Студенческий научный электронный журнал
StudArctic Forum

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

http://saf.petrsu.ru **http://petrsu.ru**

УДК 004.021

Информационные технологии и телекоммуникации

ТУКУСЕР Данила Игоревич	бакалавриат, Петрозаводский государственный университет (Петрозаводск, Российская Федерация), tukuserdanila@gmail.com

Программная реализация методов начертательной геометрии на языке программирования Python

Научный руководитель:
Городничина Мария Юрьевна

Статья поступила: 27.04.2022;
Принята к публикации: 04.05.2022;
Размещена в сети: 01.10.2022.

Аннотация. В данной статье рассмотрена разработка программы для построения эпюров и решения задач по начертательной геометрии, описана программа, проведены разнообразные тесты производительности. Автором показан подход к постановке задачи, а также реализация сохранения эпюров, дано подробное описание функционала и внешнего вида приложения. Приложено схематичное представление архитектуры программы, объяснена причина выбора такой архитектуры и метода программирования, дано пояснение по объемности и расширяемости приложения. Также раскрыты перспективы дальнейшего развития.

Ключевые слова: программа, начертательная геометрия, эпюр, решение задач, машинная графика, Python

Для цитирования: Тукусер Д. И. Программная реализация методов начертательной геометрии на языке программирования Python // StudArctic Forum. 2022. Т. 7, № 3. С. 8–13.

Во время изучения курса начертательной геометрии и инженерной графики теоретический материал подкрепляется решением различный задач. Методы начертательной геометрии, являясь основой инженерной и машинной графики, широко используются в практическом русле. Эти методы реализованы во многих программах для черчения, расчета и в системах автоматизированного проектирования тем или иным образом.

Для программной реализации методов начертательной геометрии был выбран высокоуровневый язык программирования Python. Для написания графического интерфейса существует множество разных решений. Чтобы создать графический интерфейс программы был выбран Kivy – бесплатная графическая библиотека для Python с открытым исходным кодом. Для сборки проекта в исполняемый (формата .exe) файл, который можно запустить без установок дополнительного программного обеспечения, будет использована библиотека PyInstaller. При этом программа будет написана с помощью объективно-ориентированного программирования (далее по тексту ООП), а значит обладать возможностью добавления любого количества новых функций, что создает широкие возможности для дальнейшего развития.

Основная цель работы – написать программу на языке программирования Python, для решения задач по начертательной геометрии и работы с эпюрами.

Для достижения цели были поставлены следующие главные задачи:

Разработать программу для построения и работы с эпюрами, реализовать базовые функции программы для удобной работы в программе.
Определить список основных задач, решаемых в курсе начертательной геометрии.
Разработать алгоритмы для решения задач и внедрить их.
Произвести тесты производительность программы.

Для разработки программы на первом этапе работы, необходимо понимать, какие функции нам необходимы для построения эпюра. Самые основные геометрические объекты эпюра – точка и прямая, их необходимо строить, редактировать и удалять. Каждая точка имеет следующие параметры: координаты x, y, z и название. Для удобства условимся, что в данной программе прямая будет задаваться принадлежащим ей отрезком, который будет задан по двум точкам. Для работы с точкой и отрезком будут заданы все необходимые для работы функции.

Помимо функции, которые связанны с построениями, для удобства работы в программе следует реализовать функции масштабирования, перемещения и, соответственно, обратные им – функции установления стандартного масштаба и стандартного положения осей координат.

Для выбора решаемой задачи был разработан конструктор задач – это специальное окно, в котором, основываясь на выборе из предложенных программой вариантов, пользователем выбирается задача. Его работа устроена следующим образом: пользователем выбирается ключевое слово (тип задачи), затем выбирается задача и объект этой задачи, при необходимости программа запросит еще некоторое количество данных.

В целях расширения возможностей программы нужны функции сохранения и последующего открытия эпюров. Для этого

разработан специальный виджет для открытия файлов (рисунок 3, 1). С его помощью можно найти файл сохранения в директории с сохранениями saves или в любой другой папке. Сохранения реализованы с помощью текстового файла формата .txt, где данные эпюра отформатированы определенным образом. По умолчанию названием файла устанавливается время сохранения, но пользователь может ввести любое желанное имя. В начале файла записываются параметры точек и прямые, которые строит пользователь, а затем поочередно вписываются для каждой решенной задачи параметры всех объектов этой задачи. Форматирование данных идет по следующим правилам: параметры одного объекта разделяются запятой, разные объекты одного типа разделяются точкой с запятой ‘;’, разные типы разделяются между собой вертикальной чертой ‘ | ’, данные, введенные пользователем, и решения задач разделяются знаком ‘$’, а разные задачи разделяются знаком процента ‘%’. При открытии данные сохранения переводятся в используемый программой формат и автоматически строятся на экране. Если файл сохранения имеет неверный формат, то программа не будет его открывать.

