**Задание:**

Выполнить в графической форме лабораторные работы.

При выполнении Задания 1 решить обратную задачу – нарисовать график функции.

В Задании 2, используя метод Монте-Карло определить площадь заштрихованной части фигуры. Получить точечное изображение фигуры, используя до 10000 испытаний. Выполнить оценку точности вычисления площади в процентах по отношению к реальной площади. Реальную площадь получить геометрическими методами, а при необходимости использовать интегральное исчисление.

В Задании 3 нарисовать график функции, значения которой вычисляются через ряд с точностью 10-3. Изображение графиков должно занимать большую часть экрана, сопровождаться заголовком, содержать наименования и градации осей и масштабироваться в зависимости от исходных данных. При любых допустимых значениях исходных данных изображение должно полностью помещаться на экране. Программа не должна опираться на конкретные значения разрешения экрана.

**Код:**

*from math import sqrt*

*import turtle as tr*

*#*

*def Fun1(x):*

*"""*

*Кривая из лаб. 2 Задание 1*

*"""*

*#if (x >= 5) and (x <= 7):*

*# return(None)*

*if x < -5:*

*y = 1*

*elif x >= -5 and x < 0:*

*y = -(3/5) \* x - 2*

*elif x >= 0 and x < 2:*

*y = -sqrt(4 - x\*\*2)*

*elif x >= 2 and x < 4:*

*y = x - 2*

*elif x >= 4 and x < 8:*

*y = 2 + sqrt(4 - (x - 6)\*\*2)*

*else: y = 2*

*return(y)*

*def Axis(txy, ax = 'X'):*

*"""*

*Рисование оси.*

*txy - список: [Xmin, Xmax]*

*или [Ymin, Ymax]*

*ax - 'X' или 'Y'*

*"""*

*a = txy[0]*

*b = txy[1]*

*tr.up()*

*if (ax == 'X'):*

*pb = [a, 0]*

*pe = [b, 0]*

*else:*

*pb = [0, a]*

*pe = [0, b]*

*tr.goto(pb)*

*tr.down()*

*tr.goto(pe)*

*def Mark(txy, ax = 'X'):*

*"""*

*Маркировка оси.*

*txy - список: [Xmin, Xmax]*

*или [Ymin, Ymax]*

*ax - 'X' или 'Y'*

*"""*

*a = txy[0]*

*b = txy[1]*

*tr.up()*

*for t in range(a, b):*

*if (ax =='X'):*

*pb = [t, 0]*

*pe = [t, 0.2]*

*pw = [t, -0.5]*

*else:*

*pb = [0, t]*

*pe = [0.2, t]*

*pw = [0.2, t]*

*tr.goto(pb)*

*tr.down()*

*tr.goto(pe)*

*tr.up()*

*tr.goto(pw)*

*tr.write(str(t))*

*def Arrow(txy, ax = 'X'):*

*"""*

*Рисование стрелки.*

*txy - список: [Xmin, Xmax]*

*или [Ymin, Ymax]*

*ax - 'X' или 'Y'*

*"""*

*# Параметры многоугольника*

*a = [0.1, 0, -0.1]*

*b = [-0.1, 0.3, -0.1]*

*tr.up()*

*tr.goto(0, 0) # в начало*

*tr.begin\_poly() # начинаем запись вершин*

*for i in range(2): # для всех вершин*

*tr.goto(a[i], b[i]) # многоугольника*

*tr.end\_poly() # останавливаем запись*

*p = tr.get\_poly() # ссылка на многоугольник*

*# регистрируем новую форму черепашке*

*tr.register\_shape("myArrow",p)*

*tr.resizemode("myArrow")*

*tr.shapesize(1,2,1) # растягиваем (пример)*

*if (ax == 'X'): # для оси X*

*tr.tiltangle(0) # угол для формы*

*tr.goto(txy[1]+0.2,0) # к месту стрелки*

*pw = [int(txy[1]),-1.0] # надпись*

*else: # для оси Y*

*tr.tiltangle(90) # угол для формы*

*tr.goto(0,txy[1]+0.2) # к месту стрелки*

*pw = [0.2, int(txy[1])] # надпись*

*tr.stamp() # оставить штамп - стрелка*

*# надпишем ось*

*tr.goto(pw) # к месту надписи*

*tr.write(ax, font=("Arial",14,"bold"))*

*#*

*def main():*

*# Начальные параметры*

*# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*# Границы графика: [Xmin, Xmax] и [Ymin, Ymax]*

