## МААЛЫМАТТЫК ТЕХНОЛОГИЯЛАР / ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.432
DOI 10.33514/1694-7851-2025-1/1-106-113

**Барганалиева Ж.К.**
ага окутуучу
И. Арабаев атындагы Кыргыз мамлекеттик университети
Бишкек ш.
barganalieva@mail.ru
**Султанбаева Г.С.**
педагогика илимдеринин кандидаты, доцент
И. Арабаев атындагы Кыргыз мамлекеттик университети
Бишкек ш.
gul_878787@mail.ru
**Садырова М.Р.**
ага окутуучу
И. Арабаев атындагы Кыргыз мамлекеттик университети
Бишкек ш.
sadyrova.m.r@mail.ru
**Смайылбек кызы Ч.**
ага окутуучу
И. Арабаев атындагы Кыргыз мамлекеттик университети
Бишкек ш.
Smayylbekova@mail.ru

# PYTHONДО TKINTER МЕНЕН ЖӨНӨКӨЙ ГРАФИКАЛЫК РЕДАКТОР ТҮЗҮҮ

**Аннотация:** Макала Pythondун стандарттык дистрибуциясына кирген Tkinter библиотекасын колдонуу менен Pythondо жөнөкөй графикалык редакторду түзүүгө арналган. Ал колдонуучуга сызыктарды тартууга, щетка түстөрүн өзгөртүүгө жана куралдардын жөнөкөй топтомун колдонуу менен холстту тазалоого мүмкүндүк берген негизги GUIди иштеп чыгуу процессин деталдаштырат. Макала проектти ишке ашыруу үчүн керектүү инструменттердин жана чөйрөнү даярдоонун сүрөттөлүшү менен башталат, андан кийин даяр коддуу мисал келтирилген. Фокус чычкан окуяларын башкарууга жана холст менен өз ара аракеттенүүгө бурулуп, холстто баштапкы жана акыркы координаттарын тандоо менен сызыктарды тартууга мүмкүндүк берет.

Макалада холстту тазалоо жана щетка түстөрүн өзгөртүү сыяктуу негизги функциялардан тышкары, редакторду өркүндөтүүнүн мүмкүн болгон варианттары, анын ичинде сүрөт тартуу мүмкүнчүлүктөрүңүздү кеңейтүү үчүн куралдарды жана функцияларды кошуу каралат. Кодду жана ыкмаларды этап-этабы менен түшүндүрүү материалды үйрөнүүнү жеңилдетет жана негизги Python билими бар колдонуучуларга графикалык редакторду өз алдынча ишке ашырууга мүмкүндүк берет, ал татаал графика жана сүрөттөрдү иштетүү проектисине негиз түзө алат.

**Негизги сөздөр:** Python, Tkinter, графикалык редакторс, программалоо, окутуу, окуяны иштетүү, инструменттер, холст, колдонуучу.

**Барганалиева Ж.К.**
старший преподаватель
Кыргызский государственный университет имени И.Арабаева
г. Бишкек
barganalieva@mail.ru
**Султанбаева Г.С.**
кандидат педагогических наук, доцент
Кыргызский государственный университет имени И.Арабаева
г. Бишкек
gul_878787@mail.ru
**Садырова М.Р.**
старший преподаватель
Кыргызский государственный университет имени И.Арабаева
г. Бишкек
sadyrova.m.r@mail.ru
**Смайылбек кызы Ч.**
старший преподаватель
Кыргызский государственный университет имени И.Арабаева
г. Бишкек
Smayylbekova@mail.ru

# СОЗДАНИЕ ПРОСТОГО ГРАФИЧЕСКОГО РЕДАКТОРА НА TKINTER В PYTHON

**Аннотация:** Статья посвящена созданию простого графического редактора на языке Python с использованием библиотеки Tkinter, которая входит в стандартную поставку Python. В ней подробно описан процесс разработки базового графического интерфейса, который позволяет пользователю рисовать линии, изменять цвет кисти и очищать холст с помощью простого набора инструментов. Статья начинается с описания необходимых инструментов и подготовки окружения для реализации проекта, после чего приводится готовый пример кода. Основное внимание уделено обработке событий мыши и взаимодействию с холстом, что позволяет рисовать линии, выбирая начальные и конечные координаты на холсте.

