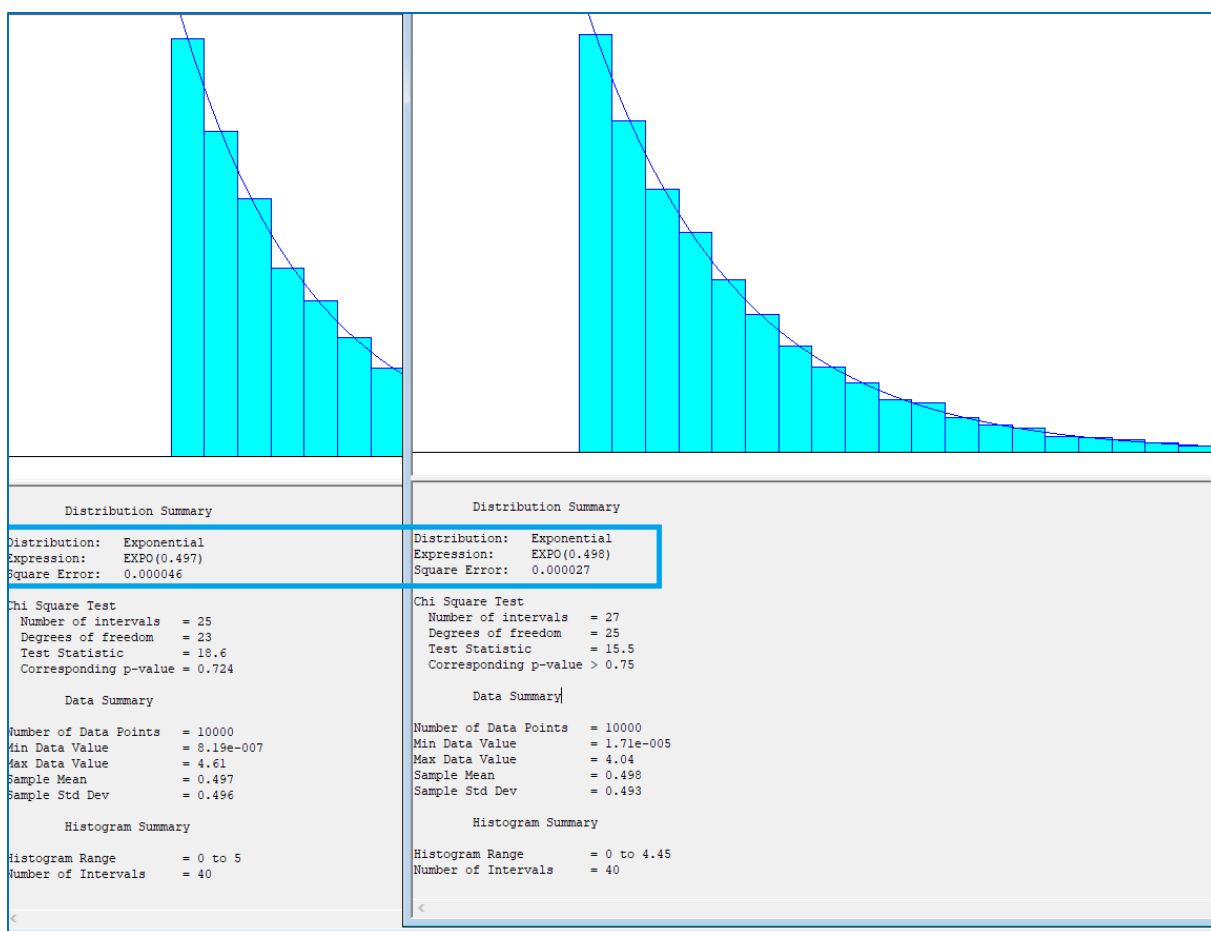
Obrázok 1 – Input analyzer - vyhodnotenie vzoriek generátora exponenciálneho rozdelenia

Dostali sme výsledok EXP(0,497), čo je exponenciálne rozdelenie pravdepodobnosti so strednou hodnotou počtu jednotiek medzi výskytmi 2 za sebou idúcich udalostí rovným približne 0,5, čo je prevrátená hodnota hodnoty  $\lambda$ . P-hodnota dosahuje hodnotu 0,724 a to je jednoznačne viac ako 0,05. Štvorcová chyba vyšla 0,000046. Pre overenie si vygenerujeme (obrázok 2) cez program Input analyzer 10\_000 vzoriek pre exponenciálne rozdelenie pravdepodobnosti s hodnotou parametra  $1/\lambda = 1/2 = 0,5$ .