*aX = [-12, 12] # левая и правая*

*aY = [-3, 5] # нижняя и верхняя*

*# Главное окно*

*Dx = 800*

*Dy = Dx / ((aX[1] - aX[0]) / (aY[1] - aY[0]))*

*tr.setup(Dx, Dy)*

*tr.reset()*

*# Число точек рисования*

*Nmax = 1000*

*#*

*# Установка мировой системы координат*

*tr.setworldcoordinates(aX[0],aY[0],*

*aX[1],aY[1])*

*#*

*tr.title("Lab\_8\_2\_1") # заголовок*

*tr.width(2) # толщина линии*

*tr.color("blue", "blue") # цвет и заливка*

*# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*tr.ht() # невидимая*

*tr.tracer(0,0) # нет задержек*

*#*

*# X - ось, метки, стрелка*

*Axis(aX, 'X')*

*Mark(aX, 'X')*

*Arrow(aX, 'X')*

*# Y - ось, метки, стрелка*

*Axis(aY, 'Y')*

*Mark(aY, 'Y')*

*Arrow(aY, 'Y')*

*#*

*# Функция*

*tr.color("green") # цвет линии*

*tr.width(3) # и толщина*

*dx = (aX[1]-aX[0])/Nmax # шаг*

*# в начало*

*x = aX[0]*

*y = Fun1(x)*

*if (y is None):*

*tr.up()*

*tr.goto(x, 0)*

*else:*

*tr.goto(x, y)*

*tr.down()*

*# рисуем*

*while x <= aX[1]:*

*x = x + dx*

*y = Fun1(x)*

*if (y is None):*

*tr.up()*

*continue*

*else:*

*tr.goto(x, y)*

*tr.down()*

*#*

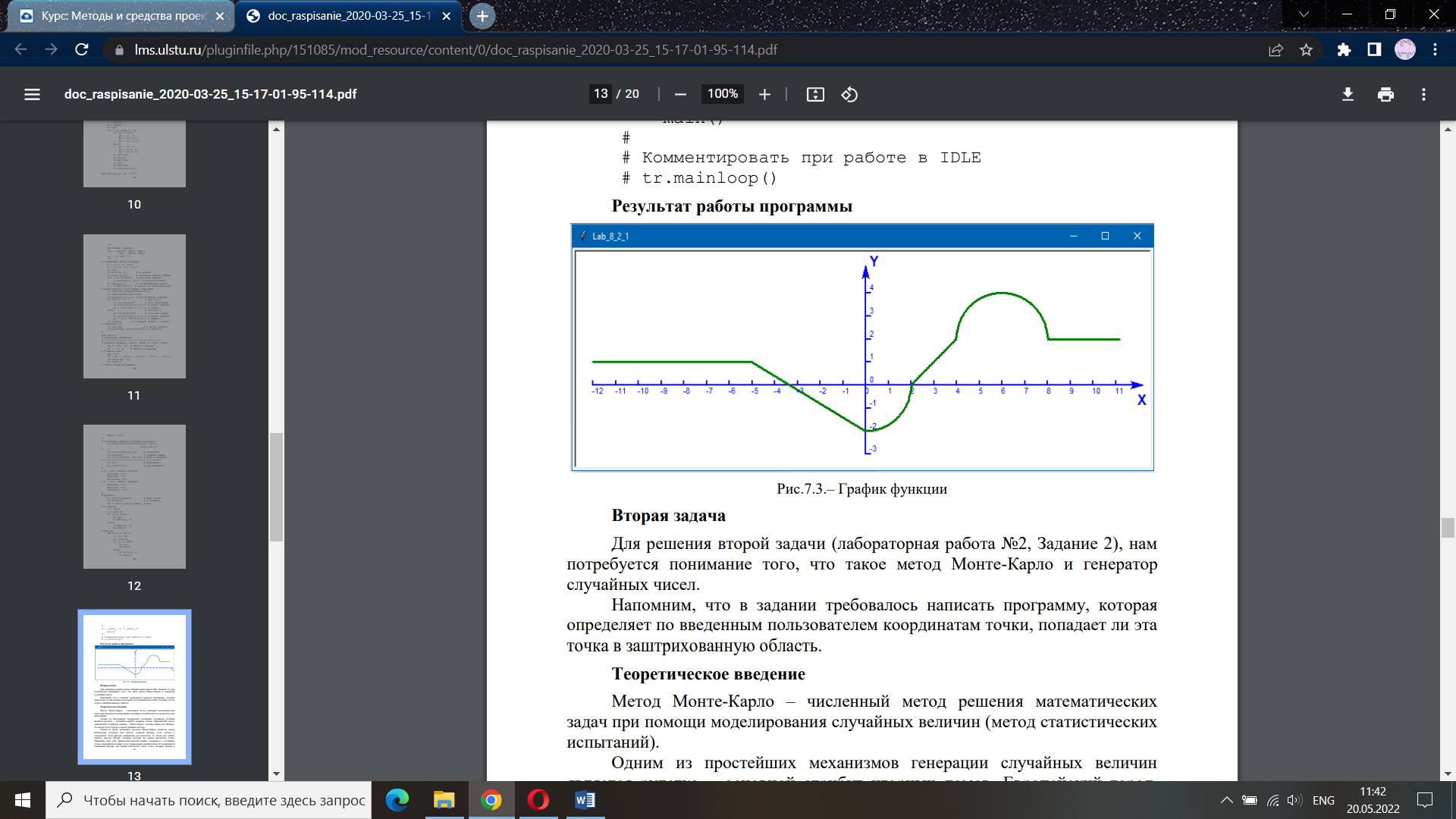
*if \_\_name\_\_ == "\_\_main\_\_":*

*main()*

*#*

*# Комментировать при работе в IDLE*

*# tr.mainloop()*



**Код:**

*# -\*- coding: cp1251 -\*-*

*import turtle as tr*

*from random import uniform*

*#*

*def fun2\_2(x,y):*

*if (x < -1) or (x > 4):*

*flag = 0 #False*

*if ((x>=-1) and (x<1) and (y>=2\*x+2)*

*and (y<=x\*\*3-4\*x\*\*2+x+6) or (x>=1)*

*and (x<=4) and (y>=x\*\*3-4\*x\*\*2+x+6)*

*and (y<=2\*x+2)):*

*flag = 1*

*else:*

*flag = 0*

*return(flag)*

*# Инициализация turtle*

*# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*# Границы графика*

*aX = [-2, 5] # левая и правая*

*aY = [-2, 11] # нижняя и верхняя*

*Dx = 300*

*Dy = Dx/((aX[1] - aX[0])/(aY[1] - aY[0]))*

*tr.setup(Dx, Dy, 200, 200)*

*tr.reset()*

*# число точек рисования*

*Nmax = 10000*

*#*

*# Установка мировой системы координат*

*tr.setworldcoordinates(aX[0], aY[0], aX[1], aY[1])*

*#*

*tr.title("Lab\_8\_2\_2") # заголовок*

*tr.width(2) # толщина линии*

