

***Кафедра информационных систем***

|  |  |
| --- | --- |
| ***Рейтинговая работа*** | ***Расчетно-аналитическое задание*** |
| ***по дисциплине*** | ***Алгоритмизация и программирование*** |
|  |  |
| ***Задание / Вариант №*** | ***Индивидуальное задание в соответствие с ФИО и ID*** |
|  |  |
| ***Тема*** | ***Создание графического интерфейса пользователя*** |
|  |  |
| ***Выполнена обучающимся группы*** | з.ИЗДт 23.1/Б1-20 |
| ***ФИО обучающегося*** | Серебряков Юрий Владимирович |
|  |  |
| ***Преподаватель*** | Зайцев Сергей Александрович |

Москва – 2022 г.

**Оглавление**

[Введение 3](#_Toc91030006)

[Постановка задачи 3](#_Toc91030007)

[Блок-схема 4](#_Toc91030008)

[Листинг программы 5](#_Toc91030009)

[Результат работы программы 8](#_Toc91030010)

[Вывод 8](#_Toc91030011)

[Список использованных источников 9](#_Toc91030012)

# Введение

Напишите программу на языке программирования Python, в которой должны быть предусмотрены следующие возможности:

1. Надпись в заголовке главного окна меняется по щелчку мыши с вашей фамилии на ваш ID и обратно.
2. При загрузке программы на главном окне отрисовывается система координат, с обозначениями осей *х* и *у,* полученными из вашего ID.
3. При загрузке программы на главном окне отрисовывается треугольник, координаты *х* которого это последние шесть цифр вашего ID, составляющих три двузначных числа, а координаты *у* – последние шесть цифр вашего ID деленного на 3, составляющих три двузначных числа. Фигура треугольника должна быть закрашена сплошной заливкой.
4. Координаты треугольника можно изменить в главном окне с помощью ввода чисел в текстовые поля. При изменении координат треугольник должен быть перерисован.
5. Цвет треугольника меняется в зависимости от выбранного виджета. Виджеты для выбора цвета сгруппированы во фрейме. Минимальное количество цветов, доступных для выбора – 6, максимальное – 16. Фигура треугольника должна быть закрашена сплошной заливкой такого же цвета.
6. При изменении пользователем размера окна виджеты смещаются относительного нового размера окна.

***Исходные данные***

**Для заголовка окна:**

- ваши фамилия, имя, отчество, напр. Иванов Иван Иванович

- ваш ID, напр.: 70115094

**Для координат треугольника:**

- координаты *х* - это последние шесть чисел вашего ID, составляющих три двузначных числа, напр.: 70115094 – это координаты *х1=*11, *х2=*50, *х3=*94

- координаты *у* – это последние шесть чисел вашего ID деленного на 3, составляющих три двузначных числа, напр.: 70115094 / 3 = 23371698, получаем координаты *у1*=37, *у2=*16, *у3=*98

Таким образом, координаты треугольника равны:

*х1; у1=*(11; 37)

*х2; у2=*(50; 16)

*х3; у3*=(94; 98)

**Примечание:**

*ФИО – Иванов Иван Иванович, и ID – 70115094 представлены в качестве примера. Естественно, при разработке программы, нужно вносить свои ФИО и ID.*

***Пример выполнения работы***

Пример выполнения работы представлен на рисунке 1.

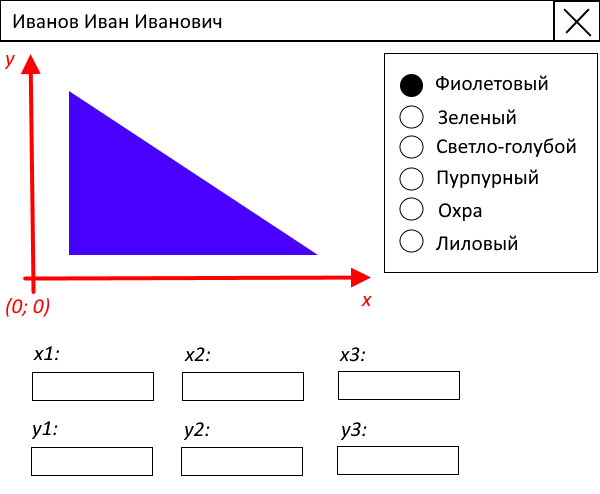


Рисунок 1 – пример выполнения работы

# Постановка задачи

В данной задаче требуется разработать программу с GUI для рисования треугольника по координатам, изначально в программе должен быть нарисован треугольник по координатам, взятым из ID студента. Я решил сразу вписать координаты в поля для ввода, чтобы пользователь знал, какие координаты у изначального треугольника.

Из менеджеров геометрии я выбрал pack, потому что pack наиболее часто используется в трех видах управления компоновкой. Два других менеджера требуют указывать конкретную позицию отображения элемента управления, а в pack можно указывать относительное положение, и точная позиция будет автоматически определена системой упаковки. Этот менеджер достаточно гибок, поэтому pack является предпочтительным менеджером для простых приложений.

