МИНОБРНАУКИ РОССИИ

ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ЭКОНОМИКИ И СЕРВИСА

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

ОТЧЁТ

ПО УЧЕБНОЙ ПРАКТИКЕ ПО ПОЛУЧЕНИЮ ПЕРВИЧНЫХ ПРОФЕССИОНАЛЬНЫХ УМЕНИЙ И НАВЫКОВ

|  |  |  |
| --- | --- | --- |
| Студент  гр. БИС–18–01 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | А.Д. Шнейдер |
| Руководитель канд. экон. наук, доцент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | В.В. Ивин |

Владивосток 2020

Задание

Используя объектно-ориентированную методологию разработки программных продуктов, создайте кроссплатформенное приложение с оконным интерфейсом, реализующее алгоритм в соответствии с вариантом задания.

Используя произвольные язык программирования и среду разработки создайте программу, которая отображает на экране монитора график кривой или поверхности (в соответствии с вариантом задания) в декартовой и полярной системах координат с центром в центре экрана монитора (окна или иной прямоугольной области экрана). При изменении размеров окна, график и все его атрибуты (координатная сетка, метки на шкале, подписи и т.д.) должны автоматически масштабироваться.

Параметры уравнения кривой или поверхности должны вводиться в специально отведённые ячейки экранной формы программы.

Используя результаты предыдущего задания создайте анимацию примитива, движущегося по траектории построенной кривой. В качестве примитива используется закрашенная окружность красного цвета радиуса r>2.

Содержание

[Введение 3](#_Toc46388259)

[1 Техническое задание 4](#_Toc46388260)

[1.1 Формулы для построения циссоиды Диоклеса 4](#_Toc46388261)

[1.2 Анализ технического анализа 4](#_Toc46388262)

[2 Выбор языка и среды разработки 4](#_Toc46388263)

[2.1 Выбор языка 4](#_Toc46388264)

[2.2 Выбор среды разработки и библиотеки 4](#_Toc46388265)

[3 Описание принципа работы программы 5](#_Toc46388266)

[3.1 Графический интерфейс 5](#_Toc46388267)

[3.2 Описание принципов работы программы 6](#_Toc46388268)

[4 Проверка работы программы 9](#_Toc46388269)

[4.1 Тестирование работы 9](#_Toc46388270)

[4.2 Тестирование работы при неправильных данных 11](#_Toc46388271)

[5 Инструкция для работы с программой 12](#_Toc46388272)

[5.1 Основной интерфейс 12](#_Toc46388273)

[5.2 Боковая панель настроек и управления 13](#_Toc46388274)

[Заключение 14](#_Toc46388275)

[Список использованных источников 15](#_Toc46388276)

[Приложение А Блок-схема программы 16](#_Toc46388277)

[Приложение B Диаграмма классов 17](#_Toc46388278)

Введение

**Целью** выполнения практики по дисциплине «Учебная практика по получению первичных общекультурных компетенций» является закрепление и углубление знаний, полученных при теоретическом обучении, а также приобретение знаний во время непосредственного выполнения и решения практических задач в области информационных систем и технологий.

**Основной задачей** учебной практики является написание программы, осуществляющей построение графика на основе формулы, предоставленной студенту. Программа должна полностью соответствовать специфическим требованиям задания, таким как отображение и масштабирование в реальном времени, возможность интерактивно изменять параметры функции.

Отображение графика в реальном времени помогает наглядному пониманию работы функции, путем интерактивного изменения параметров пользователем и моментального отображения результата.

Для выполнения практического задания был дан график циссоиды Диоклеса.

1. Техническое задание
   1. Формулы для построения циссоиды Диоклеса

Циссоида Диоклеса - плоская алгебраическая кривая 3-го порядка. Для построения графика на компьютере следует строить его по точкам, имеющим координаты x и y. Для расчёта этих координат удобно использовать частные формулы (1) и (2), где a - радиус вспомогательной окружности.

|  |  |
| --- | --- |
|  | (1) |
|  |  |
|  | (2) |

Для циссоиды диапазон t зависит от параметра a, и является углом между точкой, образующей кривую, центром координат и осью OX.

* 1. Анализ технического анализа

Для вывода графика необходимо проверять правильность вводимых данных, так как график может выходить за доступную для просмотра координатную сетку или рассчитываться некорректно.