Obrázok 2 - vygenerovanie vzoriek pre exponenciálne rozdelenie programom Input analyzer

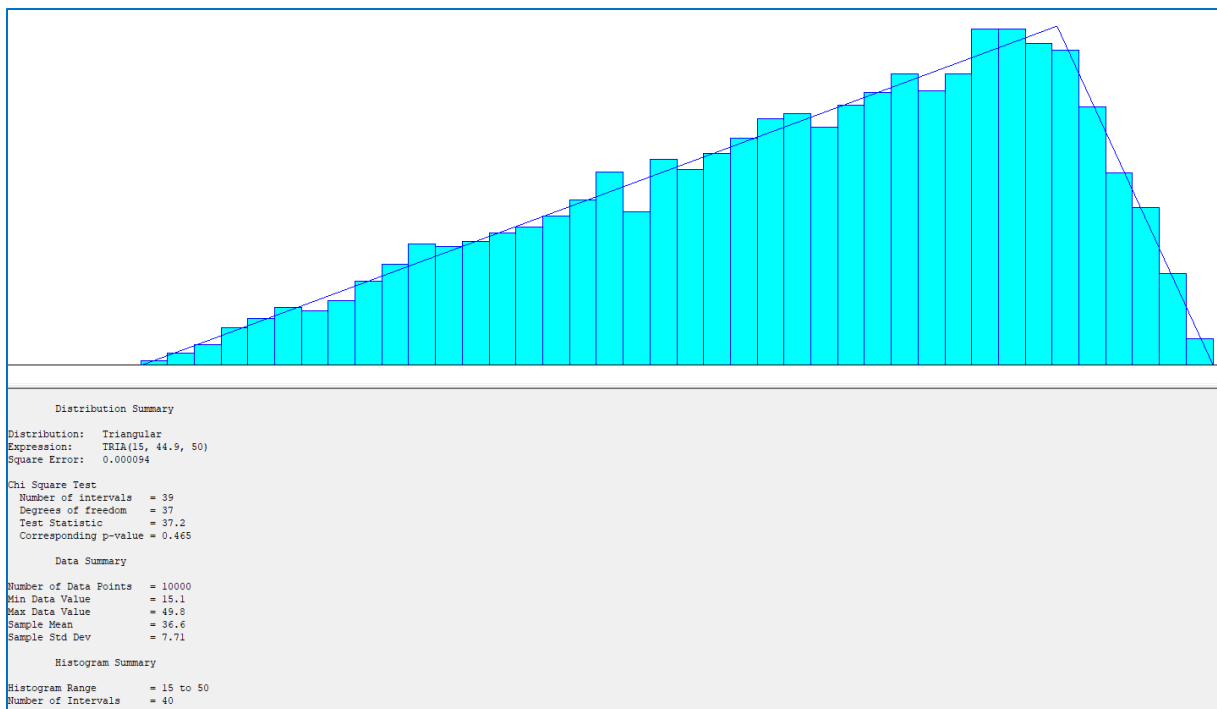
Podľa obrázku 3 môžeme vidieť, že stredná hodnota v oboch prípadoch vyšla rovnaká, a to 0,497. Okrem toho, rozdiel štvorcových chýb sa prejavil až na štvrtom desatinnom mieste.



Obrázok 3 - porovnanie výsledkov vzoriek pre exponenciálne rozdelenie pravdepodobnosti