Помимо базовых функций, таких как очистка холста и смена цвета кисти, статья также рассматривает возможные варианты улучшения редактора, включая добавление инструментов и функционала для расширения возможностей рисования. Пошаговое объяснение кода и методов обеспечивает легкость в освоении материала и позволяет пользователям с базовыми знаниями Python самостоятельно реализовать графический редактор, который может стать основой для более сложных проектов в области графики и обработки изображений.

**Ключевые слова:** Python, Tkinter, графический редактор, программирование, обучение, обработка событий, инструменты, холст, пользователь.

**Barganalieva Zh.K.**
senior lecturer
Kyrgyz State University named after I. Arabaev
Bishkek
barganalieva@mail.ru

**Sultanbaeva G.S.**
PhD, Associate Professor
Kyrgyz State University named after I. Arabaev
Bishkek
gul_878787@mail.ru
**Sadyrova M.R.**
senior lecturer
Kyrgyz State University named after I. Arabaev
Bishkek
sadyrova.m.r@mail.ru
**Smayylbek kyzy Ch.**
senior lecturer
Kyrgyz State University named after I. Arabaev
Bishkek
Smayylbekova@mail.ru

# CREATION OF A SIMPLE GRAPHICS EDITOR ON TKINTER IN PYTHON

**Abstract:** This article is about creating a simple graphical editor in Python using the Tkinter library, which is included in the standard Python distribution. It describes in detail the process of developing a basic graphical interface that allows the user to draw lines, change the color of the brush, and clear the canvas using a simple set of tools. Article starts with a description of the necessary tools and preparation of the environment for the implementation of the project, after which a ready-made code example is given. The main focus is on handling mouse events and interacting with the canvas, which allows you to draw lines by selecting the start and end coordinates on the canvas. In addition to basic functions such as clearing the canvas and changing the brush color, the article also discusses possible options for improving the editor, including adding tools and functionality to expand drawing capabilities. Step-by-step explanation of the code and methods ensures ease of learning and allows users with basic knowledge of Python to implement it on their own. a graphics editor that can become the basis for more complex projects in the field of graphics and image processing.

**Keywords:** Python, Tkinter, graphics editor, programming, training, event handling, tools, canvas, user.

В программировании Python существует множество инструментов для создания графического интерфейса пользователя (GUI). Одной из наиболее востребованных и доступных библиотек для этих целей является Tkinter. Эта библиотека предоставляет простой и эффективный способ разработки GUI, делая её идеальным выбором для начинающих разработчиков. В данной статье будет представлен процесс создания простого графического редактора с использованием Python и библиотеки Tkinter, который поддерживает базовые функции рисования, смены цвета и очистки холста.

Tkinter, являясь стандартной библиотекой для создания GUI в Python, обладает следующими преимуществами:

Простота и доступность: Встроенный в стандартную библиотеку Python, Tkinter предоставляет понятный и интуитивный API, что позволяет быстро осваивать основы GUI программирования.

Кроссплатформенность: Программы, разработанные с использованием Tkinter, могут запускаться на операционных системах Windows, macOS и Linux.

Широкое сообщество и документация: Наличие обширного сообщества разработчиков и доступ к большому количеству ресурсов, таких как учебные пособия и документация, значительно упрощают освоение и использование Tkinter [8, С.286-294].

Постановка задачи

В рамках данной статьи мы рассмотрим создание базового графического редактора с возможностями рисования линий, смены цвета кисти и очистки холста. Этот редактор будет включать в себя несколько основных элементов: холст для рисования, кнопки управления для очистки и смены цвета кисти, а также методы для реализации логики рисования.