Пример такого сохранения: 100.0,120.0,280.0,E;300.0,230.0,115.0,K|1,0$ - при открытии этой записи на экране появится: точка E(100, 120, 280), точка K(300, 230, 115) и отрезок соединяющий точку E и K. При открытии любого сохранения вне зависимости от масштаба и размера экрана будут сохранено положение всех объектов. Ввиду простоты формата можно обрабатывать большие файлы, например сохранение, содержащее в себе 10000 символов, что в среднем эквивалентно 7 задачам.

Конечная реализация графического интерфейса приложения с выбранными выше функциями имеет следующий вид (Рисунок 1):

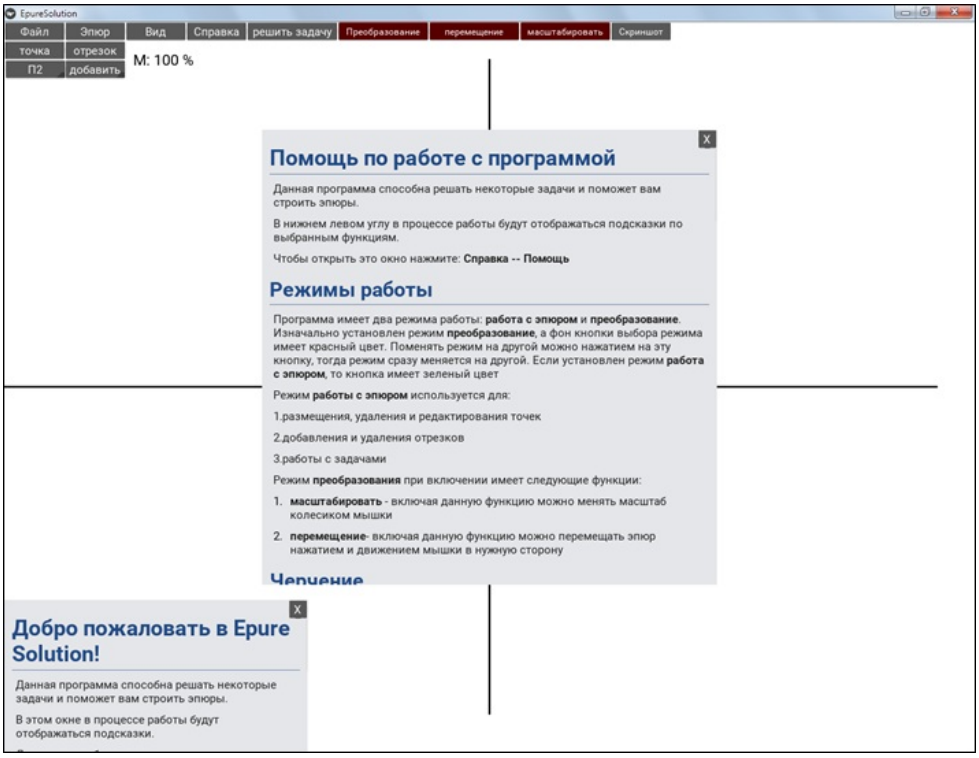

Рис. 1. Главное окно приложения

При открытии программы окно визуально делится на несколько частей: инструментальную панель в верхней части программы; окно помощи в нижнем левом углу; окно справки по центру и рабочее поле с координатыми осями. Также в программе реализованны еще несколько дополнительных окон, которые вызываются из раскрывающихся списков инструментальной панели. Среди этих окон (Рисунок 2): 1 – окно открытия файла сохранения; 2 – конструктор задач; 3 – окно сохранения файла.

