Міністерство освіти і науки України

Національний авіаційний університет

Факультет кібербезпеки, комп'ютерної та програмної інженерії

**Імітаційне моделювання**

Самостійна робота №1

«Дослідження генераторів випадкових чисел»

Роботу виконав:

студент групи СП-325

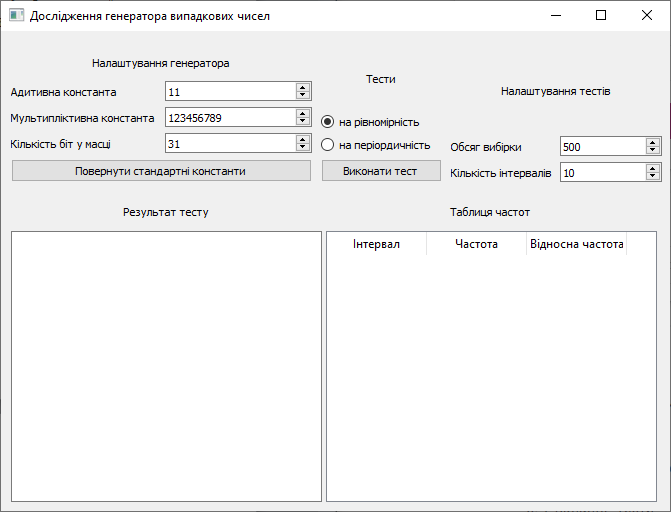
Козлов Олексій

Роботу прийняла:

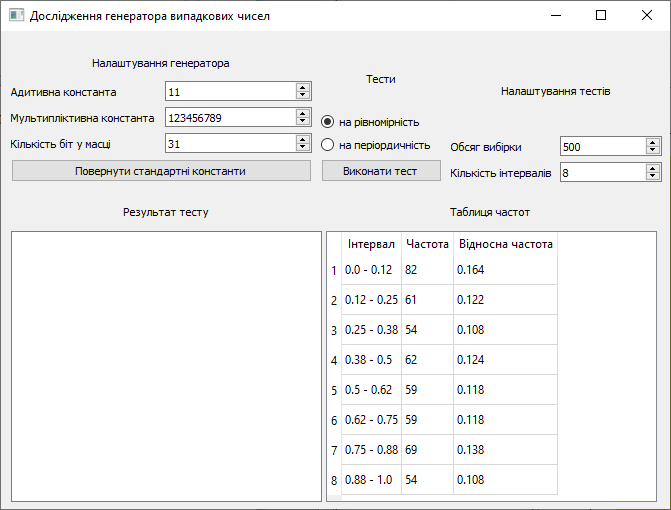
Нечипорук О.П.