1. Выбор языка и среды разработки
   1. Выбор языка

Основным языком для написания программы был выбран Python. Это удобный, кроссплатформенный, объектно-ориентированный, хорошо читабельный язык программирования. У этого языка имеются разнообразные библиотеки для работы с графикой, с помощью которых можно легко изобразить любой график, написав небольшое количество строк кода.

* 1. Выбор среды разработки и библиотеки

В качестве среды разработки был выбран редактор кода Visual Studio Code, которые не сильно уступает полноценным IDE. Он имеет огромное количество подключаемых расширений, что позволяют работать с разными языками и библиотеками, настраивать интерфейс и разработку удобной для разработчика.

Для написания программы был выбран популярный мощный фреймворк Qt. Он отлично подходит для разработки кроссплатформенных приложений с графическим интерфейсом. В первую очередь фреймворк работает с языком C++, но также имеет «привязки» и к другим языкам, в том числе и к Python. К нему есть подробная документация и множество книг. Версия Qt выбрана 5.14 как актуальная на момент написания программы.

Для построения графического интерфейса использовался Qt Designer, файлы которого генерируются в код Python с помощью UIC. Этот код выступал в роли вспомогательного для написания основного кода программы.

Для работы с Qt на Python существует два основных набора расширений – PyQt и PySide. Они имеют совсем небольшое число различий, поэтому не имеет значения какой использовать. Но так как PySide является официальным решением от Qt, то именно он был выбран для реализации программы. Приложение написано с использованием PySide2 5.15.0 и Python 3.8.3 (для работы требуется Python 3.5+).

Общая блок-схема программы представлена на рисунке A.1. Для отрисовки нет необходимости строить отдельно график в декартовой и полярной системе координат.

1. Описание принципа работы программы
   1. Графический интерфейс

В первую очередь была составлена боковая панель, включающая в себе основные элементы управления: ввод значения параметра функции, кнопка обновления графика, переключение на полярную систему координат, запуск анимации. Все элементы размещены в соответствии с рисунком 3.1.

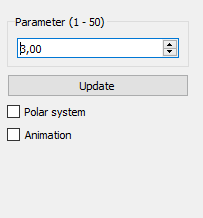


Рисунок 3.1 – Интерфейс боковой панели

По умолчанию, параметру a присвоено значение 3, так как оно является оптимальным. Для ввода параметра используется QDoubleSpinBox. С его помощью двумя правыми кнопками можно уменьшать или увеличивать значение параметра на единицу, или ввести самому в текстовое поле с точностью до 2 знака. Значение параметра ограниченно в диапазоне от 1 до 50.

Далее был создан виджет с областью рисования графика. Виджет занимает большую часть окна и располагается слева. Полный интерфейс программы представлен на рисунке 3.2.

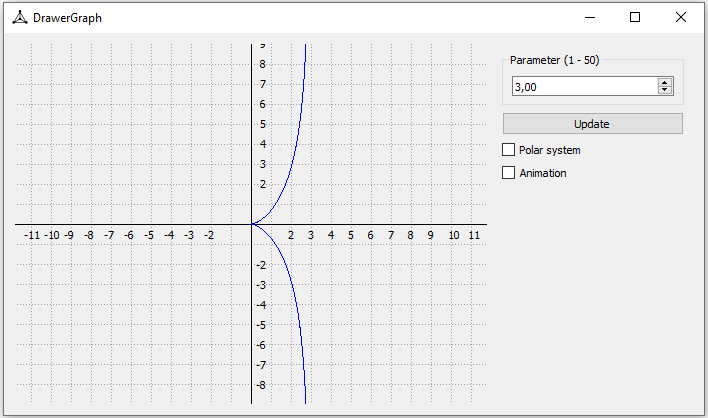


Рисунок 3.2 – Главное окно программы

Отрисовка на виджете происходит в момент вызова события paintEvent. Рисование происходит с помощью QPainter непосредственно на QWidget.

* 1. Описание принципов работы программы

Структура и диаграмма классов программы представлена в Приложении B. Создание интерфейса осуществляется в классе MainWindow, там же обрабатывается ввод нового параметра. При нажатии на кнопку «Update» происходит вызов метода, представленный на рисунке 3.3, передающий новый параметр классу GraphWidget, который занимается созданием виджета с областью рисования.