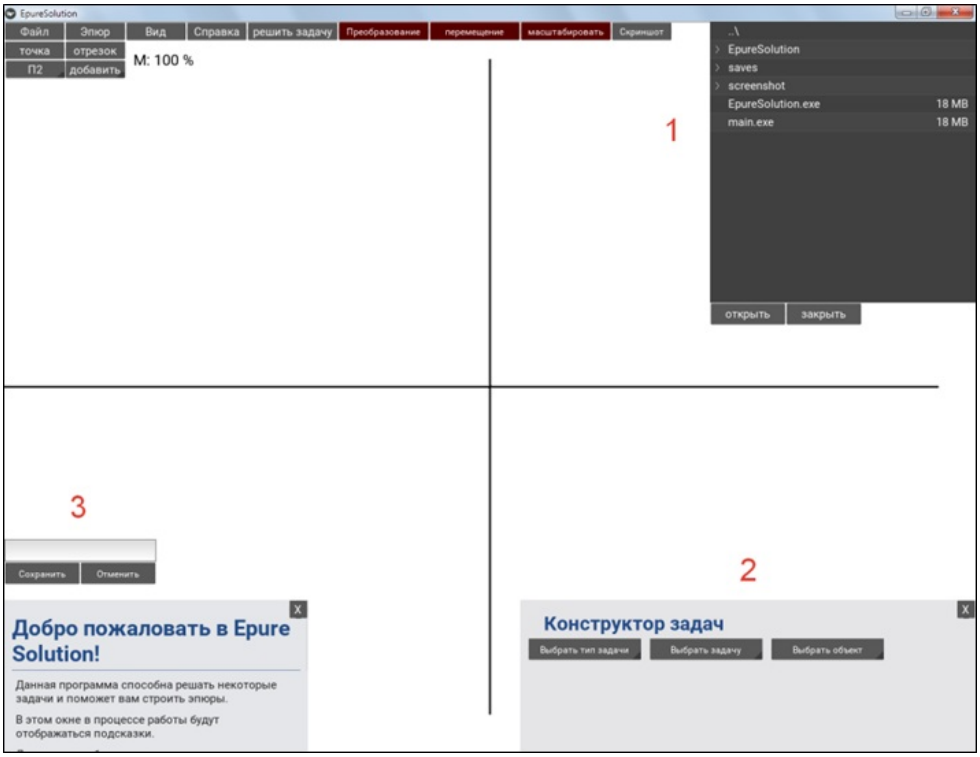

Рис. 2. Дополнительные окна

Для организации работы такого крупного приложения с большим количеством функций, а также в целях упрощения написания, поддержки и дальнейшего развития программы была применена модульная архитектура программы и ООП - рисунок 3.

В итоге программа состоит из 9 модулей, 5092 строчек программного кода. Такая объёмность объясняется тем, что алгоритмы решений включают в себя достаточно много операций, а также тем, что вся программа написана объективно-ориентировано и разбита на модули, что позволяет в дальнейшем расширять её функционал. Всего 15 алгоритмов, которые способны решить 21 задачу, занимают 1700 строк кода, самый компактный из них – алгоритм определения положения точки относительно прямой, занимает 27 строчек кода, а самый крупный алгоритм занимает 277 строчек кода.

Алгоритмы решения задач - очень объемная тема, поэтому будет дано краткое описание. В своем большинстве алгоритм решения можно разделить на несколько частей: анализ, решение и вывод. Необходимость в анализе обусловлена большим количеством частных случаев, особенностей расположения геометрических объектов и работы геометрических функций. Само решение задач выполняется с поправками на особенности расположения и работу вспомогательных функций. В отличие от реальных построений и решений, все величины и координаты находятся с помощью вычислений, а не построений, например, так находятся пересечения прямых, расстояния, перпендикуляры и т.д., а уже после этих вычислений по полученным координатам проводятся все построения. Результатом решения является массив из точек, отрезков и текста, который возвращаются в главное окно программы и с помощью методов обновления отображается на экране.

Подводя итог, стоит отметить, что, мы решаем задачи наоборот, т.е. вместо нахождения какой-либо величины с помощью построений и последующего измерения мы строим эпюр по рассчитанным алгоритмами координатам. Главные плюсы такого метода – высокая точность и возможность быстрого получения любой величины, размера, угла и т.п. эпюра. Из минусов можно выделить достаточную сложность написания алгоритмов и их отладки, поскольку мы должны работать в основном с числами и геометрическими функциями.

