**DOKUMENTACJA SYMULACJI STADIONU W DNIU KONCERTU**

**Spis treści**

1. Plik rozkład.py
   1. opis
   2. Zmienne
   3. Funkcje
2. Plik klasy.py
   1. Opis
   2. Klasy
      1. Funkcje klas
3. Plik MainPetlaB.py
   1. Opis
   2. Zmienne modelowe
   3. Inne zmienne
   4. Funkcje
   5. Opis GUI tkinter
4. **Plik rozkład.py**
5. Opis

Plik rozkład.py zawiera kilka funkcji, których celem jest wygenerowanie zmiennych losowych z rozkładu Weibulla w określonych ograniczeniach oraz wizualizacji rozkładu i estymatora gęstości dla wylosowanych zmiennych.

1. Zmienne

* Parametry Weibulla:
  + lambda\_vals – zmienna określająca jeden z parametrów rozkładu Weibulla
  + k\_vals – zmienna określająca jeden z parametrów rozkładu Weibulla
* Inne zmienne:
  + T\_max – czas całej symulacji (3 \*3600 sekund = 3 godziny)
  + Max\_ludzi – zmienna określająca maksymalną ilość możliwych do wygenerowania zmiennych losowych przy używaniu metody monte carlo.
  + Ludzie – Lista zmiennych losowych, czyli przychodzących ludzi zgodnie rozkładem Weibulla w T\_max z ograniczeniem max\_ludzi

1. Funkcje

Plik rozkład.py zawiera następujące funkcje:

* weibull\_pdf - funkcja ta zwraca wartość matematyczną z algebraicznej formy rozkładu weibulla.
* ostat – funkcja wizualizująca rozkład Weibulla z wybranymi wcześniej parametrami
* generowanie\_ludzi – funkcja generująca zmienne losowe zgodnie z metodą monte carlo. Zmienne losowe sa przekazywane do listy ‘ludzie’
* gestosc\_ludzi - funkcja wizualizująca estymator gęstości dla wylosowanych wcześniej zmiennych ‘ludzie’ przy pomocy funkcji ‘generowanie\_ludzi’
* przyjścia\_w\_czasie – funkcja korygująca rozkład zmiennych losowych tak aby oś x była osią czasu o wcześniej ustalonych parametrach.
* Wizualizacja\_w\_czasie – funkcja wizualizująca rozkład zmiennych w czasie.

1. **Plik klasy.py**
2. Opis

Plik klasy.py zawiera deklaracje paru niezbędnych do ostatecznej symulacji klas, takich jak bramka czy czlowiek, każda z tych klas ma również swoje funkcje.