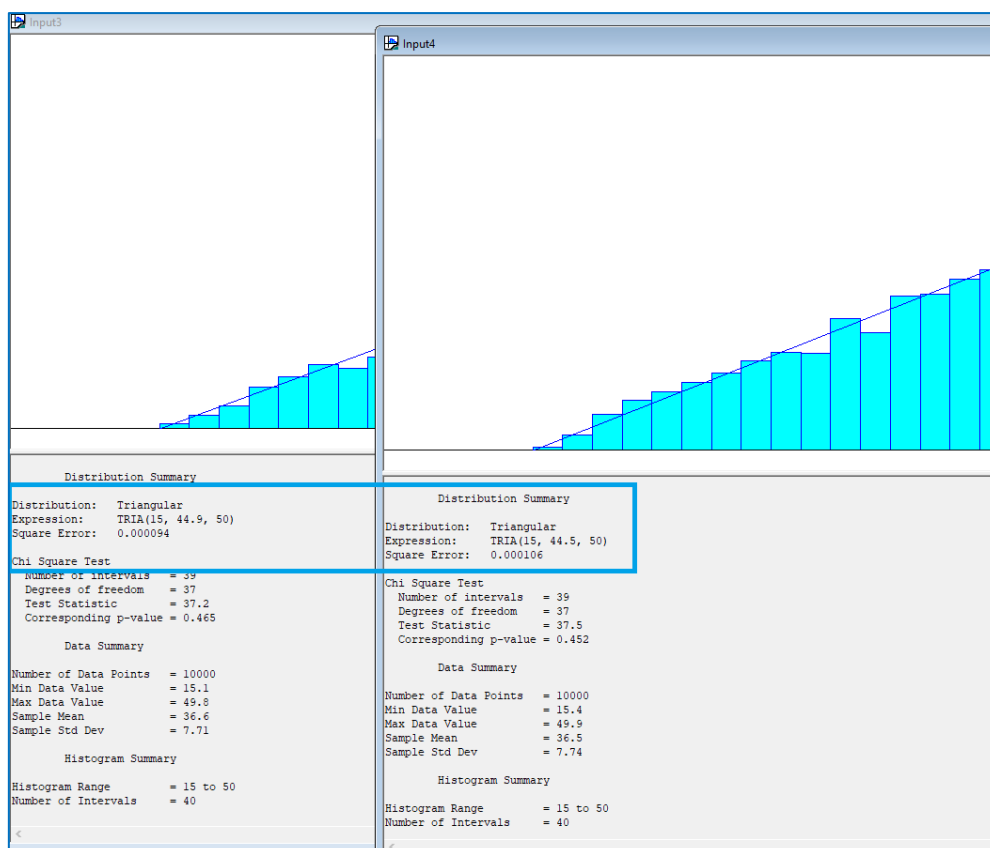
## Generátor pre trojuholníkové rozdelenie pravdepodobnosti

Podobne sme si vygenerovali 10\_000 vzoriek aj pomocou implementovaného generátora trojuholníkového rozdelenia pravdepodobnosti s parametrami:  $min=12;max=50;modus=45$ . Po uplatnení funkcionality *Fit all* programu Input analyzer sme dostali výsledky zobrazené na obrázku 4. Môžeme vidieť, že hodnoty zistených parametrov korešpondujú s parametrami, ktoré sme generátoru definovali. P-hodnota dosiahla hodnotu 0,465 a to je opäť značne viac ako 0,05.



Obrázok 4 - Input analyzer - vyhodnotenie vzoriek generátora trojuholníkového rozdelenia

Následne sme si pre spomenuté parametre vygenerovali 10\_000 vzoriek s trojuholníkovým rozdelením cez program Input analyzer.



Obrázok 5 - porovnanie výsledkov vzoriek pre trojuholníkové rozdelenie pravdepodobnosti

Podľa obrázku 5, ktorý zobrazuje výsledky porovnania, môžeme povedať, že hodnoty parametrov sa zhodujú s miernou odchýlkou v moduse. Rozdiel štvorcových chýb sa opäť prejavil až na štvrtom mieste za desatinnou čiarkou.

Na základe predchádzajúcich analýz môžeme generátory exponenciálneho aj trojuholníkového rozdelenia prehlásiť za správne naimplementované.



