



ПРОЕКТ ILLUSTRATOR
MASTER ВЫПОЛНЕН
ЩЕГОЛЕВОЙ АНАСТАСИЕЙ

Идея проекта заключается в:

1. Создание программы для рисования, которая будет простой в использовании, но при этом функциональной и интересной для пользователей.
2. Проектирование пользовательского интерфейса: разработка удобного и понятного интерфейса, включающего меню, панель инструментов и холст для рисования.
3. Выбор инструментов и функций: определение набора инструментов для рисования (кисть, ластик, различные фигуры), а также дополнительных функций (сохранение изображений, отмена действий, очистка всего холста, открытие других изображений).



При реализации своего проекта я использовала следующие классы и функции:

- три основных класса(окна приложения):
 - **Illustrator_master** (главное окно)
 - **Drawing_window** (окно для рисования)
 - **db_Window** (окно с последними работами, с подключением к базе данных);
- Классы для построения фигур: **Circle, Line, Rectangle**;
- В классе **Drawing_window** в функции **closeEvent** я использовала **QMessageBox** для того, чтобы пользователь не забыл сохранить свою работу при выходе;
- В функции **open** я создала возможность открывать как изображения формата 'PNG', 'JPG', так и текстовые файлы формата 'TXT'
- Для выбора цвета, открытия и сохранения файлов я использовала различные диалоговые окна

База данных:

- Для работы с базой данных я использовала библиотеку **SQLAlchemy**
- подключение происходит в функции **global_init**
- в классе **Main_master** происходит создание таблицы в базе данных
- в функцию **add_main_master** добавляются новые значения(id, время сохранения, путь к файлу)
- для отображения данных я воспользовалась виджетом **QTableWidget**
- функция **open_run** создана для просмотра сохранённого изображения (при помощи библиотеки **Pillow**)

В ходе работы над проектом **Illustrator master** были решены следующие задачи:

- Разработка графического интерфейса пользователя с использованием библиотеки **PyQt5**.
- Создание основных инструментов рисования, таких как кисть, ластик, построение фигур.
- Добавление функций: сохранение изображений, отмена действий, очистка всего холста, открытие других изображений.

В результате был создан полноценный графический редактор, который позволяет пользователям создавать изображения на холсте. Графический интерфейс программы интуитивно понятен и удобен в использовании, а набор инструментов для рисования достаточен для базовых задач создания изображений.

При работе над проектом были использованы знания в области объектно-ориентированного программирования, что позволило структурировать код и сделать его более читаемым и понятным. В процессе работы над проектом были изучены и применены на практике различные методы и подходы к решению задач, что способствовало улучшению навыков программирования на Python.


```
...mirror object to mirror...
mirror_mod.mirror_object = ...

operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True
```

```
Selection at the end -add
mirror_ob.select= 1
modifier_ob.select=1
context.scene.objects.active
("Selected" + str(modifier_ob.name))
mirror_ob.select = 0
= bpy.context.selected_objects[0]
data.objects[one.name].select
print("please select exactly one object")
```

--- OPERATOR CLASSES ---

```
bpy.types.Operator):
    @classmethod
    def poll(cls, context):
        if context.active_object is not None:
            return True
        return False
```

Также я планирую дорабатывать свой проект в процессе обучения программирования на Python, чтобы реализовать ещё больше функций, помогающих пользователю работать с различными изображениями.

Проект **Illustrator master** может быть использован как для обучения основам программирования и графического дизайна, так и для создания простых графических редакторов с базовыми функциями.