|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа №1**

**по курсу «Моделирование»**

|  |  |
| --- | --- |
| **Тема: «Изучение функций распределения и плотности распределения случайной величины»**  **Студент Горячев В. Г.**  **Группа ИУ7-75Б**  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель Рудаков И. В.** |  |

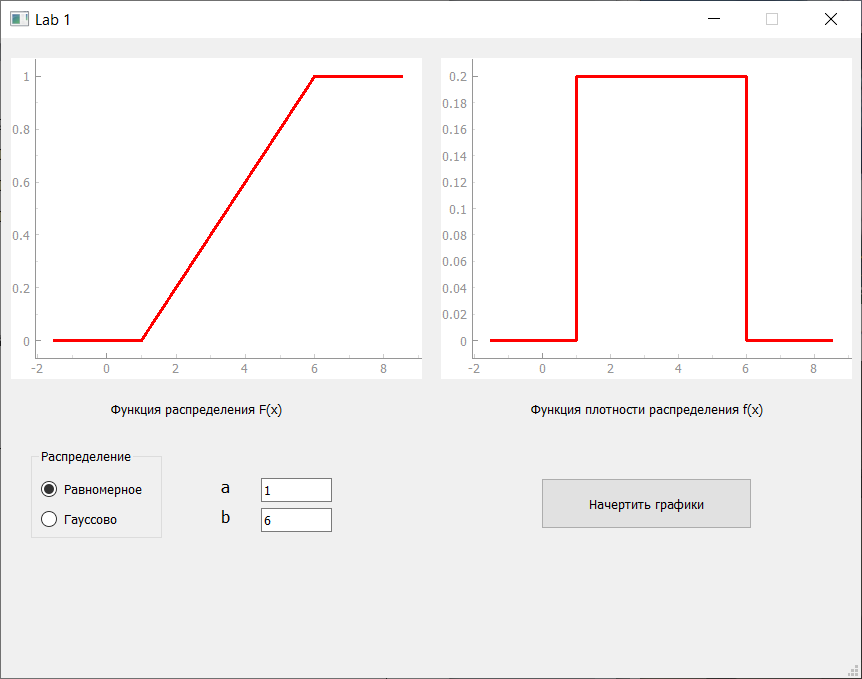
Москва

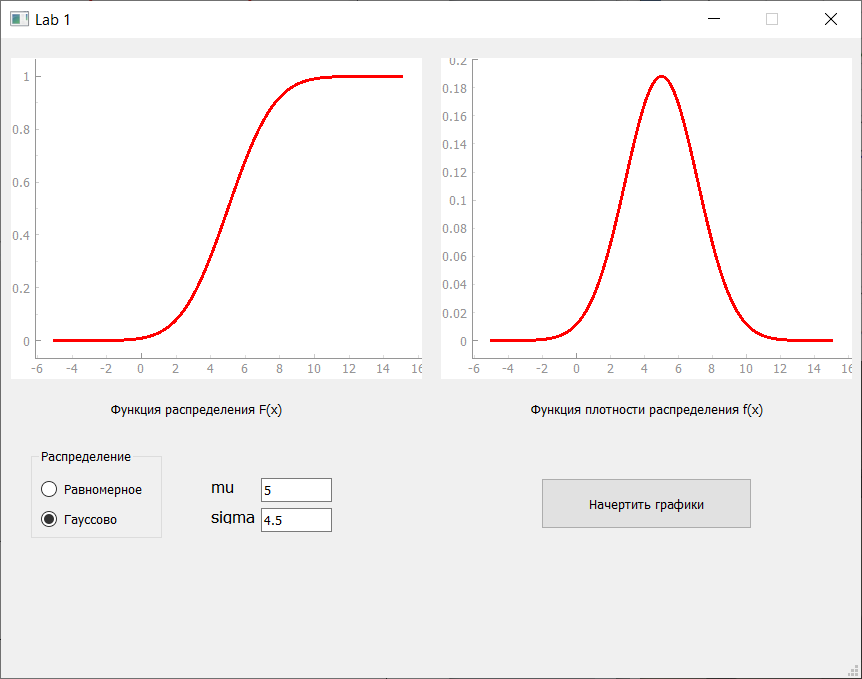
2021 г

**Задание**

Написать программу, строящую графики функции распределения и функции плотности распределения для равномерного и нормального закона распределения случайной величины. Предоставить интерфейс для задания параметров законов.

