

POLITECHNIKA POZNAŃSKA

Wydział Elektroniki i Telekomunikacji

**SYMULACJA CYFROWA
PROJEKT**

Poznań, 2019

Wykonała: Emilia Nowacka

Metoda: przeglądanie działań

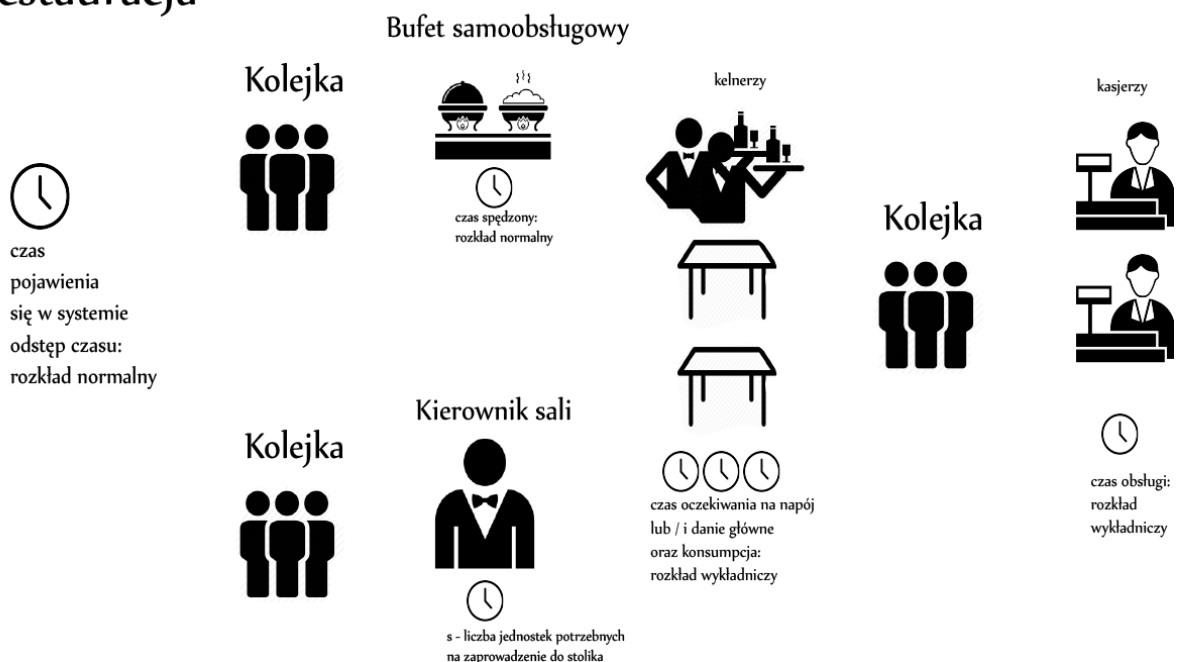
1. Treść zadania

W chińskiej restauracji pracuje k kelnerów obsługujących n_2 stolików dwuosobowych, n_3 stolików trzyosobowych oraz n_4 stolików czteroosobowych. Klienci pojawiają się w restauracji jako grupy 1-, 2-, 3- lub 4-osobowe z prawdopodobieństwami odpowiednio p_1, p_2, p_3 oraz p_4 . Odstęp czasu rozdzielający pojawienie się kolejnych grup klientów jest zmienną losową o rozkładzie normalnym ze średnią μ_a i wariancją σ_a^2 . Jeśli jest dostępny stół odpowiadający wielkości grupy (lub większy), klienci są do niego prowadzeni przez kierownika sali (czynność ta zajmuje s jednostek czasu). W przeciwnym przypadku grupa oczekuje na stół w kolejce. Średnio połowa klientów korzysta z samoobsługowego bufetu, przy którym może znajdować się jednocześnie b osób. Czas spędzany przy bufecie przez grupę klientów jest zmienną losową o rozkładzie normalnym ze średnią μ_b i wariancją σ_b^2 . Pozostali klienci są obsługiwani przez tego z kelnerów, który jako pierwszy będzie wolny. W pierwszej kolejności klienci otrzymują napoje, a następnie serwowane jest danie główne. Czas obsługi w obu przypadkach jest zmienną losową o rozkładzie wykładniczym ze średnimi odpowiednio λ_n oraz λ_j (te dwie wielkości uwzględniają zarówno czas oczekiwania na zrealizowanie zamówienia jak i sam czas podania napojów i posiłku głównego). Po zakończeniu konsumpcji, której długość jest zmienną losową o rozkładzie wykładniczym ze średnią λ_f , klient płaci jednemu z c zatrudnionych kasjerów. Czas obsługi przez kasjera jest zmienną losową o rozkładzie wykładniczym ze średnią λ_p . Zakładając, że kierownik sali zawsze wybiera stół najlepiej pasujący do danej grupy oraz stosuje jedną z podanych niżej zasad obsługi kolejki, oszacuj za pomocą odpowiedniego eksperymentu symulacyjnego:

- średni czas oczekiwania na stół,
- średnią długość kolejki oczekujących na stół,
- średni czas oczekiwania na obsługę przez kelnera od momentu zajęcia miejsca przy stoliku,
- średnią długość kolejki przy kasach.

2. Schemat modelu symulacyjnego

Restauracja



3. Opis klas wchodzących w skład systemu i ich atrybuty

Klasa: Restaurant

Opis: Klasa zrzeszająca wszystkie podstawowe obiekty potrzebne do symulacji.

Atrybuty:

- kolejka do kierownika sali
- kolejka do bufetu
- kolejka do kasjerów
- tablica obiektów kelner
- tablica obiektów kasjer
- tablica obiektów stoliki
- obiekt bufet
- zmienna przechowująca wygenerowany czas pojawienia się kolejnego klienta
- tablica czasów kolejnych zdarzeń
- zmienna przechowująca czas symulacji

Klasa: Group

Opis: Klasa zawierająca konstruktor grupy klientów

Atrybuty:

- identyfikator grupy
- wielkość grupy
- miejsce posiłku (bufet/stolik)
- czas konsumpcji
- zmienna oznaczająca etap obsługi
- czas pojawienia się grupy w systemie

Klasa: Buffet

Opis: Klasa zawierająca konstruktor bufetu oraz metody potrzebne do obsługi przy bufecie.

Atrybuty:

- zmienna określająca ilość miejsc przy bufecie
- tablica przechowująca grupy aktualnie korzystające z bufetu
- zmienna określająca ilość aktualnie wolnych miejsc przy bufecie
- metoda wykonująca obsługę klienta (dodanie do bufetu, generacja czasu konsumpcji, dodanie do listy zdarzeń końca konsumpcji)

Klasa: Table

Opis: Klasa zawierająca konstruktor stolika oraz metody potrzebne do obsługi przy stoliku.

Atrybuty:

- ilość miejsc przy stoliku
- identyfikator stolika
- zmienna przechowująca aktualną grupę, siedzącą przy danym stoliku
- metoda rozpoczęcia obsługi przy stoliku (jedzenie oraz picie)

Klasa: Headwaiter

Opis: Klasa zawierająca konstruktor kierownika sali oraz metody potrzebne do obsługi klientów przez kierownika.

Atrybuty:

- zmienna przechowująca aktualnie obsługiwaną grupę
- zmienna określająca czas końca obsługi
- zmienna określająca id aktualnie obsługiwanej grupy
- metoda obsługi klienta (dobór stolika oraz generacja czasu końca obsługi)

Klasa: Waiter

Opis: Klasa zawierająca konstruktor kelnera oraz metody potrzebne do obsługi klientów przez kelnera.

Atrybuty:

- identyfikator kelnera
- zmienna przechowująca czas końca obsługi (napój)
- zmienna przechowująca czas końca obsługi (posiłek)
- zmienna określająca, czy kelner jest zajęty czy wolny
- metoda obsługi - podanie napoju (generacja czasu końca obsługi, dodanie go do listy zdarzeń)
- metoda obsługi - podanie posiłku (generacja czasu końca obsługi, generacja czasu końca posiłku dla grupy, dodanie ich do listy zdarzeń)

Klasa: Cashier

Opis: Klasa zawierająca konstruktor kasjera oraz metody potrzebne do obsługi klientów przez kasjera.