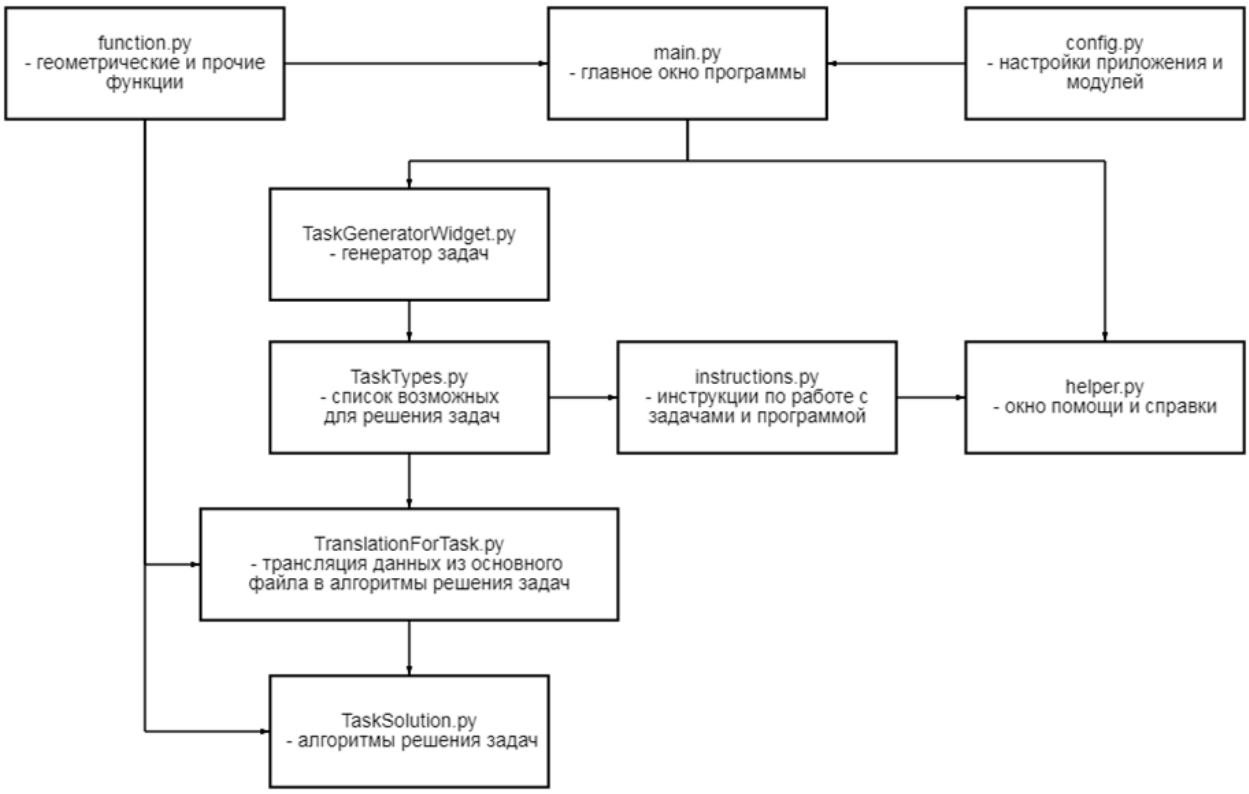

Рис. 3. Архитектура работы программы

Для оценки производительности программы было проведено несколько тестов, направленных на изучение времени обновления экрана. Проведено четыре теста на скорость обновления: точки, прямой, точки и прямой, некоторой задачи. Результаты занесены в Таблицу 1.

Из таблицы можно сделать вывод, что работа с разными типами объектов не оказывает влияния на время обновления экрана. Была выявлена проблема с производительностью. Оказалось, что на обновление одной точки требуется в среднем в десять раз больше времени, чем на обновление отрезка.

Тест скорости обновления задач был проведен на примере задачи по нахождению расстояния между точкой и плоскостью (Рисунок 4). Как видно из Таблицы 1, не смотря на то, что решение задачи включает в себя 31 объект, время для обновления одной задачи достаточно мало – всего 0,0158 секунды, что достаточно приемлемо. Из данных таблицы следует, что оптимально будет решать на одном эпюре не более 5 задач, чтобы сохранить приемлемую частоту обновления экрана.

Выявленная проблема с производительностью вызвана тем, что точка имеет название, которое является объектом библиотеки kivy класса Label, а значит отображение этого класса наиболее затратная операция при обновлении. Следовательно, в будущем для оптимизации, стоит заменить этот объект на другой менее требовательный, тогда можно уменьшить время обновления в несколько раз.

Таблица

Результаты тестов производительности

Количество точек	Время загрузки, с	Количество прямых	Время загрузки, с	Количество точек и прямых	Время загрузки, с	Количество задач	Время, с
10	0,021	10	0,0012	10	0,023	1	0,02
50	0,1	50	0,011	50	0,11	5	0,07
100	0,2	100	0,033	100	0,233	10	0,14
250	0,58	250	0,07	250	0,65	15	0,23
500	1,37	500	0,08	500	1,45	20	0,31
Ср. время для одной точки, с	0,0022	Ср. время для одного отрезка, с	0,00022	Ср. время, для отрезка и точки, с	0,0024	Ср. время для одной задачи, с.	0,0158