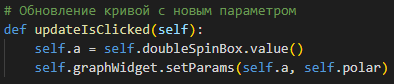


Рисунок 3.3 – Метод отправки параметр кривой и обновления графика

Далее этот параметр передается классу Drawer, который запускает отрисовку сетки и кривой. Осуществляется это с помощью метода drawPlot, представленного на рисунке 3.4.

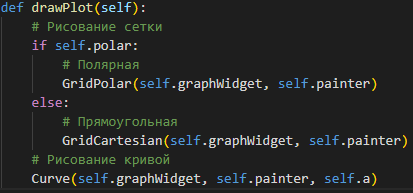


Рисунок 3.4 – Метод запуска отрисовки сетки и кривой

В зависимости от выбранной системы координат, создается соответствующий экземпляр класса, рисующий сетку: GridPolar для полярной системы, GridCartesian – прямоугольной.

Также Drawer создается экземпляр класса Curve, который рисуют кривую. Для этого в Curve имеется метод draw, представленный на рисунке 3.5.

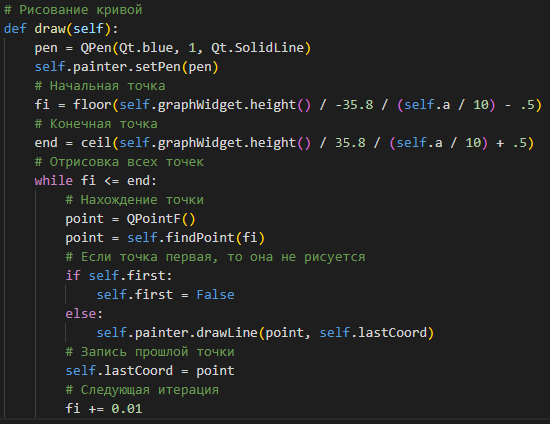


Рисунок 3.5 – Метод отрисовки кривой

Для рисования сначала высчитываются начальная и конечная точки рисования, в зависимости от размеров окна. Затем запускается цикл отрисовки, в ходе которого происходит нахождение новой точки, сохранение старой и рисование линии между ними, при этом самая первая найденная точка игнорируется, так как прошлая еще не найдена. Для нахождение новой точки вызывается метод findPoint, которому передается текущая итерация. Реализация этого метода представлена на рисунке 3.6.

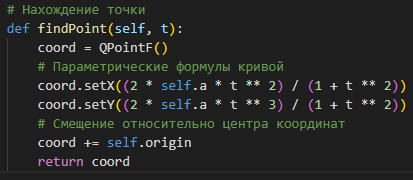


Рисунок 3.6 – Метод нахождения новой точки

В программе присутствует анимация движения примитива в форме красного эллипса по построенной кривой. Для запуска анимации необходимо отметить соответствующий чек бокс в интерфейсе программы. Отрисовкой занимается сам виджет с областью рисования. Для этого существует метод startAnimation, представленный на рисунке 3.7.

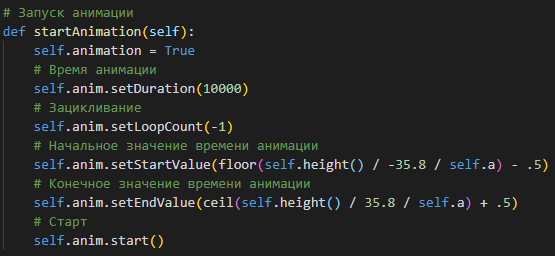


Рисунок 3.7 – Метод запуска анимации

Во время старта анимации происходит процесс нахождения точек, аналогичный рисованию кривой. По координатам найденной точки рисуется красный эллипс. Если анимация активна и при этом происходит масштабирование окна, то в событии resizeEvent начальная и конечная точки анимации изменяются в зависимости от размеров нового окна, при этом текущая анимация ставится на паузу и затем возобновляется, сделано это для того, чтобы анимация продолжалась корректно после переопределения начальной и конечной точек. Событие представлено на рисунке 3.8.