Первый шаг при создании любого GUI-приложения на Python заключается в импорте библиотеки Tkinter:

```
import tkinter as tk
from tkinter import colorchooser
```

Далее создается основное окно приложения:

```
root = tk.Tk()
root.title("Простой графический редактор")
root.geometry("600x500")
```

В основе графического редактора лежит холст, который реализуется с помощью виджета Canvas. Холст позволяет пользователю рисовать, фиксируя координаты точки начала и конца каждой линии.

Для реализации редактора создается класс SimplePaint, содержащий методы для инициализации элементов и обработки событий мыши:

```
class SimplePaint:
    def __init__(self, root):
        self.root = root
        self.canvas = tk.Canvas(self.root, bg="white", width=500, height=400)
        self.canvas.pack()
        self.pen_color = "black"
        # Привязка событий мыши
        self.canvas.bind("<Button-1>", self.start_paint)
        self.canvas.bind("<B1-Motion>", self.paint)
```

Методы start_paint и paint отвечают за обработку событий:

- start_paint сохраняет начальные координаты точки, откуда начинается рисование.
- paint создает линию между точками, образуя непрерывный рисунок.

```
    def start_paint(self, event):
        self.last_x = event.x
        self.last_y = event.y
    def paint(self, event):
        x, y = event.x, event.y
        self.canvas.create_line((self.last_x, self.last_y, x, y), fill=self.pen_color, width=2)
        self.last_x = x
        self.last_y = y
```

Теперь добавим дополнительный функционал: Очистка и смена цвета кисти

Функция очистки холста позволяет пользователю удалить все элементы с холста:

```
    def clear_canvas(self):
        self.canvas.delete("all")
```

Смена цвета кисти реализована с помощью модуля colorchooser, который предоставляет диалоговое окно для выбора цвета:

```
    def change_color(self):
        color = colorchooser.askcolor() [1, С. 140.]
        if color:
            self.pen_color = color
```

Далее добавляем кнопки для вызова этих функций:

```
        clear_button = tk.Button(self.root, text="Очистить", command=self.clear_canvas)
        clear_button.pack()
        color_button = tk.Button(self.root, text="Сменить цвет", command=self.change_color)
        color_button.pack()
```

Для запуска приложения создается основная функция, которая инициализирует главное окно и запускает цикл обработки событий:

```
def main():
    root = tk.Tk()
    app = SimplePaint(root)
    root.mainloop()
if __name__ == "__main__":
    main()
```

Полный код программы выглядит следующим образом:

```
import tkinter as tk
from tkinter import colorchooser
class SimplePaint:
    def __init__(self, root):
        self.root = root
        self.root.title("Простой графический редактор")

        self.pen_color = "black"
        self.canvas = tk.Canvas(self.root, bg="white", width=400, height=400)
        self.canvas.pack()
        self.canvas.bind("<Button-1>", self.start_paint)
        self.canvas.bind("<B1-Motion>", self.paint)
        clear_button = tk.Button(self.root, text="Очистить", command=self.clear_canvas)
        clear_button.pack()
      color_button = tk.Button(self.root, text="Сменить цвет", command=self.change_color)
        color_button.pack()
    def start_paint(self, event):
        self.last_x = event.x
        self.last_y = event.y
    def paint(self, event):
        x, y = event.x, event.y
        self.canvas.create_line((self.last_x, self.last_y, x, y), fill=self.pen_color, width=2)
        self.last_x = x
```

```
            self.last_y = y
        def clear_canvas(self):
            self.canvas.delete("all")
        def change_color(self):
            color = colorchooser.askcolor()[2, С. 156]
            if color:
                self.pen_color = color
    def __main__():
        root = tk.Tk()
        app = SimplePaint(root)
        root.mainloop()
    if __name__ == "__main__":
        __main__()
```

На рисунке 1 показано изображение после запуска программы.