Atrybuty:

- identyfikator kasjera
- zmienna przechowująca aktualnie obsługiwaną grupę
- zmienna przechowująca czas końca obsługi
- metoda obsługi klienta (generacja czasu końca obsługi, dodanie go do listy zdarzeń)

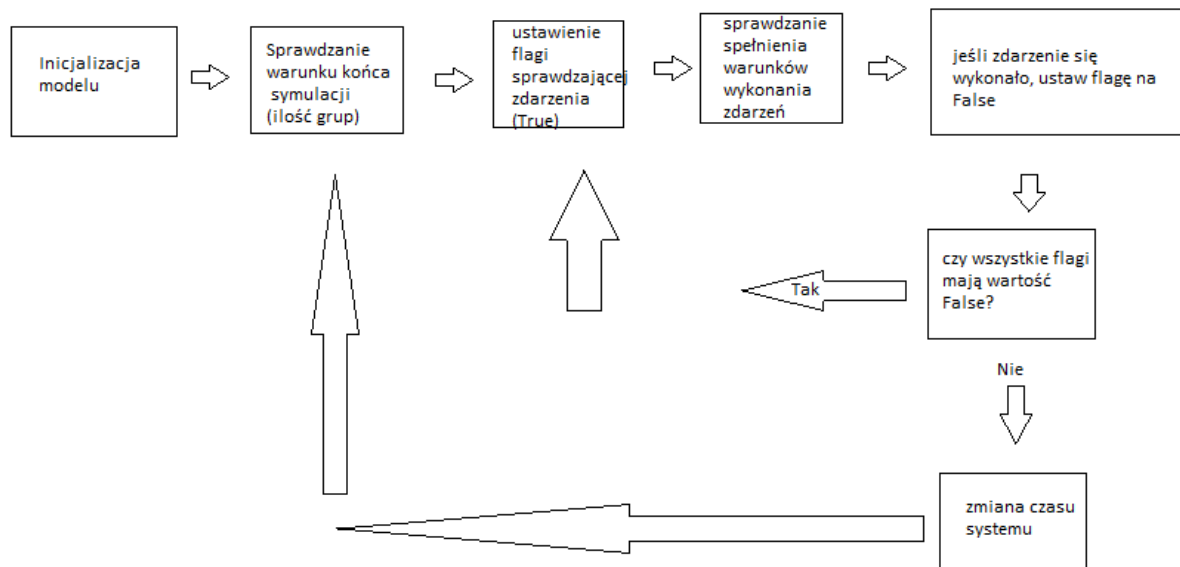
Klasa: Stats

Opis: Klasa wykorzystywana do zbierania danych statystycznych

Atrybuty:

- licznik grup w systemie
- tablica przechowująca wielkość licznika
- tablica przechowująca chwile czasowe, w których pojawiają się klienci lub zostają usunięci
- zmienna przechowująca średnią czasu oczekiwania na stolik
- zmienna przechowująca liczbę grup czekających na stolik
- metoda licząca fazę początkową
- metoda licząca średnią czasu oczekiwania na stolik
- metoda licząca średnią czasu oczekiwania na kelnera
- metoda licząca średnią długość kolejki do stolika
- metoda licząca średnią długość kolejki do kas

4. Schemat blokowy pętli głównej



5. Lista zdarzeń czasowych i warunkowych

Warunkowe:

- początek obsługi przez kierownika sali;
- początek posiłku (bufet);
- początek posiłku (stolik);
- początek obsługi przez kelnera;
- początek obsługi przez kasjera.

Czasowe:

- pojawienie się grupy klientów w systemie;
- zakończenie obsługi przez kierownika sali;
- zakończenie posiłku (bufet);
- zakończenie podawania napoju;
- zakończenie podawania posiłku (zwolnienie kelnera i zaplanowanie końca konsumpcji);
- zakończenie posiłku (stolik);

- pojawienie się grupy w kolejce do kas;
- zakończenie obsługi przez kasjera;

6. Parametry wywołania programu

Liczba grup: 200

Faza początkowa określona przez ilość grup znajdujących się aktualnie w systemie.