1. Klasy

* Rockman – klasa przedstawiająca człowieka przychodzącego na koncert (do kolejki), instacje tej klasy posiadaja pewne paramatry związane z badaniem wyniku i przebiegu symulacji, oraz pare funkcji umożliwiających logiczne powiązanie instancji Rockman z bramką.
  1. Parametry klasy Rockman:
     + - Czy\_pleak – Boolowska wartość informująca czy instancja klasy Rockman posiada plecak
       - T\_w\_kolejce – Integer informujący o ilości czasu spędzonym w kolejce dla konkretnej instancji Rockman
       - Bramka – zmienna pozwalająca przypisać bramkę do której Rockman ma się udać
       - Imie – zmienna informująca o numerze instancji Rockmana(albo imieniu)
  2. Funkcje klasy Rockman:
     + - \_\_str\_\_ - funkcja wywolywana przy próbie wywołania instancji Rockmana, zwraca informacje o czy\_plecak, t\_w\_kolejce
       - Set\_bramka – funkcja pozwalająca przypisać instancje bramki do instancji Rockmana
       - Usun\_sie\_z\_bramki – funkcja wywolujaca funkcje klasy bramka ‘Usun\_osobe\_z\_bramki’
       - Get\_czy\_plecak – funkcja zwracająca wartość czy\_plecak
       - Set\_czy\_plecak – funkcja ustawiająca wartość czy\_plecak na True/False (w zaleznosci która wpiszemy), po wcześniejszym sprawdzeniu czy wpisaliśmy True/False/1/0
       - Get\_t\_w\_kolejce – funkcja zwracająca wartość t\_w\_kolejce
       - Dodaj\_t\_w\_kolejce – funkcja zwiększająca t\_w\_kolejce o podaną wartość, przy braku podania konkretnej wartości, do t\_w\_kolejce dodawane jest 1
* Bramka – klasa reprezentująca bramke sprawdzająca bilet, dowod itp. Przed wejściem na obiekt stadionu. Bramka jest obiektem który klasa rockman spotyka po przejściu kolejki. Instancje tej klasy posiadają informacje o tym ile już jest sprawdzany rockman na bramce, przypisaną kolejke do bramki, numer bramki oraz pare funkcji umożliwiających przebieg sprawdzania i przypisywania.
  1. Parametry klasy bramka:
     + - Czy\_bramka\_otwarta – boolowska wartość informujaca czy bramka jest otwarta czy zamknieta
       - Czas\_sprawdzania – integer zawierający informacje ile jest już sprawdzany przypisany wcześniej obiekt rockman
       - Max\_czas\_spradzania – integer zawierający informacje o dlugosci sprawdzania na bramce instancji klasy Rockman
       - Osoba\_sprawdzana – zmienna zawierająca informacje o przypisanej do bramki instancji klasy Rockman
       - Kolejka\_bramki – zmienna zawierająca informacje o przypisanej do tej instancji Bramki instancji Kolejka (przy deklarowaniu instancji wymagane jest podanie kolejki przypisanej do tej bramki)
       - Imie – zmienna zawierająca informacje o numerze bramki
  2. Funkcje klasy bramka:
     + - Get\_czy\_bramka\_otwarta – funkcja zwracająca informacje o czy\_bramka\_otwarta
       - Set\_czy\_bramka\_otwarta – funkcja ustawiająca wartość czy\_bramka\_otwarta na True/False (w zaleznosci która wpiszemy), po wcześniejszym sprawdzeniu czy wpisaliśmy True/False/1/0
       - Get\_czas\_sprawdzania - funkcja zwracająca informacje o czas\_sprawdzania
       - Set\_czas\_sprawdzania – funkcja ustawiająca czas\_sprawdzania na podaną wartość, przy braku podania konkretnej wartości, czas sprawdzania ustawiany jest na 0
       - Dodaj\_czas\_sprawdzania - funkcja zwiększająca czas\_sprawdzania o podaną wartość, przy braku podania konkretnej wartości, do czas\_sprawdzania dodawane jest 1
       - Przypisz\_osobe\_do\_bramki – funkcja pozwalająca przypisać instancje klasy Rockman do tej instancji klasy Bramka.
       - Usun\_osobe\_z\_bramki – funkcja pozwalająca na zerwanie przypisania instancji klasy Rockman do tej instancji klasy Bramka
       - Get\_przypisana\_osoba – funkcja pozwalająca zwrócić zawartośc osoba\_sprawdzana
       - Set\_kolejka – funkcja pozwalająca przypisać kolejke do bramki
* Bramka\_plecak – dziecko klasy bramka, zwierające jedynie zmienione parametry o maksymalnym czasie sprawdzania
* Kolejka – klasa kolejki na zasadzie FIFO, kolejka zawiera pare podstawowych mechanizmów do obsługi kolejkowiczów.
  1. Parametry klasy Kolejka
     + - Kolejka – lista zawierająca informacje o obiektach w kolejce
       - Bramka\_kolejki – zmienna zawierająca przypisaną intancje bramki do kolejki
       - Imie – zmienna zawierająca numer/imie kolejki
  2. Funkcje klasy Kolejka
     + - Dodaj\_element – funkcja dodająca element do listy Kolejka
       - Get\_element – funkcja zwracająca informacje o elemencie z kolejki przy podanym numerze
       - Usun\_element – funkcja usuwająca element z listy Kolejka, gdy Kolejka jest pusta, zwracany jest komunikat o ‘pustości’ kolejki
       - Czy\_pusta – funkcja zwracająca informacje czy długość listy kolejka jest równa 0
       - Rozmar – funkcja zwracająca informacje o długości listy Kolejka
       - Usun\_rockmana – funkcja usuwająca pierwszy element (klasy Rockman) listy Kolejka
       - Set\_kolejka – funkcja przypisująca bramke do kolejki
       - Get\_pierwszy – funkcja zwracająca pierwszy element listy Kolejka

1. **Plik MainPetlaB.py**
   1. Opis

MainPetlaB.py zawiera faktyczny proces symulacji dany w temacie tego sprawozdania. Cały proces podzielony jest na funkcje, aby nie robić tak zwanego „spaghetti code’u”, funkcje te są podzielone na dwa typy: „symulacyjne” oraz „testowe”, będące częścią weryfikacyjną procesu tworzenia symulacji komputerowej. Opis zmiennych podzieliłem też na zmienne wejścia, oraz inne zmienne, by podkreślić te zmienne które należą do wejść, którymi możemy manipulować, a nie zakłóceń na które nie mamy wpływu.