# Листинг программы

import tkinter as tk

class App(tk.Tk):

    def \_\_init\_\_(self):

        super().\_\_init\_\_()

        # Вводим переменные

        id = "70174464"

        fio = "Серебряков Юрий Владимирович"

        y0 = 150

        x0 = 10

        header = tk.BooleanVar(False)

        colorName = tk.StringVar()

        colorName.set("blue")

        # Создаем фреймы

        colFrame = tk.Frame(self, borderwidth=1, relief=tk.GROOVE)

        colFrame.pack(side='right', anchor='n', fill='x', padx=10, expand = 1,

pady=10)

        triFrame = tk.Frame(self)

        triFrame.pack(side='top', pady= (10, 0))

        x1y1Frame = tk.Frame(self)

        x1y1Frame.pack(side='left', anchor='s', expand=1, padx=10, pady=10)

        x2y2Frame = tk.Frame(self)

        x2y2Frame.pack(side='left', anchor='s', expand=1, padx=10, pady=10)

        x3y3Frame = tk.Frame(self)

        x3y3Frame.pack(side='left', anchor='s', expand=1, padx=10, pady=10)

        # Класс для создания кнопок

        class RBColor:

            def \_\_init\_\_(self, color):

                tk.Radiobutton(colFrame, text=color, variable=colorName, value=color, command=paint).pack(anchor='w')

        # Класс для создания полей ввода

        class TitEntry:

            def \_\_init\_\_(self, coord, framec, value):

                tk.Label(framec, text=coord).pack(anchor='w')

                self.coords = tk.StringVar()

                self.coords.set(value)

                self.coords.trace('w', inputPainting)

                self.entryEd = tk.Entry(framec, textvariable=self.coords, bd=3)

                self.entryEd.pack(expand=1)

        # Базовый заголовок

        self.title(fio)

        # Функция для смены заголовка

        def headerChange(event):

            header.set(not header.get())

            if (header.get() == False): self.title(fio)

            else: self.title(id)

        # Бинд смены заголовка по нажатию мыши

        self.bind('<Button-1>', headerChange)

        # Функция для рисования треугольника при задавании координат

        def inputPainting(a, b, c):

            for i in range (len(coordsList)):

                if coordsList[i].coords.get() != "":

                    startCoordsList[i] = coordsList[i].entryEd.get()

            canv.coords(rectangle, x0+int(startCoordsList[0]), y0-int(startCoordsList[1]),

                                    x0+int(startCoordsList[2]), y0-int(startCoordsList[3]),

                                    x0+int(startCoordsList[4]), y0-int(startCoordsList[5]))

        # Создаем площадь для рисования фигур

        canv = tk.Canvas(triFrame, width=200, height=200)

        # Рисуем систему координат

        canv.create\_line(15,160,15,10, width=2, arrow=tk.LAST, fill="red")

        canv.create\_line(10,150,150,150, width=2, arrow=tk.LAST, fill="red")

        canv.create\_text(20,170, text="(0; 0)", fill="red", font=("Helvectica", "10"))

        canv.create\_text(25, 10, text="y", fill="red", font=("Helvectica", "10"))

        canv.create\_text(150, 160, text="x", fill="red", font=("Helvectica", "10"))

        # Ищем координаты из ID

        x1 = int(id[2:4])

        x2 = int(id[4:6])

        x3 = int(id[6:8])

        yID = str(int(id)/3)

        y1 = int(yID[2:4])

        y2 = int(yID[4:6])

        y3 = int(yID[6:8])

        startCoordsList = [x1, y1, x2, y2, x3, y3]

        # Рисуем базовый треугольник

        rectangle = canv.create\_polygon([x0+x1, y0-y1], [x0+x2, y0-y2], [x0+x3, y0-y3], fill=colorName.get())

        canv.pack()

        # Рисуем треугольник в зависимости от выбранного цвета

        def paint():

            canv.itemconfig(rectangle, fill=colorName.get())

        # Создаем кнопки

        RBColor('blue')

        RBColor('red')

        RBColor('green')

        RBColor('yellow')

        RBColor('orange')

        RBColor('pink')

        # Создаем поля

        x1Ed = TitEntry("x1:", x1y1Frame, x1)

        y1Ed = TitEntry("y1:", x1y1Frame, y1)

        x2Ed = TitEntry("x2:", x2y2Frame, x2)

        y2Ed = TitEntry("y2:", x2y2Frame, y2)

        x3Ed = TitEntry("x3:", x3y3Frame, x3)

        y3Ed = TitEntry("y3:", x3y3Frame, y3)

        coordsList = [x1Ed, y1Ed, x2Ed, y2Ed, x3Ed, y3Ed]

program = App()

program.mainloop()

# Результат работы программы

Вид программы при запуске изображен на рисунке 2.