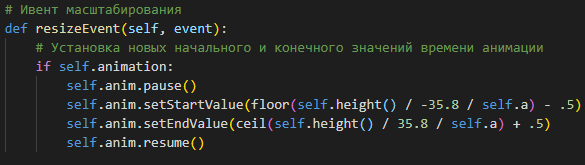


Рисунок 3.8 – Событие масштабирования

При обновлении размеров окна, при изменении параметров, вызове кнопки «Update» или вызове метода repaint вызывается событие paintEvent, которое представлено на рисунке 3.9.

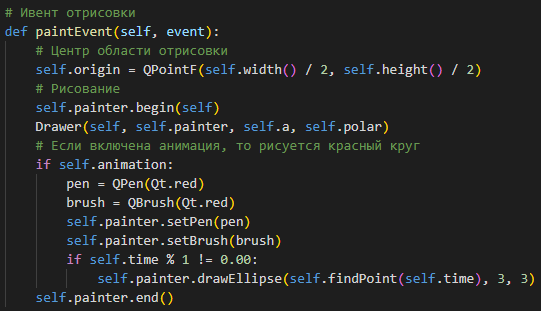


Рисунок 3.9 – Событие рисования

При вызове события запускается рисование, передаются параметры в класс Drawer и, если включена анимация, рисует красный эллипс.

1. Проверка работы программы
   1. Тестирование работы

При открытии программы график автоматически строится по стандартному значению. Пример нарисованного графика представлен на рисунке 4.1

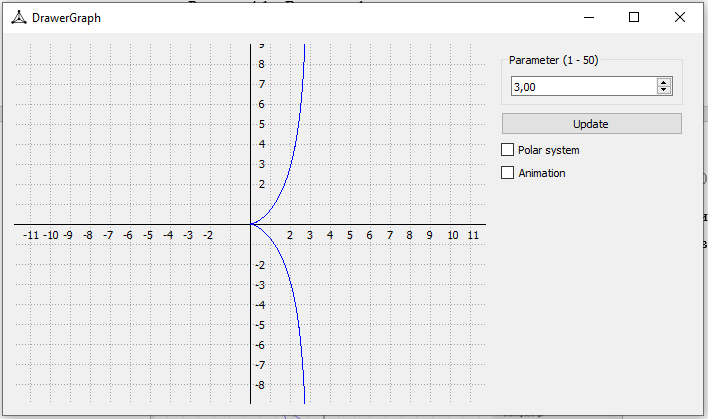


Рисунок 4.1 – Вывод графика при стандартных значениях

При изменении параметра вручную и нажатии на кнопку «Update», график изменяется по новому параметру в соответствие с рисунком 4.2.

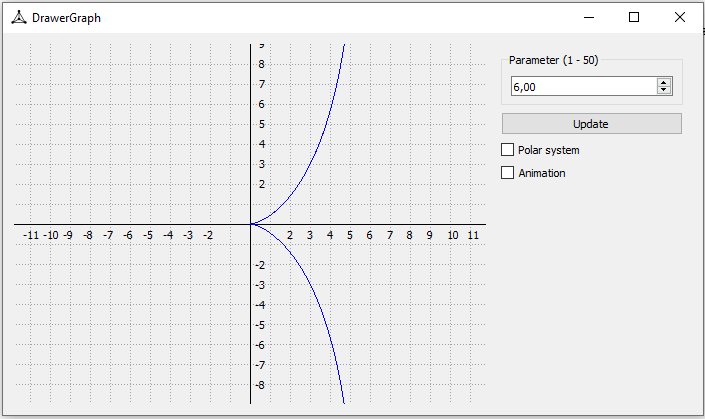


Рисунок 4.2 – Вывод графика при изменении значения параметра

При переключении типа сетки на полярную систему координат, вывод графика работает исправно в соответствие с рисунком 4.3.