**Результаты**





**Код программы**

|  |
| --- |
| # This Python file uses the following encoding: utf-8  import sys  from PyQt5.QtWidgets import QApplication, QMainWindow, QErrorMessage, QMessageBox  from PyQt5.QtCore import pyqtSlot, QSize  import pyqtgraph as pg  import numpy as np  from math import sqrt  from scipy.stats import norm  def **ud\_function**(a, b, x):  result = 1  if a <= x < b:  result = (x - a) / (b - a)  elif x < a:  result = 0  return result  def **ud\_density**(a, b, x):  result = 0  if a <= x <= b:  result = 1 / (b - a)  return result  def **norm\_function**(x, mu, sigma):  return norm.cdf(x, mu, sqrt(sigma))  def **norm\_density**(x, mu, sigma):  return norm.pdf(x, mu, sqrt(sigma))  class **MyWindow**(QMainWindow):  def **\_\_init\_\_**(self):  from gui import Ui\_MainWindow  super(MyWindow, self).\_\_init\_\_()  self.ui = Ui\_MainWindow()  self.ui.setupUi(self)  self.initUi()    def **initUi**(self):  self.ui.curve = self.ui.plotView\_F.plot(name = "Line", pen=pg.mkPen('r', width=3))  self.ui.curve2 = self.ui.plotView\_f.plot(name = "Line2", pen=pg.mkPen('r', width=3))  self.ui.pushButton.clicked.connect(self.showPlot)  self.ui.stackedWidget.setCurrentIndex(0)  self.ui.radioButton\_uniform.clicked.connect(  lambda : self.ui.stackedWidget.setCurrentIndex(0)  )  self.ui.radioButton\_normal.clicked.connect(  lambda : self.ui.stackedWidget.setCurrentIndex(1)  )  *@pyqtSlot()*  def **showPlot**(self):  trig = False  if self.ui.radioButton\_uniform.isChecked():  try:  a = float(self.ui.lineEdit\_A.text())  except:  trig = True  self.showError("Некорректный ввод параметра а!")  try:  b = float(self.ui.lineEdit\_B.text())  except:  trig = True  self.showError("Некорректный ввод параметра b!")  if trig:  return  if b < a:  a, b = b, a  elif a == b:  self.showError("Невозможно построить графики при таких параметрах.")  return  delta = b - a  x = np.arange(a - delta / 2, b + delta / 2, 0.001)  y\_function = [ud\_function(a, b, \_x) for \_x in x]  y\_density = [ud\_density(a, b, \_x) for \_x in x]  elif self.ui.radioButton\_normal.isChecked():  try:  mu = float(self.ui.lineEdit\_mu.text())  except:  self.showError("Некорректный ввод параметра mu!")  try:  sigma = float(self.ui.lineEdit\_sigma.text())  except:  self.showError("Некорректный ввод параметра sigma!")  if trig:  return  x = np.arange(mu - 10, mu + 10 , 0.001)  y\_function = norm\_function(x, mu, sigma)  y\_density = norm\_density(x, mu, sigma)  else:  print("?\_?")  self.ui.curve.setData(x, y\_function)  self.ui.curve2.setData(x, y\_density)  # pg.plot(x, y, pen=None, symbol='o')  def **showError**(self, text):  QErrorMessage(self).showMessage(text)  def **showMessage**(self, title, text):  msgBox = QMessageBox(self)  msgBox.setText(text); msgBox.setWindowTitle(title); msgBox.resize(QSize(400, 200))  msgBox.exec()  if \_\_name\_\_ == "\_\_main\_\_":  app = QApplication([])  win = MyWindow()  win.show()  sys.exit(app.exec\_()) |