*tr.ht() # невидимая*

*tr.tracer(0,0) # 0 - задержка между обновлениями*

*# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*#*

*# рисование функции*

*tr.up()*

*mfun = 0 # точек попало в*

*# заштрихованную область*

*for n in range(Nmax):*

*# генерируем координаты точки*

*x = uniform(aX[0],aX[1])*

*y = uniform(aY[0],aY[1])*

*tr.goto(x,y)*

*if fun2\_2(x,y) != 0: # попала*

*tr.dot(3,"green")*

*mfun += 1*

*# else: # не попала*

*# tr.dot(3, "#ffccff")*

*#*

*tr.color("blue", "blue") # цвет и заливка*

*#*

*# Рисуем оси*

*# Ось X*

*tr.up()*

*tr.goto(aX[0], 0)*

*tr.down()*

*tr.goto(aX[1], 0)*

*# Ось Y*

*tr.up()*

*tr.goto(0, aY[1])*

*tr.down()*

*tr.goto(0, aY[0])*

*#*

*# координатные метки*

*# и надписи на оси X*

*tr.up()*

*for x in range(aX[0], aX[1]):*

*tr.goto(x, 0.1)*

*tr.down()*

*tr.goto(x, 0)*

*tr.up()*

*tr.sety(-0.4)*

*coords = str(x)*

*tr.write(coords)*

*#*

*# на оси Y*

*for y in range(aY[0], aY[1]):*

*tr.goto(0, y)*

*tr.down()*

*tr.goto(0.1, y)*

*tr.up()*

*tr.setx(0.2)*

*coords = str(y)*

*tr.write(coords)*

*#*

*# Рисуем стрелки*

*# на оси X*

*poli = [0, 0.1, 0, -0.1, 0]*

*Arrbeg = int(aX[1])*

*Xpoli = [Arrbeg, Arrbeg - 0.1, Arrbeg+0.3,*

*Arrbeg - 0.1, Arrbeg]*

*tr.goto(Xpoli[0],poli[0])*

*tr.begin\_fill()*

*tr.down()*

*for i in range(1,5):*

*tr.goto(Xpoli[i], poli[i])*

*tr.end\_fill() # заливаем стрелку*

*# Надпишем ось X*

*tr.up()*

*tr.goto(Arrbeg, -0.7)*

*tr.write("X", font=("Arial",14,"bold"))*

*#*

*# на оси Y*

*Arrbeg = int(aY[1])*

*Ypoli = [Arrbeg, Arrbeg - 0.1, Arrbeg + 0.3,*

*Arrbeg - 0.1, Arrbeg]*

*tr.up()*

*tr.goto(poli[0], Ypoli[0])*

*tr.begin\_fill()*

*tr.down()*

*for i in range(1,5):*

*tr.goto(poli[i],Ypoli[i])*

*tr.end\_fill()*

*# Надпишем ось Y*

*tr.up()*

*tr.goto(0.2, Arrbeg)*

*tr.write("Y", font=("Arial",14,"bold"))*

*Sf = (aX[1] - aX[0]) \* (aY[1] - aY[0]) \* mfun/Nmax*

*tr.goto(1, 9)*

*fstr = "N = {0:8d}\nNf = {1:8d}\nSf = {2:8.2f}"*

*meseg = fstr.format(Nmax, mfun, Sf)*

*tr.write(meseg, font=("Arial",12,"bold"))*

*print(meseg)*

*# Комментировать при работе в IDLE*

*# tr.done()*