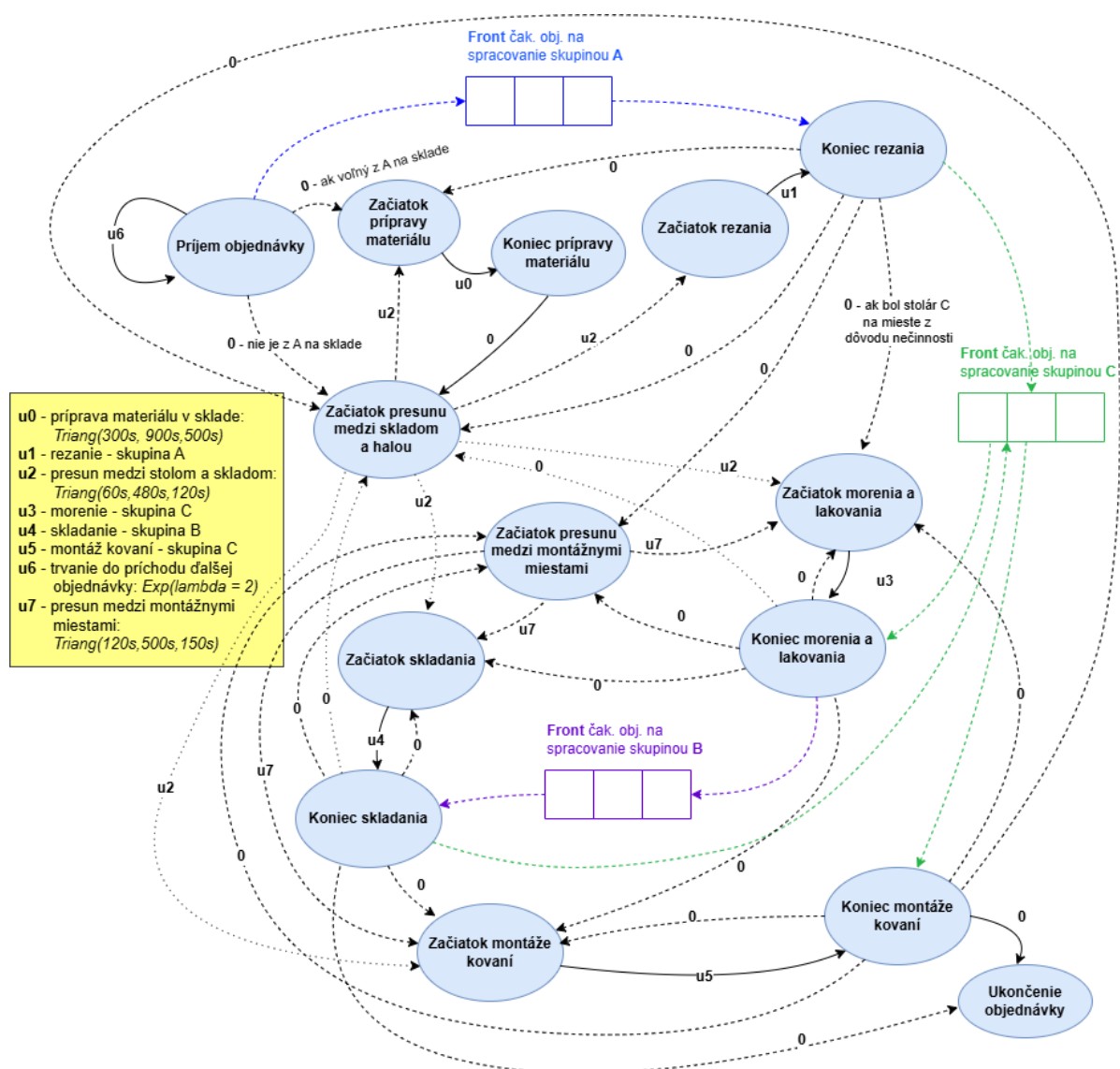
## Diagram udalostnej simulácie

Pre správnu implementáciu je nevyhnutným základom si dobre navrhnuť diagram udalostí, ktorého prvky poukazujú na vzťahy medzi vzájomným plánovaním udalostí. Na obrázku 6 je znázornený diagram udalostnej simulácie.

**Začiatok činnosti** (prípravy materiálu, rezania, morenia a lakovania, skladania, montáže kovaní) **môže naplánovať len svoj koniec činnosti**.

**Ukončenie činnosti** (rezania, morenia a lakovania, skladania, montáže kovaní) **môže naplánovať**:

- **začiatok svojej činnosti**
- **prechod medzi halou a sklodom** (pokiaľ ešte stolár nezačal žiadnu prácu alebo ide o stolára zo skupiny A, ktorý sa musí vždy vrátiť do skladu pri spracovaní novej objednávky)
- **prechod medzi montážnymi miestami**