* 1. Zmienne wejścia
* Max\_ludzi – zmienna zawierająca informacje o liczbie ludzi, którzy „zakupili bilet na koncert”, czyli liczba ludzi, którzy chcą wejść tego dnia na obiekt koncertu.
* Ile\_bramek – zmienna zawierająca informacje o liczbie bramek, które zamierzamy otworzyć (a co za tym idzie kolejek), ta liczba jest jednak sumarczyną ilością bramek, oznacza to, że jest to liczba bramek nie-plecakowych + liczbą bramek plecakowych, więc zmienna Ile\_bramek musi być liczbą parzystą.
* T\_delta – zmienna zwierająca informacje o czasie rozgrzewania modelu, czyli jest to czas w którym symulacja zachowuje się normalnie poza faktem iż bramki sprawdzające są zamknięte.
  1. Inne zmienne
* T\_max – zmienna zawierająca informacje o czasie trwania symulacji, pobierana ona jest z wcześniejszego pliku „rozkład.py”
* T\_start\_koncertu – zmienna zawierająca informacje o czasie, w którym zaczyna się koncert.
* T\_dodatkowy\_czas – zmienna zawierająca informacje o czasie, dodatkowym trwania symulacji po T\_start\_koncertu, jest to czas w którym „spóźnialscy” mogą wejść podczas trwania symulacji.
* Czas\_aktualny – zmienna zawierająca informacje o aktualnym czasie trwania symulacji
* Lista\_czasow\_czekania – Lista zawierająca czasy czekania każdego z Rockmanow, którzy weszli na obiekt
* Lista\_czasow\_czekania\_w\_chwili\_t\_start – Lista zwierająca czasy czekania każdego z Rockmanow, którzy zdążyli wejść do czasu T\_start\_koncertu
* Ludzie – Lista zawierająca czasy w których pojawiać się będą ludzie
* Ludzie\_za\_bramkami – zmienna zawierająca informacje o liczbie ludzi którzy zdążyli opuścić system, przez przejście przez bramkę.
* Ludzie\_za\_bramkami\_w\_chwili\_t\_start - zmienna zawierająca informacje o liczbie ludzi którzy zdążyli opuścić system, przez przejście przez bramkę w chwili T\_start\_koncertu
* Bramki – Lista zawierająca instancje klasy Bramka
* Bramki\_plecakowe – Lista zawierająca instancje klasy Bramka\_plecak
* Kolejki – Lista zawierająca instancje klasy Kolejka
* Nr\_rockmana – zmienna zawierająca informacje o „imieniu” rockmana (numery wygenerowanych rockmanów żeby się nie pogubić)
  1. Funkcje
* Ustaw\_parametry\_na\_start – funkcja ustawiająca parametry na wartości startowe tak aby symulacje można powtórzyć, bez wyłączania i włączania programu
* Stworz\_bramki\_kolejki – funkcja tworzaca instacje klas: Bramka, Bramka\_plecak oraz Kolejka, zgodnie z ustalonymi parametrami Ile\_bramek
* Wybierz\_najkrotsza\_kolejke – funkcja znajdująca najkrótszą dostępną kolejkę
* Zrob\_ludzi – funkcja wykorzystujaca funkcje przyjścia\_w\_czasie z pliku rozkład.py by uzyskac czasy przychodzenia rockmanów w czasie symulacji.
* Dodawanie\_zgodnie\_z\_czasem – funkcja tworząca tyle instancji klasy Rockman(a przy okazji nadająca im parametr Czy\_plecak) oraz dodająca ich do kolejki (zgodnie z funkcją wybierz\_najkrotsza\_kolejke\_zwykla) ile powinno przyjść w tym momencie ludzi (z listy ludzie)
* Rozgrzewanie\_modelu – funkcja implementująca przychodzenie ludzi, zanim jeszcze bramki się otworzą.
* Iteracja\_bramka\_zwykla – funkcja implementująca sprawdzanie warunków bramki (takich jak: wybieranie 1szego elementu z kolejki jeżeli bramka jest pusta, wypuszanie Rockmanow z bramki jeżeli już zostali sprawdzeni, obsługiwanie Rockmanow, odsyłanie Rockmanów do bramki plecakowej)
* Iteracja bramka\_plecakowa – funkcja implementująca sprawdzenie warunków bramki plecakowej ( takich jak: wybieranie 1szego elementu z kolejki jeżeli bramka jest pusta, wypuszanie Rockmanow z bramki jeżeli już zostali sprawdzeni, obsługiwanie Rockmanow)
* Symulacja\_glowna\_czesc – funkcja wykorzystujaca niektóre z wyżej wspomnianych funkcji aby przeprowadzić symulacje PO rozgrzewaniu modelu (iteracje bramek, dodawanie zgodnie z czasem) oraz dodający wartość czasu czas\_aktualny (przy okazji sprawdzając pytania „czy już się zaczął koncert?” oraz „czy już koniec symulacji?”
* Symulacja – funkcja po kolei odpalająca funkcje: ustaw\_parametry\_na\_start  
  zrob\_ludzi  
  stworz\_bramki\_kolejki  
  rozgrzewanie\_modelu  
  symulacja\_czesc\_glowna  
  Czyli po prostu przeprowadzająca całą symulacje.
* Test\_bramek – funkcja testowa, sprawdzająca stan bramek (czy dobrze zostały przypisane kolejki do bramek)
* Wyprintuj\_ludzi\_w\_kolejkach – funkcja testowa sprawdzająca stan kolejek (kto w nich stoi, ile ich jest)
  1. Opis GUI tkinter

Aby latwiej manipulować zmiennymi wyjściowymi, oraz uporządkować informacje, stworzyliśmy GUI dzięki bibliotece tkinter, można dzięki temu dopasować zmienne wejściowe zgodnie ze swoimi potrzebami, przy okazji stworzyliśmy funkcje Set\_start, która uruchamia symulacje oraz wyświetla informacje na temat dokonanej symulacji po kliknięciu „Symuluj!” w GUI.