Министерство образования и науки России

Рыбинский государственный авиационный технический университет

имени П.А. Соловьева

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

ДИСЦИПЛИНА:

Объектно-ориентированное программирование

ОТЧЁТ   
ПО ЛАБАРАТОРНОЙ РАБОТЕ № 1

Выполнил Медведева А.А.

(фамилия, имя, отчество)

студент гр. ЗИС-23

Преподаватель Пруктишина В.А.

(фамилия, имя, отчество)

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

г. Рыбинск

2024 г.

**Содержание**

[1. Постановка задачи 3](#_Toc184309861)

[2. Спецификация 4](#_Toc184309862)

[3. Теоретические сведения 7](#_Toc184309864)

[3. Организация данных 9](#_Toc184309865)

[5. Текст программы 12](#_Toc184309866)

1. **Постановка задачи**

Задание на лабораторную работу №1:

На языке программирования Python разработать графический редактор Требования:

* должны использоваться визуальные компоненты, с кнопками и формами ввода;
* должны соблюдаться основные принципы ООП, элементы точка и линии должны быть классами, рисование — методы;
* код выкладывать на github, можно protected репо с выдачей доступа аккаунту.

1. **Спецификация**

Разработанная программа представляет собой упрощённый графический редактор, реализованный с использованием языка программирования Python и библиотеки графического интерфейса **Tkinter**. Программа предназначена для создания простых графических объектов, таких как точки и линии, а также для свободного рисования. Графический интерфейс программы предоставляет удобные инструменты для взаимодействия с пользователем.

**Функциональные возможности**

Графический редактор предоставляет следующие функции:

1. Построение точек и линий
   * Пользователь может выбрать режим «Точка», далее нажав мышью на холсте, добавить точку в указанное место;
   * Цвет точки определяется выбранным цветом в палитре, а её размер задаётся толщиной, которая настраивается через поле ввода «Толщина»;
   * В режиме «Точка» программа позволяет автоматически соединять точки линиями. После добавления двух и более точек между ними строится линия с заданной толщиной и цветом.
2. Свободное рисование
   * Пользователь может выбрать режим «Рисование», зажать левую кнопку мыши и провести на холсте линию произвольной формы.
   * В этом режиме произвольная линия рисуется с использованием текущих настроек толщины и цвета.
3. Выбор цвета
   * Программа предоставляет возможность изменять цвет рисуемых объектов. Для этого используется кнопка «Цвет» с окном выбора цвета, которое позволяет пользователю подобрать нужный цвет.
   * При изменении цвета на панели инструментов, новый цвет будет применяться только к новым создаваемым объектам, ранее созданные объекты не поменяют свой цвет.
4. Настройка толщины линий и точек
   * Толщина всех графических элементов настраивается вручную через поле ввода. Пользователь может задать любое целое число в пикселях.
   * Настройки толщины позволяет создавать как тонкие линии для детализированных рисунков, так и толстые линии для крупных элементов.

**Архитектура программы**

Графический редактор разработан с применением объектно-ориентированного подхода (ООП), что позволяет структурировать код, упрощает его чтение и поддержку, а также предоставляет возможность для будущего расширения функционала. Программа состоит из следующих ключевых компонентов:

**Класс «Point»**

* + Представляет отдельную точку, нарисованную на холсте.
  + Хранит информацию о координатах точки, её цвете и толщине.
  + Реализует метод «draw», который создаёт графический объект точки на холсте.

**Класс «Line»**

* + Представляет линию, соединяющую две точки.
  + Содержит информацию о координатах начала и конца линии, её цвете и толщине.
  + Реализует метод «draw», отвечающий за визуализацию линии между заданными точками.

**Класс «Draw»**

* + Это основной класс программы, управляющий графическим интерфейсом и обработкой пользовательских действий.
  + Реализует создание холста и панели инструментов.
  + Обрабатывает события мыши, такие как щелчки и движения, а также переключение между режимами работы.
  + Управляет текущими настройками (цвет, толщина).