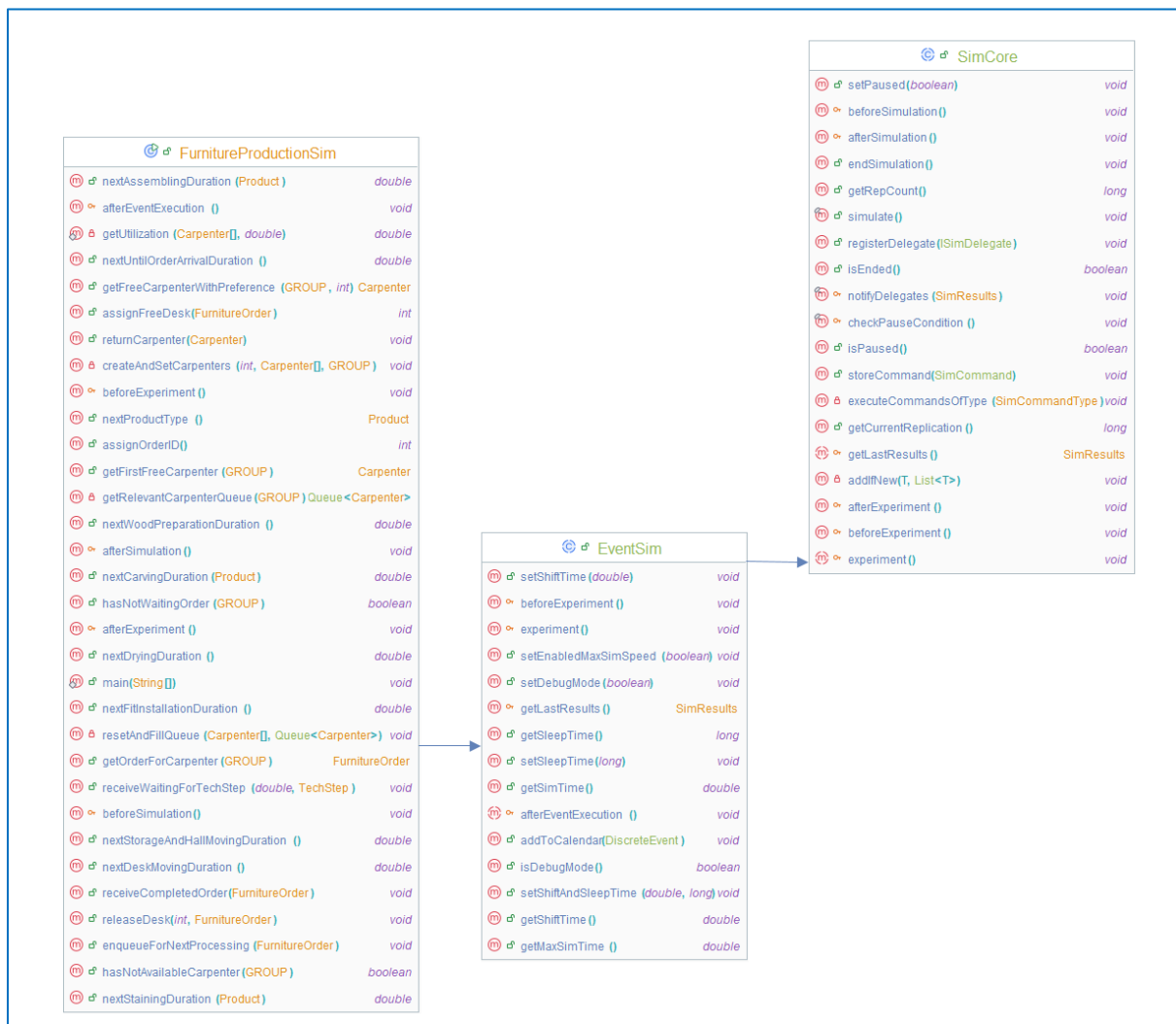
Obrázok 6 – diagram udalostnej simulácie výroby nábytku

# Architektúra riešenia

V tomto riešení sme použili MVC architektúru.