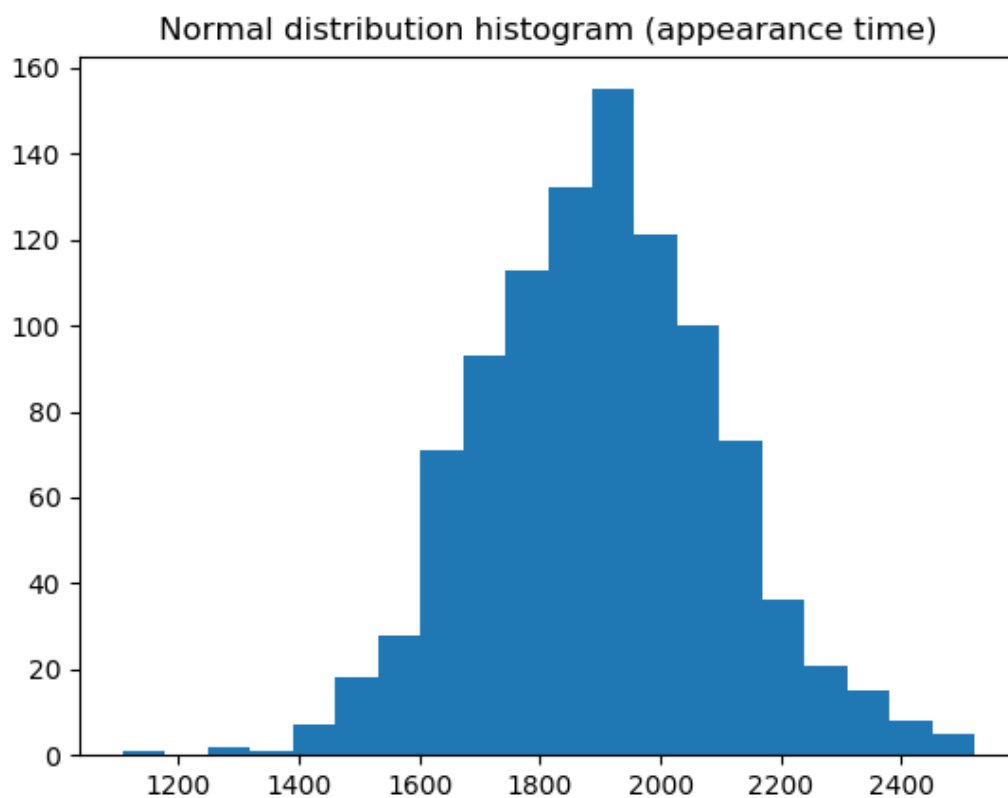
Prawdopodobieństwa: $P(1) = 0.11$, $P(2) = 0.33$, $P(0.33)$, $P(0.23)$

Czas pojawienia się kolejnego klienta: 2000(200)

Wszystkie inne dane podane dla D1 zostały takie same.