**Назначение**

Программа предназначена для образовательных целей. Её функциональность охватывает основные действия, необходимые для визуализации простых графических объектов. Редактор имеет простой и понятный интерфейс, не требует сложных настроек, что делает его доступным для пользователей любого уровня подготовки. Объектно-ориентированный подход к разработке также обеспечивает основу для будущих расширений и улучшений.

# Теоретические сведения

**Запуск программы**

1. Для запуска программы необходимо найти исполняющий файл «programm.exe».
2. Дважды щёлкнуть на исполняемый файл, чтобы открыть графический редактор. После запуска программы на экране появится окно с белым холстом и панелью инструментов.

**Графический интерфейс**

После запуска программы пользователь увидит следующее:

* **Холст -** белое поле, предназначенное для создания графических элементов. Все рисунки, включая точки, линии и свободные линии, размещаются именно на этом холсте;
* **Панель инструментов** **-** верхняя часть окна, содержащая кнопки и элементы управления;
* **Кнопка «Цвет»** **-** открывает палитру для выбора цвета, который будет применяться к новым графическим элементам;
* **Поле ввода толщины** **-** позволяет задавать толщину линий и точек. Толщина указывается в пикселях, пользователь может вводить любое положительное целое число;
* **Кнопка «Точка» -** переключает программу в режим построения отдельных точек и соединительных линий между ними;
* **Кнопка «Рисование»** **-** переключает программу в режим свободного рисования произвольных линий.

**Принципы работы**

1. Режим «Точка»

После выбора кнопки «Точка» пользователь может щёлкнуть мышью в любом месте холста, чтобы создать отдельную точку. Каждая точка имеет выбранный цвет и толщину. Если пользователь добавляет несколько точек подряд, между ними автоматически строится линия.

1. Режим «Рисование»

После выбора кнопки «Рисование» пользователь может зажать левую кнопку мыши и, двигая курсор по холсту, рисовать произвольные линии.

1. Выбор цвета

Для изменения цвета объектов необходимо нажать кнопку «Цвет» на панели инструментов. Откроется окно выбора цвета, где пользователь может выбрать любой оттенок. Новый цвет будет применяться ко всем последующим графическим элементам, но уже созданные элементы сохранят предыдущий цвет.

1. Настройка толщины

Толщина линии или точки задаётся в поле ввода, находящемся на панели инструментов. Если пользователь укажет некорректное значение (например, не число, а букву), программа автоматически применит стандартную толщину по умолчанию **-** 3 пикселя.

**Пример работы**

Для построения последовательности точек с линиями:

* 1. Нажмите кнопку «Точка».
  2. Щёлкните мышью в разных местах холста.
  3. Между точками последовательно будут строиться линии.

Для рисования произвольных линий:

* 1. Нажмите кнопку «Рисование».
  2. Зажмите левую кнопку мыши и перемещайте курсор, рисуя линии.

1. **Организация данных**

Код программы организован в соответствии с принципами объектно-ориентированного программирования (ООП). Основные элементы графического редактора представлены в виде классов, а их функциональность реализована через методы. Ниже подробно рассмотрены ключевые компоненты программы и их взаимодействие.

**Основные классы:**

1. **Класс «Point» -** этот класс отвечает за создание отдельных точек на холсте.

**Атрибуты:**

* + - «canvas»: объект холста, на котором будет отображаться точка.
    - «x», «y»: координаты точки на холсте.
    - «color»: цвет точки.
    - «thickness»: толщина точки.

**Методы:**

* + - «init»: конструктор класса, инициализирует параметры точки и вызывает метод «draw».
    - «draw»: метод, создающий точку в виде овала с использованием метода «create\_oval» объекта холста.

Принцип работы: при создании объекта класса «Point» автоматически вызывается метод «draw», который рисует точку с заданными параметрами.

1. **Класс «Line» -** класс реализует построение линии между двумя заданными точками.

**Атрибуты:**

* + - «Canvas»: объект холста, на котором будет отображаться линия.
    - «x1», «y1»: координаты начальной точки линии.
    - «x2», «y2»: координаты конечной точки линии.
    - «color: цвет линии.
    - «thickness»: толщина линии.