## Logika

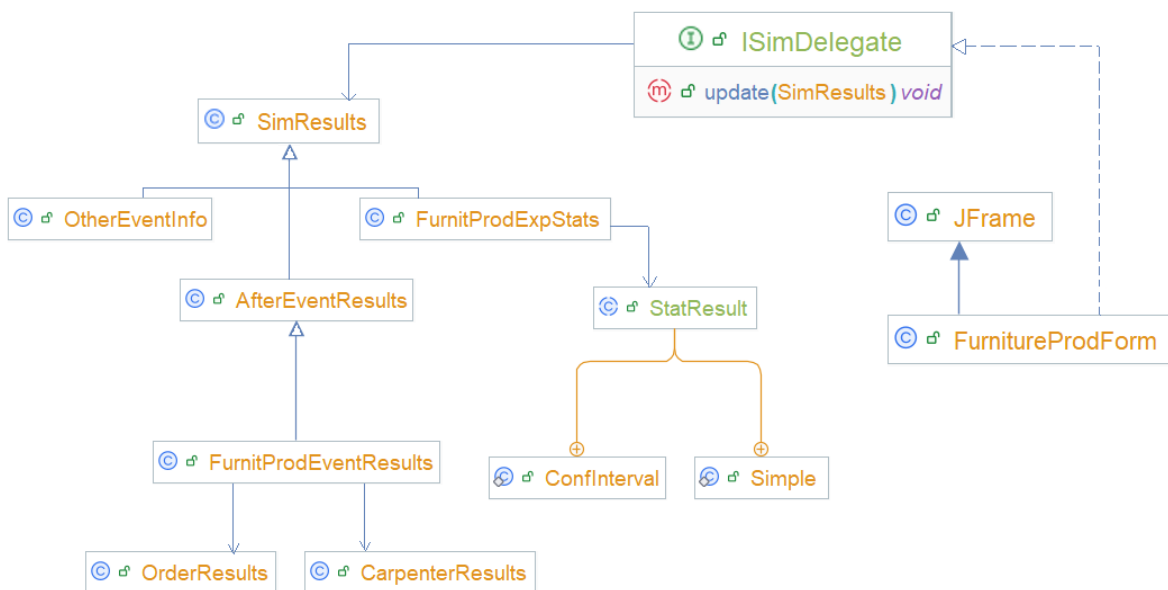
Vytvorili sme si potomka pre udalostnú simuláciu, ktorého sme následne realizovali pre vytvorenie simulácie výroby nábytku (trieda *FurnitureProductionSim* na obrázku 7).



Obrázok 7 - udalostná simulácia

Na obrázku 8 je znázornená spolupráca medzi kľúčovými triedami pre simuláciu výroby nábytku. Hlavnou triedou je samotná simulácia *FurnitureProdSim*, ktorá riadi vykonávanie udalostí (obrázok 9), ktoré sú potomkami triedy *FurnitureProdEvent*.