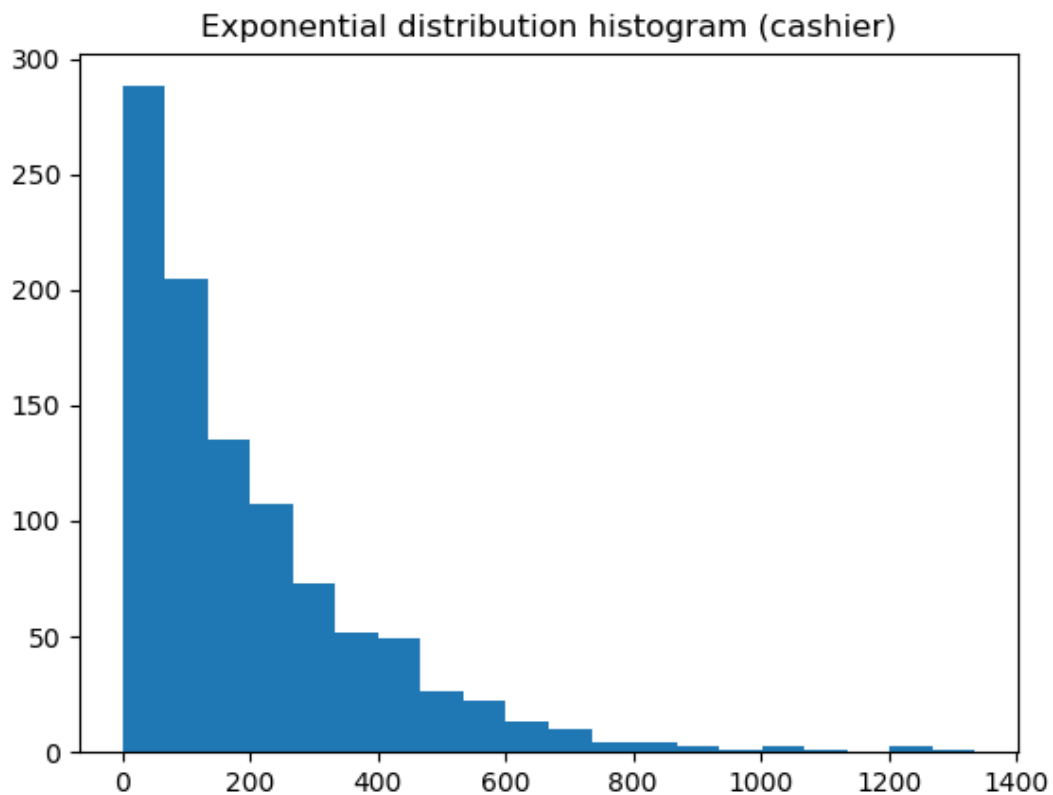
7. Generatory

Generator z rozkładem normalnym:



Pobiera próbki z rozkładu normalnego (Gaussowskiego).

Generator z rozkładem wykładniczym:



Pobiera próbki z rozkładu wykładniczego.

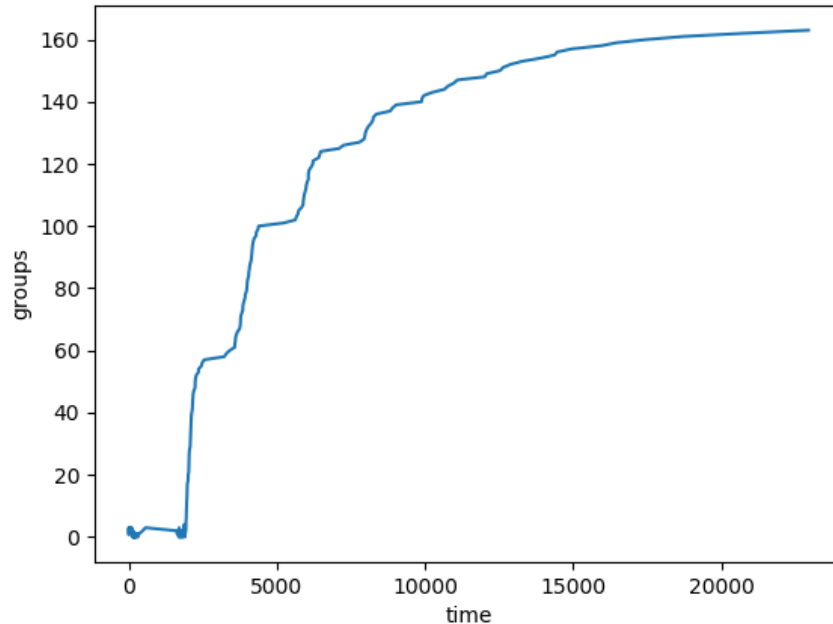
Niezależność sekwencji losowych została zachowana dzięki zmiennej ilości grup. W ten sposób sekwencje losowe nie były ze sobą powiązane. Ponadto, do każdego czasu został wykorzystany oddzielny generator, nawet jeśli dane zmienne miały być generowane przez generator o takim samym typie rozkładu. Przewidziano przeprowadzenie 10 symulacji, każda symulacja zawierała oddzielny proces inicjalizacji systemu, przez co zachowano niezależność sekwencji losowych.

8. Zastosowana metoda testowania i weryfikacji poprawności działania programu

Symulacja była testowana po każdej modyfikacji, sprawdzając krokowo czy wszystkie procesy, obiekty oraz zdarzenia są poprawnie wykonywane. Brak komunikatu o którymś zdarzeniu, oznaczał błąd w symulacji. Komunikaty wykonania zdarzenia były weryfikowane, na podstawie danych liczbowych, np. czasu, numeru grupy, itp, w celu zachowania logiki całej symulacji, a także szybkiego wykrycia błędów w programie.