**Методы:**

* + - «init»: конструктор класса, инициализирует параметры линии и вызывает метод «draw».
    - «draw»: метод, создающий линию с использованием метода «create\_line» объекта холста.

Принцип работы: линия строится автоматически при создании объекта класса «Line».

1. **Класс Draw -** класс программы, управляющий графическим интерфейсом и обработкой событий. Этот класс отвечает за создание холста, панелей инструментов и взаимодействие пользователя с программой.

**Атрибуты:**

* + - «root»: главное окно приложения.
    - «canvas»: объект холста для рисования.
    - «current\_color»: текущий цвет, используемый для рисования.
    - «current\_thickness»: текущая толщина линий и точек.
    - «previous\_point»: хранит объект предыдущей точки для построения линий между точками.
    - «drawing\_mode»: текущий режим рисования («point» или «draw»).

**Методы:**

* «init»: инициализирует интерфейс, настраивает холст, панель инструментов и события.
* «create\_toolbar»: создаёт панель инструментов с кнопками для выбора цвета, режима рисования и ввода толщины.
* «bind\_canvas\_events»: привязывает события мыши к функциям обработки нажатий и движения.
* «choose\_color»: открывает диалог выбора цвета, позволяя изменить текущий цвет.
* «get\_thickness»: считывает значение толщины из текстового поля. Если введённое значение некорректно, используется стандартная толщина.
* «set\_point\_mode»: устанавливает режим построения точек.
* «set\_draw\_mode»: устанавливает режим свободного рисования.
* «on\_click»: обрабатывает нажатие левой кнопки мыши. В зависимости от режима рисования:
* В режиме «point» создаёт новую точку, а также соединяет её линией с предыдущей, если таковая имеется.
* «on\_drag»: обрабатывает движение мыши при зажатой кнопке. В режиме «draw» создаёт свободные линии.

**Принципы работы:**

**Графический интерфейс:**

* + При запуске программы создаётся главное окно приложения с холстом и панелью инструментов.
  + Пользователь выбирает параметры (цвет, толщину, режим) и взаимодействует с холстом.

**Режимы работы:**

* **Режим «point»:** пользователь щёлкает мышью на холсте, чтобы создать отдельные точки. Если последовательность точек продолжается, между ними автоматически строятся линии.
* **Режим «draw»:** пользователь зажимает левую кнопку мыши и рисует свободные линии, перемещая курсор.

Программа построена так, чтобы каждый элемент графического редактора представлялся отдельным классом. Это упрощает расширение функциональности, поскольку для добавления новых инструментов или форм достаточно реализовать соответствующий класс с методами рисования.

1. **Текст программы**

import tkinter as tk  
from tkinter.colorchooser import askcolor  
  
# Класс для отдельной точки  
class Point:  
 def \_\_init\_\_(self, canvas, x, y, color, thickness):  
 self.canvas = canvas  
 self.x = x  
 self.y = y  
 self.color = color  
 self.thickness = thickness  
 self.draw()  
  
 def draw(self):  
 # Метод рисует точку на заданной позиции  
 self.item = self.canvas.create\_oval(  
 self.x - self.thickness // 2,  
 self.y - self.thickness // 2,  
 self.x + self.thickness // 2,  
 self.y + self.thickness // 2,  
 fill=self.color, outline=self.color  
 )  
  
# Класс для линии, которая соединяет две точки  
class Line:  
 def \_\_init\_\_(self, canvas, x1, y1, x2, y2, color, thickness):  
 self.canvas = canvas  
 self.x1 = x1  
 self.y1 = y1  
 self.x2 = x2  
 self.y2 = y2  
 self.color = color  
 self.thickness = thickness  
 self.draw()  
  
 def draw(self):  
 # Метод рисует линию между двумя заданными точками  
 self.item = self.canvas.create\_line(  
 self.x1, self.y1, self.x2, self.y2,  
 fill=self.color, width=self.thickness  
 )  
  