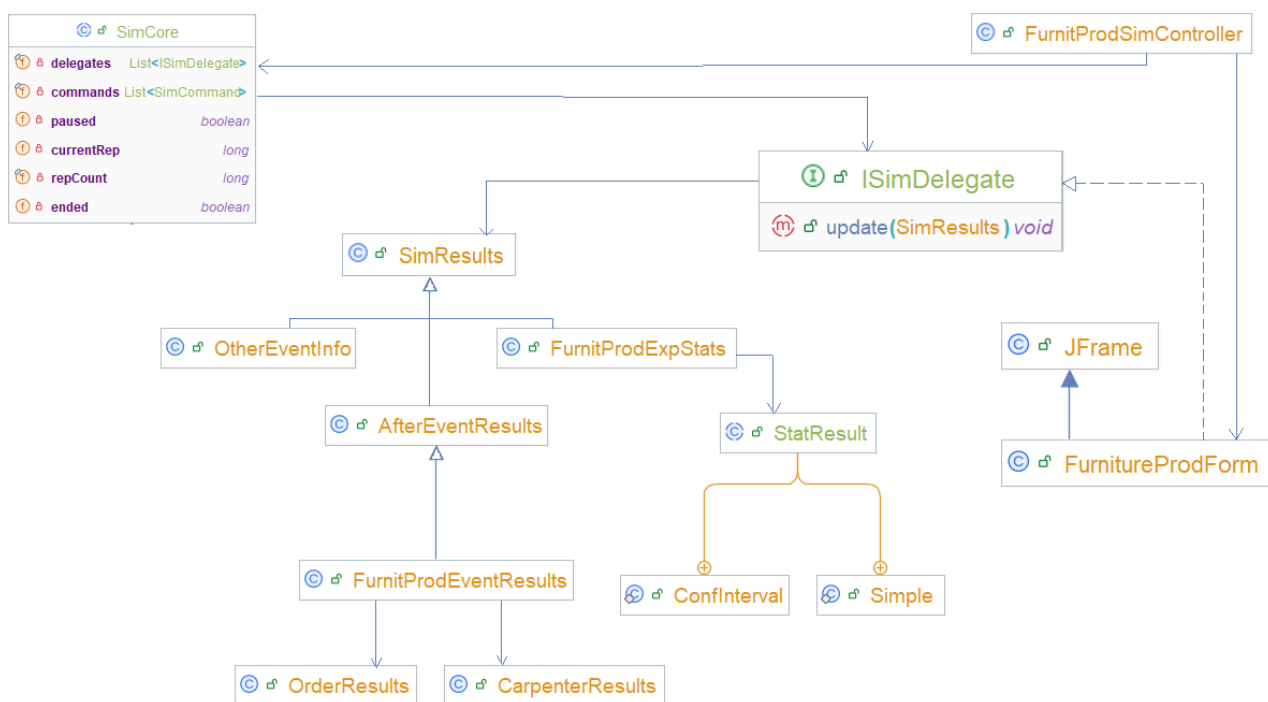




Obrázok 10 – simulačné výsledky

## Grafické užívateľské rozhranie

Trieda simulácie notifikuje GUI o zmenách prostredníctvom návrhového vzoru *Observer* (obrázok 11). Simulácia posielala do GUI výsledky – potomkov triedy *SimResults* - cez interface *ISimDelegate*, ktorý GUI implementuje. Komunikácia z GUI do simulácie prebieha cez kontrolér *FurnitProdSimController*.



Obrázok 11 - GUI s implementáciou aktualizácie simulačných výsledkov

## Experimenty

V tejto časti si ukážeme niekoľko experimentov, ktorými chceme optimalizovať priemernú dobu vyhotovenia objednávky. Zároveň musíme akceptovať kritérium, že objednávka musí byť priemerne vyhotovená do 16 hodín, t.j. dvoch pracovných dní. Budeme teda vyberať počty stolárov pre skupiny A, B a C, aby sme vedeli efektívne spracovávať objednávky.

Dôležité je si uvedomiť, že stolári skupiny C, ktorí vykonávajú morenie a lakovanie, potrebujú oveľa viac času na dokončenie svojej práce ako ostatní stolári. Z toho logicky vyplýva, že ich musí byť najviac, pretože inak sa im bude iba zvyšovať počet nespracovaných objednávok. Všetky nasledovné experimenty budeme simulovať na 10 000 replikáciach.

### Experiment A

Pre úvod, na základe našich predchádzajúcich úvah, skúsme použiť konfiguráciu **A=1;B=1;C=10** pracovníkov. Pri takejto konfigurácii sme dostali priemerný čas vyhotovenia objednávky až **400** hodín.

### Experiment B

Na základe výsledkov experiment A môžeme predpokladať, že budeme potrebovať celkovo viac stolárov. Vyskúšame konfiguráciu **A=2;B=3;C=13** pracovníkov. Priemerný čas spracovania objednávky sa nám znížil skoro na polovicu (približne **226** hodín) voči experimentu A.

### Experiment C

Stolárov v skupine C je pravdepodobne ešte stále málo. Preto skúsime konfiguráciu **A=2;B=2;C=18**. Výsledky sa nám omnoho zlepšili, nakoľko sme dostali priemernú dobu iba **13.23** hodiny.

### Experiment D

Skúsme ešte pridať do skupiny C dvoch stolárov a do skupiny B jedného, pretože priemerne dlhšie pracuje stolár B ako stolár A. Konfigurácia **A=2;B=3;C=20** nám vylepšila čas na **11.16** hodiny.

### Experiment E, F, G

Následne sme vyskúšali konfiguráciu s počtom stolárov **A=3;B=3** a **C** sme **menili**. Všetky výsledky sú v tabuľke 1.

ID experimentu	Stolári A	Stolári B	Stolári C	95% interval spoľahlivosti pre priemernú dobu vyhotovenia objednávky [hod]
A	1	1	10	< 408.35630   408.64945   408.94261 >
B	2	3	13	< 225.98378   226.31557   226.64736 >
C	2	2	18	< 13.21332   13.23164   13.24996 >
D	2	3	20	< 11.15844   11.16256   11.16669 >
E	3	3	20	< 10.61991   10.62332   10.62673 >
F	3	3	25	< 10.14167   10.14278   10.14390 >
G	3	3	18	< 12.36099   12.37885   12.39671 >

Tabuľka 1- výsledky experimentov simulácie s 95% intervalmi spoľahlivosti pre priemernú dobu vyhotovenia

## Vyhodnotenie experimentálnej časti

Na základe tabuľky 1 sa môžeme rozhodnúť rôzne. Záleží to od disponibility našich zdrojov. Veľmi dobré výsledky sme dosiahli pre experiment **E** a **F**.