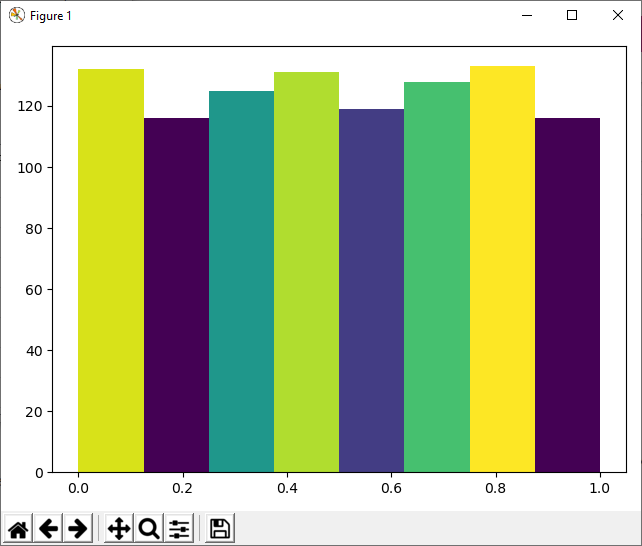
Київ – 2020

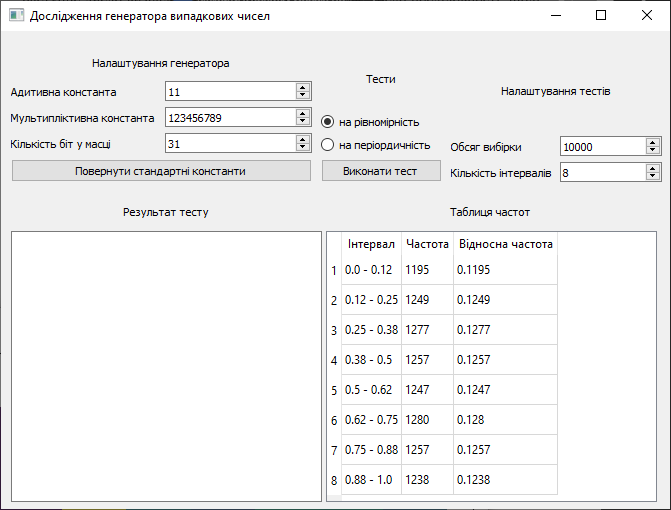
1. Інтерфейс користувача

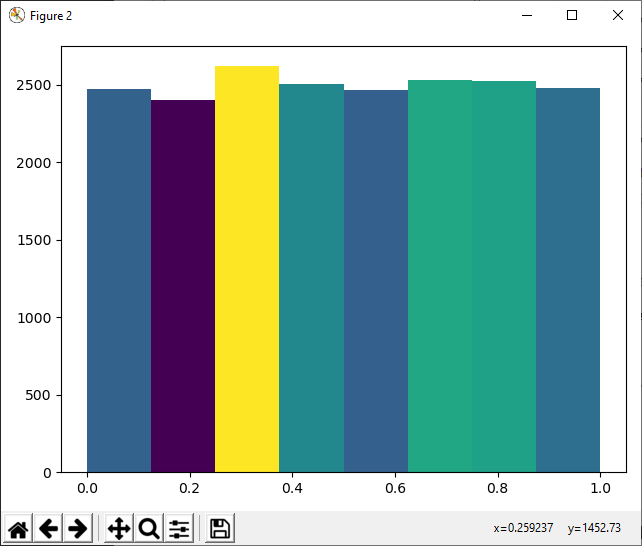


1. Тест на рівномірність розподілу випадкових величин

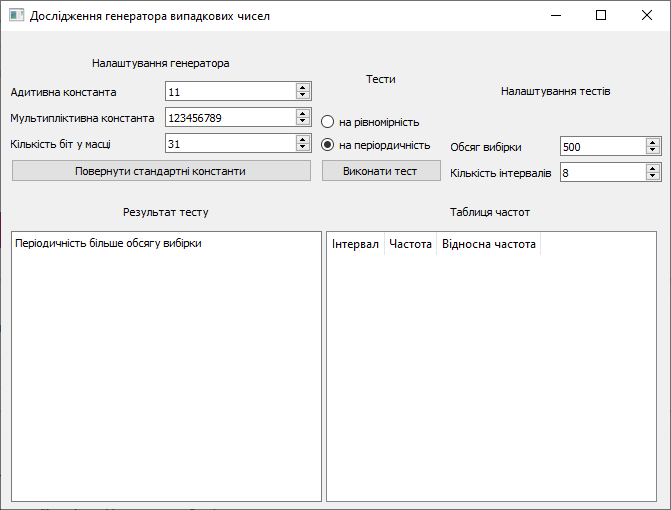


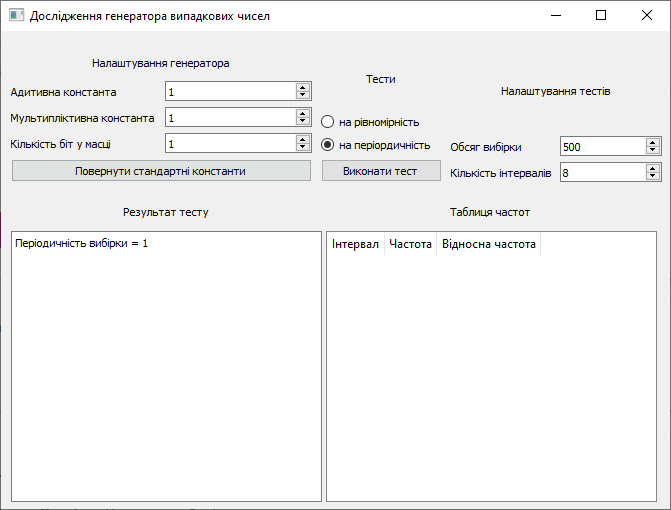






1. Тест на періодичність генерації випадкових чисел





1. Код програми

from PyQt5 import QtWidgets  
from PyQt5.QtWidgets import \*  
from matplotlib import colors, pyplot  
from numpy import \*  
from IM\_SR\_ui import Ui\_MainWindow  
import sys  
  