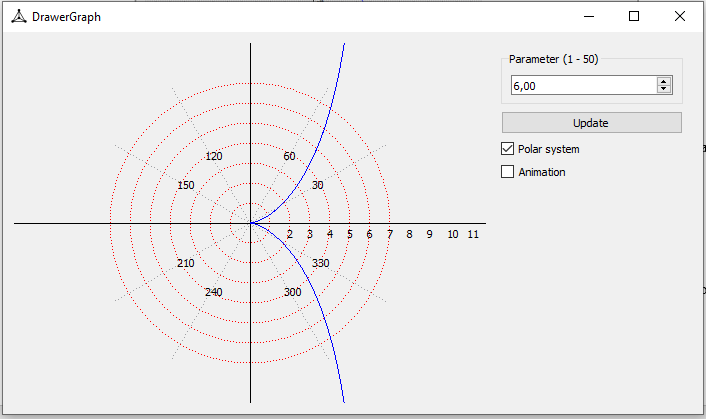


Рисунок 4.3 – Вывод графика при смене координатной сетки

При включении анимации появляется маленький красный эллипс, который движется по кривой в соответствие с рисунком 4.4.

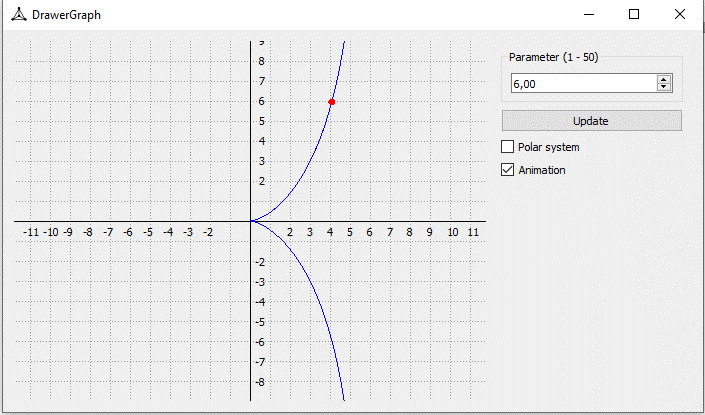


Рисунок 4.4 – Вывод графика с анимацией

Анимация делает один проход по всей кривой, затем повторяется. При смене координатной сетки и нажатии на кнопку «Update» анимация перезапускается.

* 1. Тестирование работы при неправильных данных

Единственным уязвимым местом, которое можно нарушить работу программы, является ввод нового параметра. Пользователь можно ввести отрицательные значения или некорректные символы.

Защита от таких неправильных данных осуществляется стандартными средствами фреймворка Qt при использовании QDoubleSpinBox. Ввести туда получится только числа, включая дробные, а также не даст выйти за пределы диапазона, заданного программно. Программа просто не даст ввести число меньшее 1 или большее 50. Пример ввода некорректных данных представлен на рисунке 4.5.

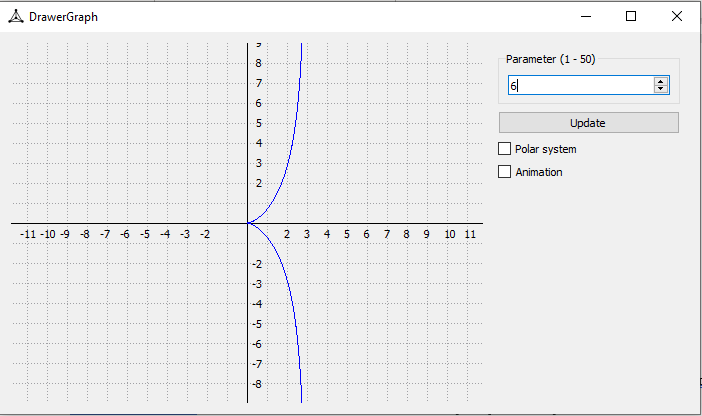


Рисунок 4.5 – Пример ввода некорректных данных

При попытке ввести параметр равный 60, программа просто не дает поставить вторую цифру.

1. Инструкция для работы с программой
   1. Основной интерфейс

Интерфейс программы делится на две части: область рисования и боковая панель, расположенные слева и справа соответственно. Расположение представлено на рисунке 5.1.