# Основной класс для графического редактора  
class Draw:  
 def \_\_init\_\_(self, root):  
 self.root = root  
 self.root.title("Редактор")  
  
 # Начальные параметры цвета и толщины линии  
 self.current\_color = "#000000"  
 self.current\_thickness = 3  
  
 # Переменные для хранения точек при построении линии  
 self.previous\_point = None  
  
 # История действий для отмены, если понадобится  
 self.history = []  
  
 # Создание холста  
 self.canvas = tk.Canvas(root, bg="white", width=600, height=400)  
 self.canvas.pack()  
  
 # Создание панели инструментов  
 self.create\_toolbar()  
  
 # Привязка событий к холсту  
 self.bind\_canvas\_events()  
 self.drawing\_mode = None  
  
 def create\_toolbar(self):  
 # Создание панели инструментов для выбора цвета, толщины линии и режима рисования  
 toolbar = tk.Frame(self.root)  
 toolbar.pack()  
  
 tk.Button(toolbar, text="Цвет", command=self.choose\_color).pack(side=tk.LEFT)  
 tk.Label(toolbar, text="Толщина:").pack(side=tk.LEFT)  
 self.thickness\_entry = tk.Entry(toolbar, width=3)  
 self.thickness\_entry.insert(0, str(self.current\_thickness))  
 self.thickness\_entry.pack(side=tk.LEFT)  
  
 tk.Button(toolbar, text="Точка", command=self.set\_point\_mode).pack(side=tk.LEFT)  
 tk.Button(toolbar, text="Рисование", command=self.set\_draw\_mode).pack(side=tk.LEFT)  
  
 def bind\_canvas\_events(self):  
 # Привязка событий к холсту для рисования  
 self.canvas.bind("<Button-1>", self.on\_click)  
 self.canvas.bind("<B1-Motion>", self.on\_drag)  
  
 def choose\_color(self):  
 # Выбор цвета с помощью диалогового окна и сброс предыдущей точки  
 color = askcolor(color=self.current\_color)[1]  
 if color:  
 self.current\_color = color  
 self.previous\_point = None # сброс предыдущей точки для новой линии  
  
 def get\_thickness(self):  
 # Получение толщины линии из поля ввода  
 try:  
 return int(self.thickness\_entry.get())  
 except ValueError:  
 return self.current\_thickness  
  
 def set\_point\_mode(self):  
 # Установка режима "Точка" для построения линий по точкам  
 self.drawing\_mode = "point"  
 self.previous\_point = None # Сброс предыдущей точки  
  
 def set\_draw\_mode(self):  
 # Установка режима "Рисование" для свободного рисования  
 self.drawing\_mode = "draw"  
  
 def on\_click(self, event):  
 # Обработка нажатия мыши в зависимости от выбранного режима  
 thickness = self.get\_thickness()  
  
 if self.drawing\_mode == "point":  
 # Если в режиме "Точка", создаем новую точку  
 new\_point = Point(self.canvas, event.x, event.y, self.current\_color, thickness)  
 self.history.append(new\_point.item)  
  
 if self.previous\_point:  
 # Если есть предыдущая точка, создаем линию между предыдущей и новой точками  
 line = Line(  
 self.canvas,  
 self.previous\_point.x, self.previous\_point.y,  
 event.x, event.y,  
 self.current\_color, thickness  
 )  
 self.history.append(line.item)  
  
 # Сохраняем текущую точку как предыдущую для следующего шага  
 self.previous\_point = new\_point  
  
 def on\_drag(self, event):  
 # Обработка движения мыши при зажатой кнопке в режиме "Рисование"  
 if self.drawing\_mode == "draw":  
 thickness = self.get\_thickness()  
 # Создаем линию между последней и текущей позицией курсора  
 line = self.canvas.create\_line(  
 event.x - 1, event.y - 1, event.x, event.y,  
 fill=self.current\_color, width=thickness  
 )  
 self.history.append(line)  
  
root = tk.Tk()  
app = Draw(root)  
root.mainloop()