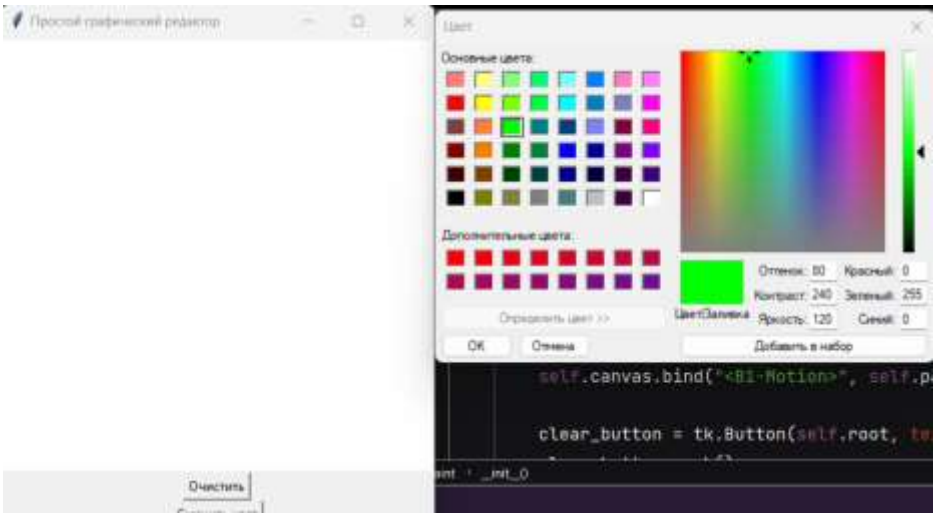


Рис.1. Изображение после запуска программы.

На рисунке 2 показан рисунок, который нарисовал второклассник на данном графическом редакторе.




Рис.2. Рисунок, нарисованный второклассником

На рисунке 3 отображается фрагмент кода на Python.

```python
class SimplePaint:
    def __init__(self, root):
        self.root = root
        self.root.title("Простой графический редактор")

        self.pen_color = "black"

        self.canvas = tk.Canvas(self.root, bg="white", width=400, height=400)
        self.canvas.pack()

        self.canvas.bind("<Button-1>", self.start_paint)
        self.canvas.bind("<B1-Motion>", self.paint)

        clear_button = tk.Button(self.root, text="Очистить", command=self.clear_canvas)
```

Рис.3.Фрагмент кода.

Итак, создание простого графического редактора с использованием библиотеки Tkinter — это удобный способ для начинающих разработчиков Python освоить базовые концепции программирования GUI. Данная статья продемонстрировала ключевые аспекты разработки редактора, такие как работа с холстом, обработка событий мыши и использование диалоговых окон для выбора цвета. Tkinter, благодаря своей простоте и поддержке со стороны сообщества, является оптимальным инструментом для создания несложных графических приложений.

## Список использованной литературы

**1.** *Асанбекова Н.О., Барганалиева Ж.К. Практикум программирования на Python. -Бишкек, 2021 – 140 с.*

**2.** *Садырова М.Р., Барганалиева Ж.К. Python с нуля. – Бишкек, 2024 – 156 с.*

**3.** *Гэддис Т.* **Начинаем программировать на Python. –** 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с.

**4.** *Лучано Рамальо* **Python. К вершинам мастерства.** – М.: ДМК Пресс, 2016. – 768 с.

5. *Любанович Билл* Простой Python. Современный стиль программирования. – СПб.: Питер, 2016. – 480 с.: – (Серия «Бестсепперы O'Reilly»).

6. *Рейтц К., Шлюссер Т.* Автостопом по Python. – СПб.: Питер, 2017. – 336 с.: ил. – (Серия «Бестселлеры O'Reilly»).

**7.** *Свейгарт, Эл.* **Автоматизация рутиных задач с помощью Python: практическое руководство для начинающих.** Пер. с англ. – М.: Вильямс, 2016. – 592 с**.**

**8.** Барганалиева Ж.К., Садырова М.Р., Асанбекова Н.О., **"Pythondo tkinter менен интерактивдүү тест түзүү",** Вестник КГУ им. И. Арабаева. – 2024. – № 1, – С. 286-294

**Рецензент: кандидат технических наук, и.о. доцента Керимов У.Т.**