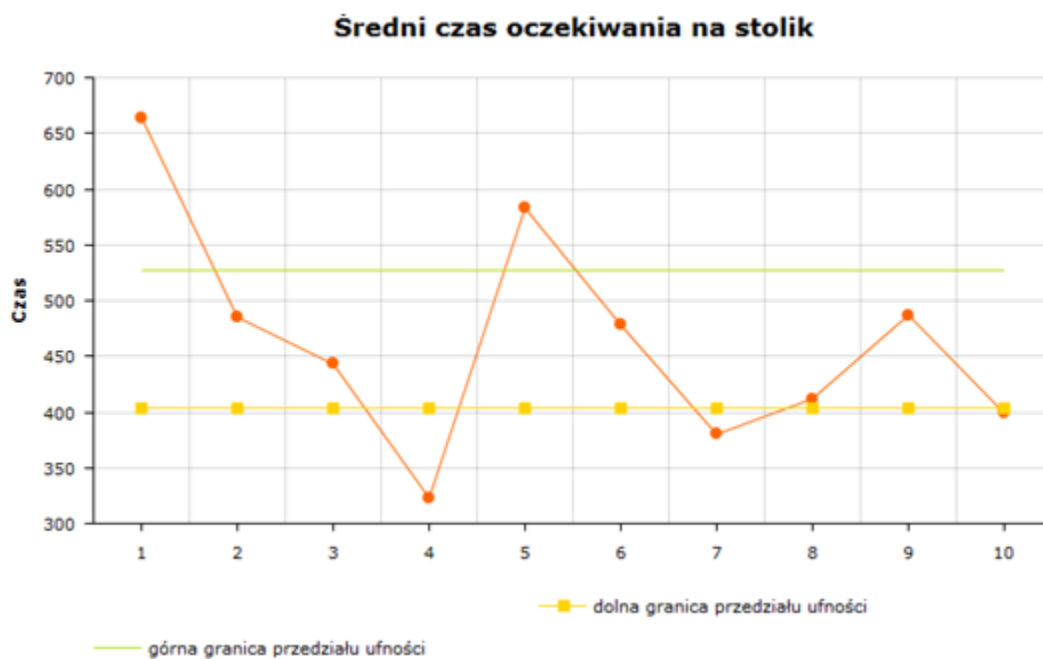
9. Wyniki symulacji

Długość fazy początkowej

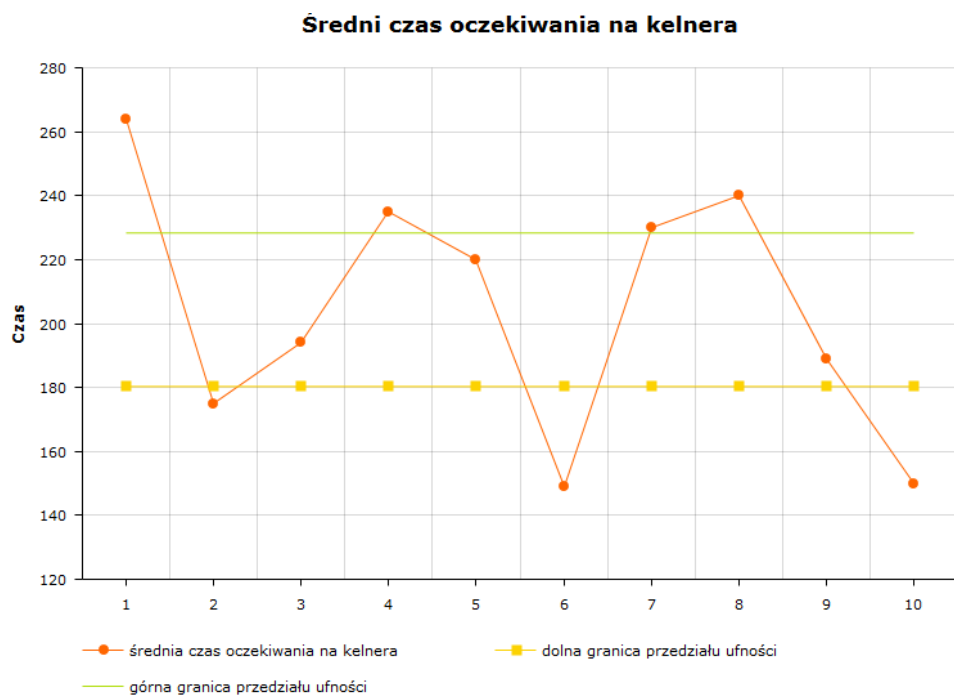


Faza początkowa wyznaczana była na podstawie ilości grup znajdujących się w systemie. Za każdym razem, gdy nowa grupa pojawiła się w systemie, licznik grup został zwiększany, a jego wartość zapisywana w tablicy przechowującej wielkość licznika. Dodatkowo w innej tablicy zapisywany był czas, w którym była pobierana wartość licznika. W ten sposób można było wyznaczyć wykres, przedstawiający ilość grup