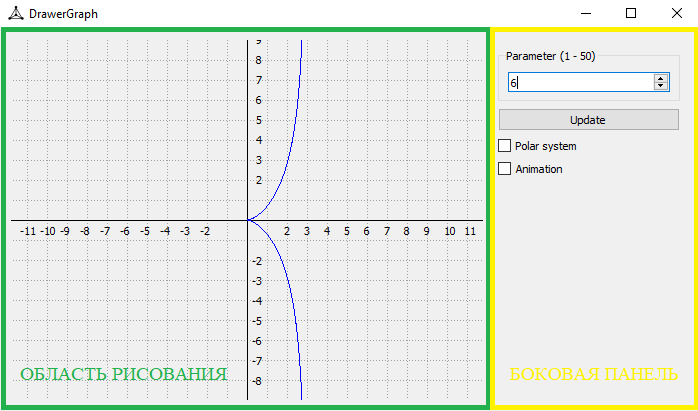


Рисунок 5.1 – Основной интерфейс программы

Область рисования содержит график кривой, отображаемый в зависимости от выбранного параметра и настроек в боковой панели.

* 1. Боковая панель настроек и управления

На боковой панели располагаются: поле для ввода параметра a, кнопка «Update» для обновления графика, чек бокс «polar system» для включения полярной сетки и чек бокс «animation» для включения анимации. Панель представлена на рисунке 5.2.

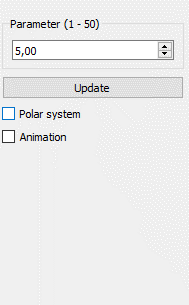


Рисунок 5.2 – Боковая панель

Все элементы боковой панели подписаны, поэтому в ней можно легко ориентироваться.

Заключение

В процессе прохождения практики была разработана программа для отображения графика циссоиды Диоклеса с использованием объектно-ориентированной методологии.

В процессе создания программы были освоены принципы разработки ООП приложений, имеющих графический интерфейс. Рассмотрены способы создания и вывода графического изображения в реальном времени.

Благодаря прохождению курса учебной практики были освоены принципы работы с нестандартными библиотеками, объектно-ориентированными языками программирования, разработкой графических интерфейсов.

Список использованных источников

1. Официальная документация Python [Электронный ресурс] // Python – Режим доступа: https://www.python.org/doc/
2. Официальная документация библиотеки Qt [Электронный ресурс] // Qt Reference – Режим доступа: https://doc.qt.io/qtforpython/index.html
3. Информация о формуле циссоиды Диокла [Электронный ресурс] // Wikipedia – Режим доступа: https://ru.wikipedia.org/wiki/Циссоида\_Диокла
4. СК-СТО-ТР-04-1.005-2015. Требования к оформлению текстовой части выпускаемых квалификационных работ, курсовых работ (проектов), рефератов, контрольных работ, отчетов по практикам, лабораторным работам [Электронный ресурс] // Библиотека ВГУЭС – Режим доступа: http://lib.vvsu.ru/russian/doc/СК-СТО-ТР-04-1.005-2015.pdf

Приложение А  
Блок-схема программы

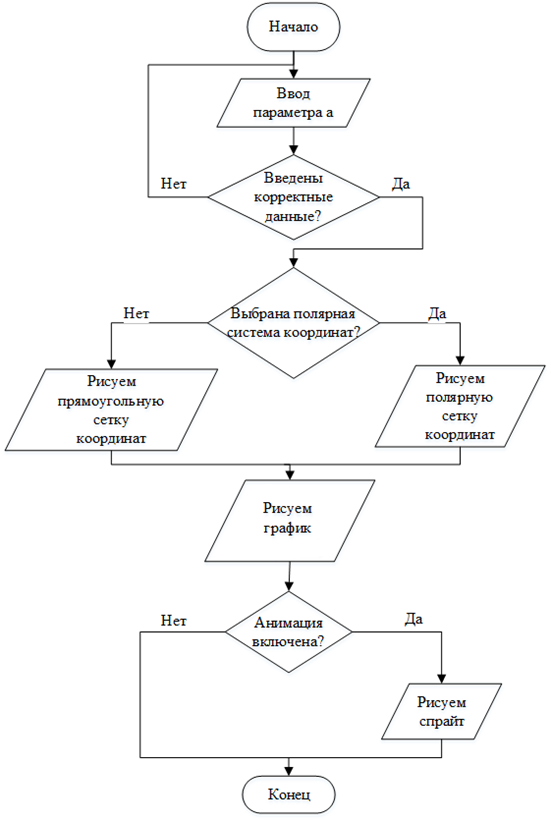


Рисунок A.1 – Блок-схема программы

Приложение B  
Диаграмма классов

