1. **Implementacja i opis generatorów, uwzględniająca różne ziarna, dla różnych parametrów i symulacji.**

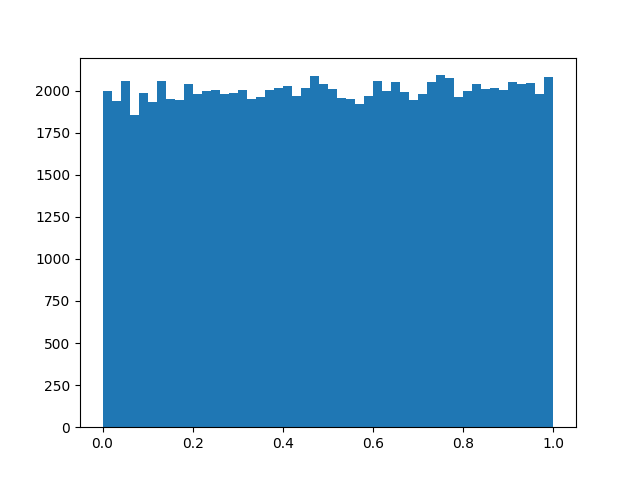
Generatory zostały zaimplementowane w ramach klasy *Generator* na podstawie pseudokodu z polecenia, testowanie ich działania zostało przeprowadzone w narzędziu *Python Console*. Została utworzona lista *fibon*, która przyjmuje wartości z ciągu Fibonacciego. Z tej listy za pomocą *random.choice* losowo wybierana jest jedna wartość i ona służy, jako ziarno w tym konkretnym użyciu generatora.

|  |
| --- |
| import math  import random  class Generator:  def \_\_init\_\_(self):  self.M: int = 2147483647  self.A: int = 16807  self.Q: int = 127773  self.R: int = 2836  self.fibon = [1, 2, 3, 5, 8, …, 701408733]  self.kernel: int = random.choice(self.fibon)  def randomizer(self):  h = math.floor(self.kernel / self.Q)  self.kernel = self.A \* (self.kernel - self.Q \* h) - self.R \* h  if self.kernel < 0:  self.kernel += self.M  return self.kernel / self.M  def randminmax(self, min: int, max: int):  return self.randomizer() \* (max - min) + min  def rndexp(self, lambdavar: int):  k = self.randomizer()  return -(1.0 / lambdavar) \* math.log(k)  def rndzeroone(self, p: float):  k = self.randomizer()  if k < p:  return 1  else:  return 0 |

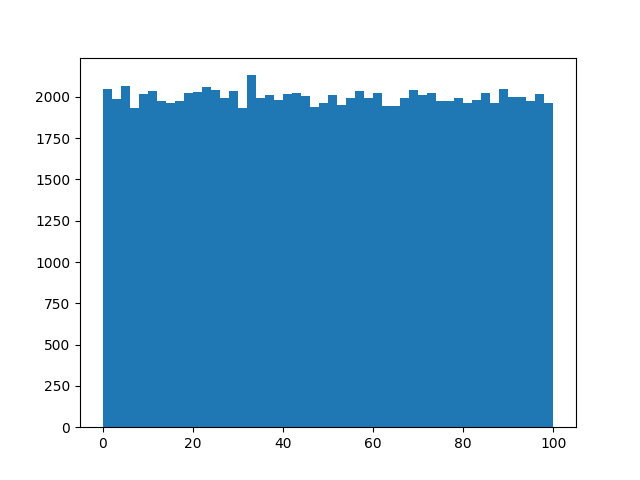
1. **Histogramy generatorów.**

Do wygenerowania histogramów generatorów został utworzony skrypt *Tester.py*.

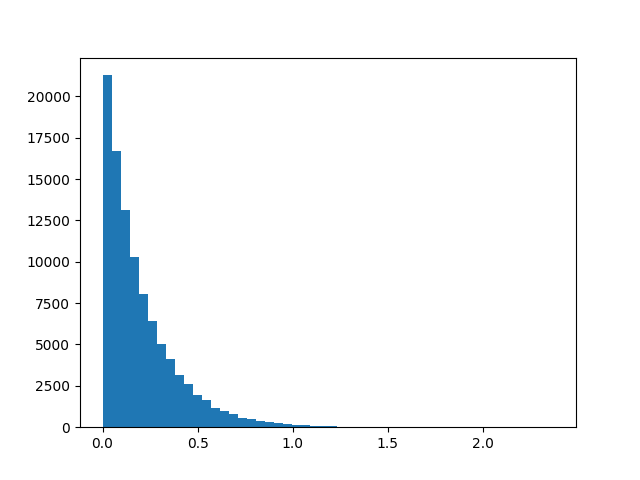
* Metoda *randomizer*



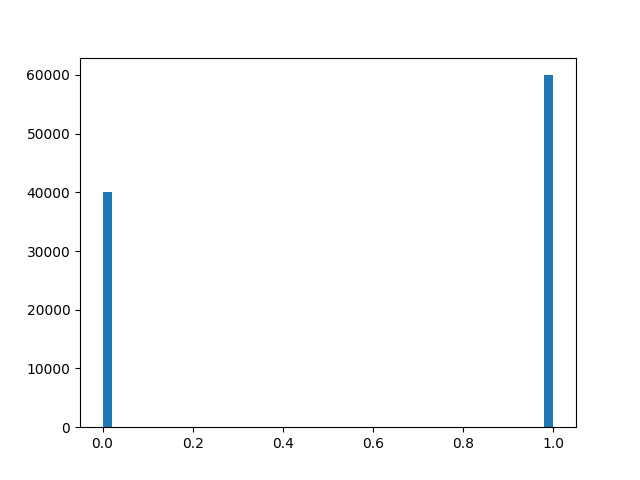
* Metoda *randminmax*, dla wartości *min* = 0 i *max* = 100



* Metoda *rndexp*, dla wartości *lambdavar* = 5



* Metoda *rndzeroone*, dla wartości *p* = 0.6



1. **Implementacja zbierania statystyk, uwzględniająca koniec fazy początkowej i różne warunki końca symulacji.**

Przed uruchomieniem skryptu, użytkownik może podać, jak długo ma trwać symulacja. Odpowiada za to zmienna *\_runtime*, gdzie wartość 1 oznacza jedną sekundę, dodatkowo jest jeszcze zmienna *\_clock*, w wypadku której, wartość 1 oznacza jedną milisekundę. Program będzie pracować, dopóki liczba cykli (zwiększana o wartość zegara przy każdej iteracji) jest mniejsza od wartości zmiennej *\_runtime* przemnożonej przez 1000.

1. **Wyświetlanie uzyskanych wyników w konsoli programu.**