class MyWindow(QMainWindow, Ui\_MainWindow):  
  
 def \_\_init\_\_(self):  
 QMainWindow.\_\_init\_\_(self)  
 self.setupUi(self)  
  
 self.setWindowTitle("Дослідження генератора випадкових чисел")  
 self.radioButton.setChecked(True)  
 self.pushButton.clicked.connect(self.Reset)  
 self.pushButton\_2.clicked.connect(self.Compute)  
  
 def Reset(self):  
 self.spinBox.setValue(11)  
 self.spinBox\_2.setValue(123456789)  
 self.spinBox\_3.setValue(31)  
  
 def Compute(self):  
 U0 = random.random()  
 U = [U0]  
 A = self.spinBox.value()  
 B = self.spinBox\_2.value()  
 M = self.spinBox\_3.value()  
 length = self.spinBox\_4.value()  
 number\_of\_intervals = self.spinBox\_5.value()  
  
 for \_ in range(length - 1):  
 U.append((U[-1] \* B + A) % M % 1)  
  
 if self.radioButton.isChecked():  
 self.textBrowser.setHtml("")  
 INTERVALS = 1 / number\_of\_intervals  
 self.tableWidget.setRowCount(number\_of\_intervals)  
 frequency = [0] \* int(1 / INTERVALS)  
 intervals = []  
 left\_interval\_border = 0  
 right\_interval\_border = INTERVALS  
 for interval in range(number\_of\_intervals):  
 stringOfInterval = str(float("{0:.2f}".format(left\_interval\_border))) + " - " + str(  
 float("{0:.2f}".format(right\_interval\_border)))  
 intervals.append(stringOfInterval)  
 left\_interval\_border += INTERVALS  
 right\_interval\_border += INTERVALS  
 self.tableWidget.setItem(interval, 0, QTableWidgetItem(stringOfInterval))  
 self.tableWidget.resizeColumnsToContents()  
  
 for number\_idx in range(length):  
 U.append((U[-1] \* B + A) % M % 1)  
 left\_interval\_border = 0  
 right\_interval\_border = INTERVALS  
 for interval\_idx in range(int(number\_of\_intervals)):  
 if left\_interval\_border <= U[number\_idx] < right\_interval\_border:  
 frequency[interval\_idx] += 1  
 break  
 else:  
 left\_interval\_border += INTERVALS  
 right\_interval\_border += INTERVALS  
  
 for interval\_idx in range(int(number\_of\_intervals)):  
 self.tableWidget.setItem(interval\_idx, 1, QTableWidgetItem(str(frequency[interval\_idx])))  
 self.tableWidget.setItem(interval\_idx, 2, QTableWidgetItem(str(frequency[interval\_idx] / length)))  
 self.tableWidget.resizeColumnsToContents()  
  
 fig, axs = pyplot.subplots(1, 1, sharey=True, tight\_layout=True)  
  
 axs.hist(U, number\_of\_intervals)  
 N, bins, patches = axs.hist(U, bins=number\_of\_intervals)  
  
 fracs = N / N.max()  
  
 norm = colors.Normalize(fracs.min(), fracs.max())  
  
 for thisfrac, thispatch in zip(fracs, patches):  
 color = pyplot.cm.viridis(norm(thisfrac))  
 thispatch.set\_facecolor(color)  
  
 pyplot.show()  
 else:  
 self.tableWidget.setRowCount(0)  
 self.tableWidget.resizeColumnsToContents()  
 try:  
 period = U.index(U0, 1)  
 self.textBrowser.setHtml("Періодичність вибірки = " + str(period))  
 except ValueError:  
 self.textBrowser.setHtml("Періодичність більше обсягу вибірки")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app = QtWidgets.QApplication(sys.argv)  
 my\_app = MyWindow()  
 my\_app.show()  
 sys.exit(app.exec\_())

**Висновок:** в ході виконання самостійної роботи ми виконали два види тестування, а саме тестування періодичності та рівномірності розподілу, над послідовністю згенеровано адитивно-мультиплікативно конгруентним методом. Тест рівномірності розподілу показав, що метод може задовільно генерувати рівномірно розподіленні величини при правильних параметрах, при чому рівномірність покращується при збільшенні розміру вибірки. Тест періодичності показав, що при правильних параметрах вона перевищує максимально можливий розмір вибірки, а при неправильних параметрах, періодичність буде становити невелике число.