Векторный редактор CLI v2.0

Простой векторный редактор с интерфейсом командной строки (CLI) на языке программирования Python.

Описание

Данный редактор реализует базовый функционал работы с различными фигурами:

* 2D фигуры: точка, отрезок, круг, квадрат, прямоугольник, овал, правильный многоугольник
* 3D фигуры: параллелепипед, тетраэдр

Редактор позволяет создавать фигуры, просматривать список созданных фигур, получать детальную информацию о фигурах, удалять фигуры, а также сохранять и загружать список фигур в файл через ввод команд в консоли. Интерфейс командной строки имеет цветное форматирование для улучшения читаемости и запросы подтверждения для критических операций.

Структура проекта

* **shape.py** - базовые абстрактные классы для всех фигур
* **shapes\_2d.py** - реализация 2D фигур
* **shapes\_3d.py** - реализация 3D фигур
* **main.py** - основной модуль с CLI интерфейсом

Запуск

bash

python3 main.py

или

bash

chmod +x main.py

./main.py

Доступные команды

* **help** - показать справку по командам
* **create <тип> <параметры>** - создать новую фигуру
* **list** - показать список всех фигур
* **info <id>** - показать информацию о фигуре
* **delete <id>** - удалить фигуру
* **clear** - удалить все фигуры
* **save <filename>** - сохранить фигуры в файл
* **load <filename>** - загрузить фигуры из файла
* **exit** - выйти из редактора

Примеры использования

Создание фигур

> create point 1 2 MyPoint

Создана фигура: MyPoint (1): Point(1.0, 2.0)

> create line 0 0 5 5 MyLine

Создана фигура: MyLine (2): Line((0.0, 0.0), (5.0, 5.0)), Length: 7.07

> create circle 3 3 2 MyCircle

Создана фигура: MyCircle (3): Circle(center=(3.0, 3.0), radius=2.0)

> create square 0 0 4 MySquare

Создана фигура: MySquare (4): Square(bottom\_left=(0.0, 0.0), side\_length=4.0)

> create rectangle 1 1 5 3 MyRectangle

Создана фигура: MyRectangle (5): Rectangle(bottom\_left=(1.0, 1.0), width=5.0, height=3.0)

> create oval 10 10 8 5 MyOval

Создана фигура: MyOval (6): Oval(center=(10.0, 10.0), radius\_x=8.0, radius\_y=5.0)

> create polygon 5 5 6 2 MyPolygon

Создана фигура: MyPolygon (7): RegularPolygon(center=(5.0, 5.0), sides=6, side\_length=2.0)

> create parallelepiped 0 0 0 3 4 5 MyParallelepiped

Создана фигура: MyParallelepiped (8): Parallelepiped(origin=(0.0, 0.0, 0.0), width=3.0, height=4.0, depth=5.0)

> create tetrahedron 1 1 1 3 MyTetrahedron

Создана фигура: MyTetrahedron (9): Tetrahedron(center=(1.0, 1.0, 1.0), edge\_length=3.0)

Просмотр списка фигур

> list

Список фигур:

1: MyPoint (1): Point(1.0, 2.0)

2: MyLine (2): Line((0.0, 0.0), (5.0, 5.0)), Length: 7.07

3: MyCircle (3): Circle(center=(3.0, 3.0), radius=2.0)

4: MySquare (4): Square(bottom\_left=(0.0, 0.0), side\_length=4.0)

5: MyRectangle (5): Rectangle(bottom\_left=(1.0, 1.0), width=5.0, height=3.0)

6: MyPolygon (6): RegularPolygon(center=(5.0, 5.0), sides=6, side\_length=2.0)

7: MyParallelepiped (7): Parallelepiped(origin=(0.0, 0.0, 0.0), width=3.0, height=4.0, depth=5.0)

8: MyTetrahedron (8): Tetrahedron(center=(1.0, 1.0, 1.0), edge\_length=3.0)

Получение информации о фигуре

> info 3

Информация о фигуре 3:

{

"id": 3,

"name": "MyCircle",

"type": "Circle",

"dimension": 2,

"center": {

"x": 3.0,

"y": 3.0

},

"radius": 2.0,

"area": 12.566370614359172,

"perimeter": 12.566370614359172

}

Удаление фигуры

> delete 2

Вы уверены, что хотите удалить фигуру: MyLine (2): Line((0.0, 0.0), (5.0, 5.0)), Length: 7.07? (y/n)

> y

Удалена фигура: MyLine (2): Line((0.0, 0.0), (5.0, 5.0)), Length: 7.07

Очистка всех фигур

> clear

Вы уверены, что хотите удалить все фигуры (7 шт.)? (y/n)

> y

Удалено фигур: 7

Сохранение фигур в файл

> create point 5 5 Point1

Создана фигура: Point1 (1): Point(5.0, 5.0)

> create oval 10 10 8 5 Oval1

Создана фигура: Oval1 (2): Oval(center=(10.0, 10.0), radius\_x=8.0, radius\_y=5.0)

> save my\_shapes

Фигуры успешно сохранены в файл 'my\_shapes.shapes'

Загрузка фигур из файла

> load my\_shapes

Фигуры успешно загружены из файла 'my\_shapes.shapes'

Загружено фигур: 2

> list

Список фигур:

1: Point1 (1): Point(5.0, 5.0)

2: Oval1 (2): Oval(center=(10.0, 10.0), radius\_x=8.0, radius\_y=5.0)

Расширение функциональности

Для добавления новых типов фигур необходимо:

1. Создать новый класс, наследующийся от Shape2D или Shape3D
2. Реализовать все абстрактные методы
3. Добавить новый тип фигуры в словарь shape\_types в классе VectorEditor