w systemie, podczas całego czasu pętli symulacyjnej. Jak widać na wykresie, na samym początku wartość ilości grup oscyluje wokół zera, następnie gwałtownie rośnie, by w okolicy 10000 jednostek czasu, uzyskać wartość, która później utrzymuje się do końca symulacji. Faza początkowa została wyznaczona dla bardzo dużej ilości grup klientów - 200.

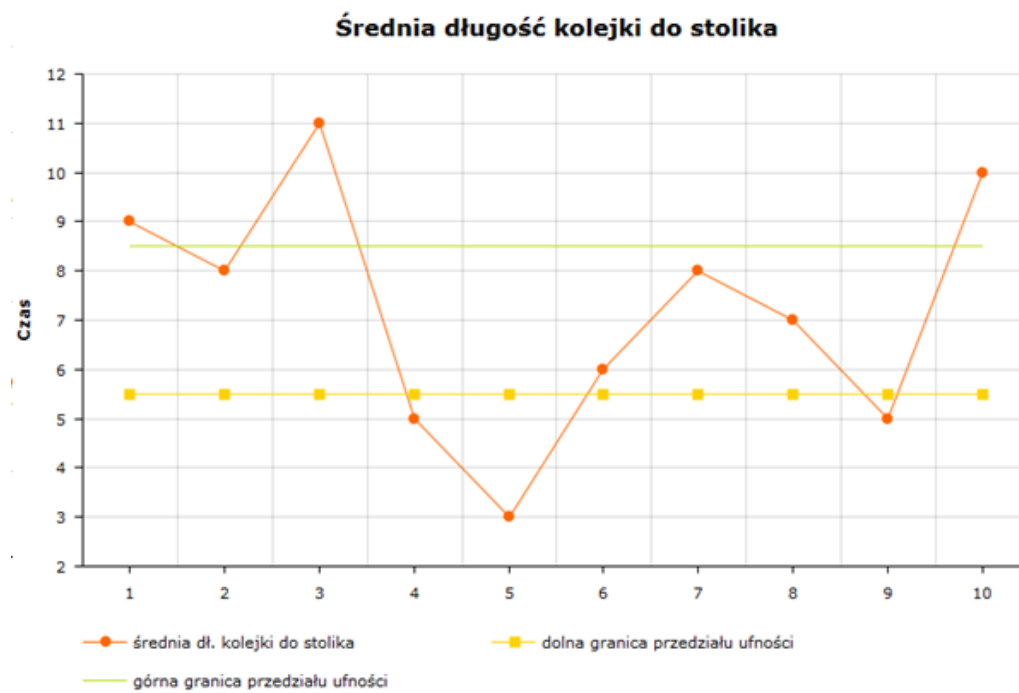
Średni czas oczekiwania na stół:



Średni czas oczekiwania na kelnera:



Średnia długość kolejki do stolika:



Średnia długość kolejki do kas:

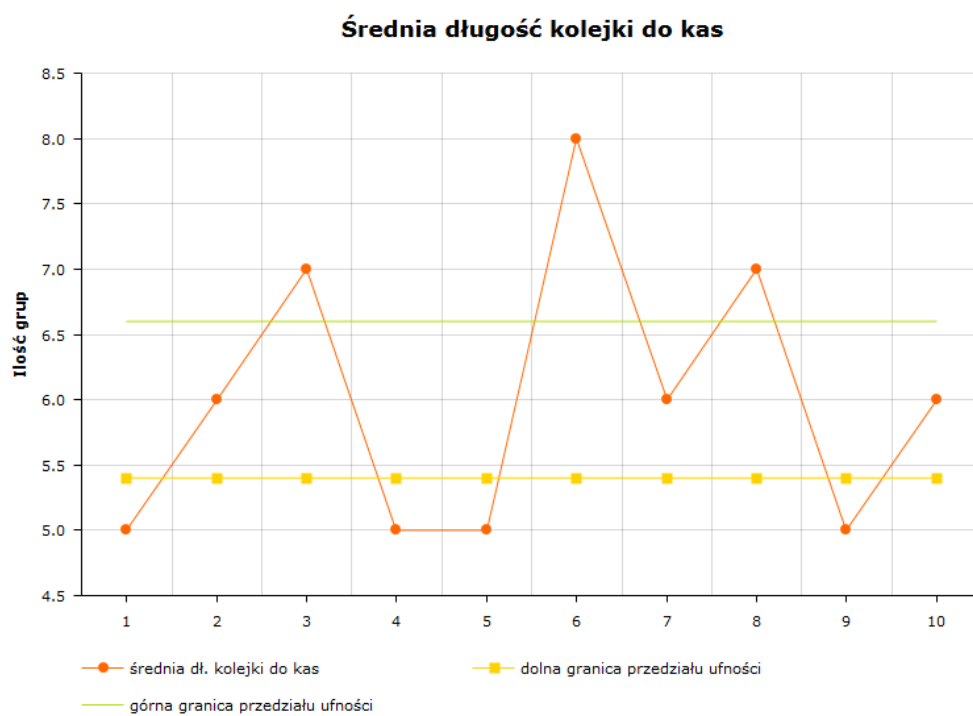


Tabela wyników:

Numer przebiegu symulacji	Średni czas oczekiwania na stół	Średnia długość kolejki do stolika	Średnia długość kolejki do kas	Średni czas oczekiwania na kelnera
1	664	9	5	261
2	485	8	6	175
3	443	11	7	194
4	323	5	5	235
5	583	3	5	220
6	478	6	8	149
7	380	8	6	230
8	412	7	7	240
9	487	5	6	189
10	399	10	5	150

Uśrednione wyniki ze wszystkich przebiegów:

Średni czas oczekiwania na stół: 465.4

Średnia długość kolejki do stolika: 7

Średnia długość kolejki do kas: 6

Średni czas oczekiwania na kelnera: 204.3

Przedziały ufności:

Średni czas oczekiwania na stół: od 403.7 do 527.2

Średnia długość kolejki do stolika: od 5.5 do 8.5

Średnia długość kolejki do kas: od 5.4 do 6.6

Średni czas oczekiwania na kelnera: od 180.3 do 228.3

10. Wnioski

Biorąc pod uwagę temat symulacji, nasuwa się wniosek, iż można przeprowadzić wirtualną symulację właściwie każdego realnego procesu/projektu/maszyny, który możemy stworzyć w rzeczywistym świecie. Bardzo przydatną umiejętnością jest programowanie symulacji cyfrowej. Dzięki niej możemy z pewnym przybliżeniem określić efekty, których można spodziewać się po fizycznej realizacji danego projektu.

Z mojej symulacji wynika, iż restauracja o zadanych parametrach, czyli m. in. ilości stolików oraz obsługi, dobrze poradzi sobie z obsługą klientów, nawet jeśli będzie ich wielu. Czasy

oczekiwania oraz kolejki są stosunkowo krótkie, dlatego można spodziewać się dużych obrotów w restauracji. Stan symulacji ustabilizował się, po pewnym, określonym czasie, co oznacza, że proces przebiegł zgodnie z planem.

Cały proces tworzenia projektu symulacji restauracji wiele mnie nauczył. Fakt wybrania języka Python jako technologii do tego programu uważam za bardzo dobrą decyzję. Wpłynęła ona na pewno pozytywnie na budowę kodu oraz jego skomplikowanie.