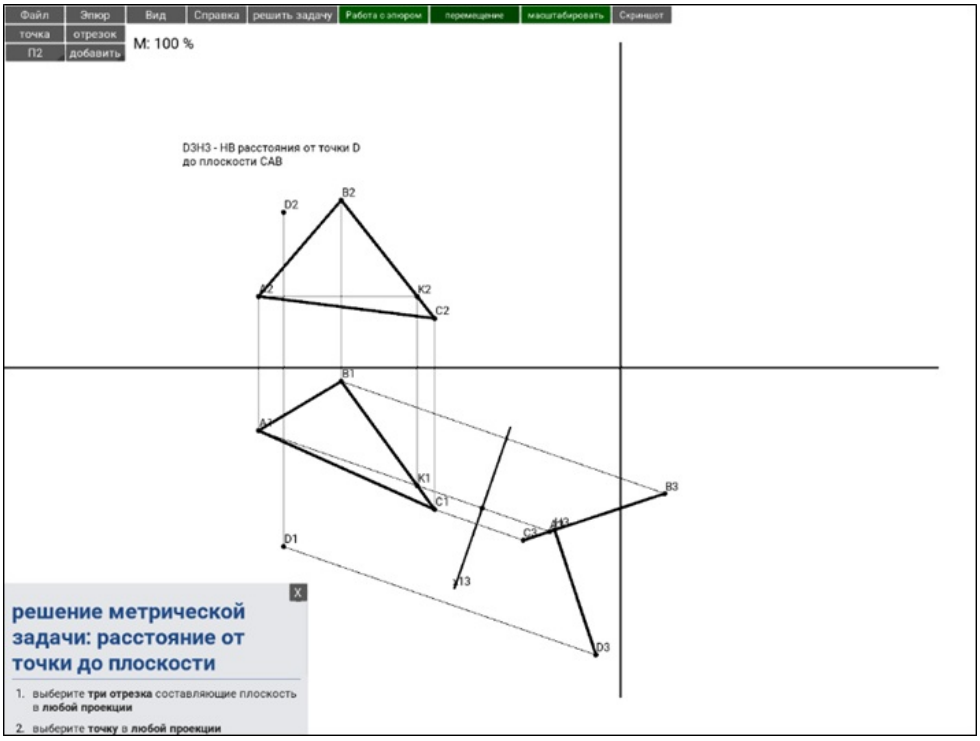

Рис. 4. Пример решения задачи по нахождению расстояния между точкой и плоскостью

Данная программа применима для построения эпюров, решения метрических, позиционных и смешанных задач. В ходе работы были реализованы: все необходимые функции для решения задач, черчения и работы с эпюрами; система сохранений; алгоритмы решения задач. Выделены общие проблемы представленной реализации и их возможные решения. Использование модульной конструкции и применение ООП позволяет в будущем добавлять новые функции и задачи, а также оптимизировать некоторые функции. Развивать программу можно в разных направлениях: от создания приложения для проверки преподавателями студенческих работ до реализации приложения по визуализации и решения задач в многомерном пространстве.

СПИСОК ЛИТЕРАТУРЫ

Глоговский В. В. Начертательная геометрия на алгоритмической основе. Львов : «Вища школа», издательство при Львовском университете, 1978. 148 с.

Желоние Е. И. Алгоритмы решения метрических задач : учебное пособие. Йошкар-Ола : МарПИ, 1989. 88 с.

Information Technology and Telecommunications

Danila TUKUSER	bachelor's degree, Petrozavodsk State University (Petrozavodsk, Russian Federation), tukuserdanila@gmail.com

Software implementation of descriptive geometry methods in the Python programming language

Scientific adviser:
Maria Y. Gorodnichyna
Paper submitted on: 04/27/2022;
Accepted on: 05/04/2022;
Published online on: 10/01/2022.

Abstract. This article discusses the development of a program for constructing diagrams and solving problems in descriptive geometry, describes the program, and conducted various performance tests. The author shows an approach to setting the problem, as well as the implementation of saving diagrams, a detailed description of the functionality and appearance of the application. A schematic representation of the program architecture is attached, the reason for choosing such an architecture and programming method is explained, an explanation is given for the volume and extensibility of the application. The prospects for further development are also disclosed
Keywords: program, descriptive geometry, diagrams, tasks solving, computer graphics, Python

For citation: Tukuser, D. Software implementation of descriptive geometry methods in the Python programming language. StudArctic Forum. 2022, 7 (3): 8–13.

REFERENCES

Glogovskij, V. V. Nachertatel'naya geometriya na algoritmicheskoj osnove. L'vov: "Vishha shkola", izdatel'stvo pri L'vovskom universitete, 1978. 148 s.

Zhelonie, E. I. Algoritmy' resheniya metricheskix zadach: Uchebnoe posobie. Joshkar-Oia : MarPI, 1989. 88 s.