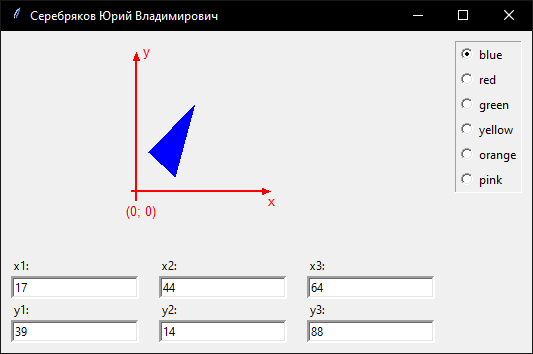


Рисунок 2 – изначальный вид программы

На рисунке 3 изображена программа после выбора цвета треугольника, изменения координат и смены заголовка через щелчок мыши.

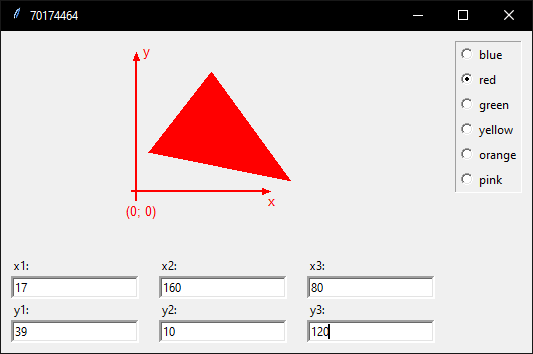


Рисунок 3 – пример работы программы

На рисунке 4 изображено расширенное окно программы, чтобы показать масштабируемость интерфейса.

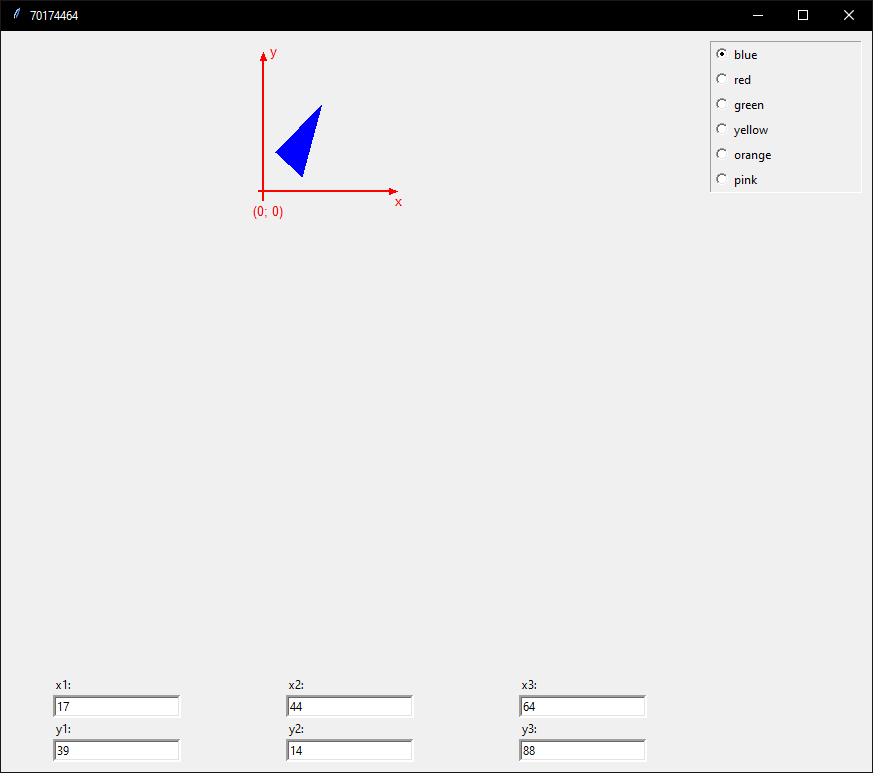


Рисунок 4 – расширенное окно программы

# Вывод

Использовав библиотеку tkinter мы разработали программу с GUI, которая рисует треугольник по ID студента, позволяет изменять цвет треугольника, имеет возможность менять координаты треугольника, меняет заголовок от щелчка мыши и является масштабируемой.

# Список использованных источников

1. ГОСТ 7.32-2017 СИБИД. Отчет о научно исследовательской работе. Структура и правила оформления
2. Лекции электронного университета МУИВ
3. It-black помощь в обучении всем, кто увлекается “Информационными технологиями” и желает повысить свои знания в этой сфере. – URL: <https://it-black.ru> (Дата обращения 29.04.2022)
4. Python world уроки программирования python для начинающих – URL: https://pythonworld.ru (Дата обращения 29.04.2022)
5. CodeRoad перевод популярных вопросов со StackOverflow – URL: <https://coderoad.ru> (Дата обращения 29.04.2022)