

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт Космических и информационных технологий
Кафедра вычислительной техники

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на разработку программы для визуализации теории графов

Преподаватель

Васильев В.С.

подпись

Студент КИ19-08Б, 031940024

Звягин С.А.

подпись

Красноярск 2020

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Основания для разработки.....	4
2 Назначение разработки.....	4
3 Требования к программе.....	4
3.1 Требования к функциональным характеристикам.....	4
3.2 Требования к составу выполняемых функций.....	5
3.3 Условия эксплуатации.....	7
3.4 Требования к составу и параметрам технических средств.....	7
3.5 Требования к информационной и программной совместимости.....	7
4 Требования к программной документации.....	7
5 Стадии и этапы разработки.....	7
6 Порядок контроля и приемки.....	8
ПРИЛОЖЕНИЕ А Макеты пользовательского интерфейса.....	9

ВВЕДЕНИЕ

Оконное приложение «GraphHelper» предназначено для нужд дискретной математики, а в частности для теории графов. Данное приложение способно визуализировать основные положения теории графов, таких как построение различных графов и произведение ряда операций над ними.

Областью применения программного продукта «GraphHelper» является сфера образования, так, например, «GraphHelper» может стать незаменимым помощником при освоении целого блока курса «Дискретная математика».

1 Основания для разработки

Разработка ведётся в рамках освоения автором дисциплин Б1.В.ОД.7 «Междисциплинарный курсовой проект базового уровня» и Б1.В.03 «Основы проектирования программного обеспечения», согласно учебному плану по программе бакалавриата (2019 год набора, очная форма) направления 09.03.01 Информатика и вычислительная техника.

2 Назначение разработки

Программный продукт «GraphHelper» обеспечивает функциональность и наглядность основных положений теории графов по средствам их визуализации на области «холста» пользовательского интерфейса.

Таким образом, функционал «GraphHelper» включает в себя следующие пункты:

- Создание и удаление вершин, рёбер, петель, дуг графа;
- Слияние вершин простого графа;
- Стягивание ребра простого графа;
- Дополнение простого графа;
- Объединение графов;
- Пересечение графов;
- Алгоритмы над графами;
- Сохранение графа;
- Возможность отключения текстовых пометок на области холста;
- Определение типа графа;
- Составление матриц, ассоциативных с графом.

3 Требования к программе

3.1 Требования к функциональным характеристикам

3.1.1 Требования к составу выполняемых функций: Программа «GraphHelper» должна выполнять следующие функции, для обеспечения собственной надобности:

- а. Удобство пользовательского интерфейса;
- б. Удобство импорта и экспорта графов в файл;

в. Наглядность результата выполнения операций над графом.

3.1.2 Требования к организации входных и выходных данных: Программа «GraphHelper» подразумевает использование в качестве входных данных либо построение графа на области, либо задание графа с помощью характеризующей его матрицы, либо загрузку графа из файла, помимо этого в качестве входных данных можно подразумевать и отдаваемые пользователем команды по реализации тех или иных операций над графом. На рисунке 1 представлена организация входных и выходных данных:

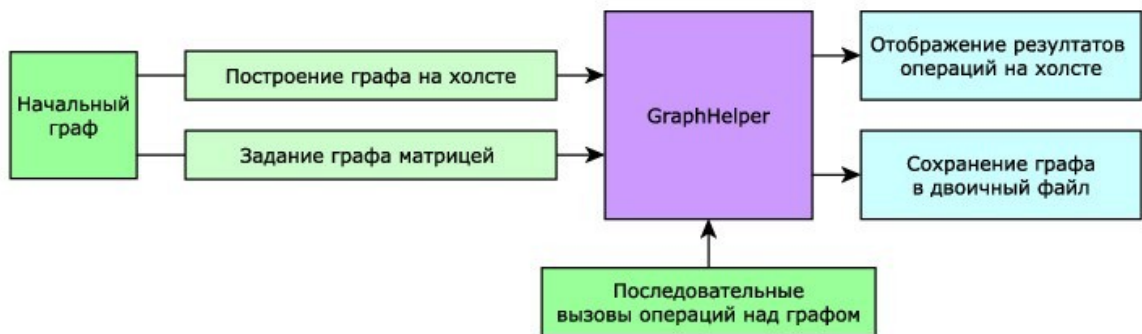


Рисунок 1 – Организация входных и выходных данных

В качестве выходных данных представляется отображение результатов операций на интерактивной области холста с возможностью последующего сохранения графа в двоичный файл, структура которого изображена на рисунке 2:

Код типа графа	кол-во вершин(n)	кол-во линий(k)	Вершина 1		...	Вершина n		Линия 1		...	Линия k	
			Коорд. X	Коорд. Y		Коорд. X	Коорд. Y	Вершина $i \in [1, n]$	Вершина $j \in [1, n]$		Вершина $i \in [1, n]$	Вершина $j \in [1, n]$

Рисунок 2 – Структура двоичного файла

3.2 Требования к составу выполняемых функций

Требования к составу выполняемых функций отражены через UML диаграмму прецедентов, изображённой на рисунке 3, на следующей странице:

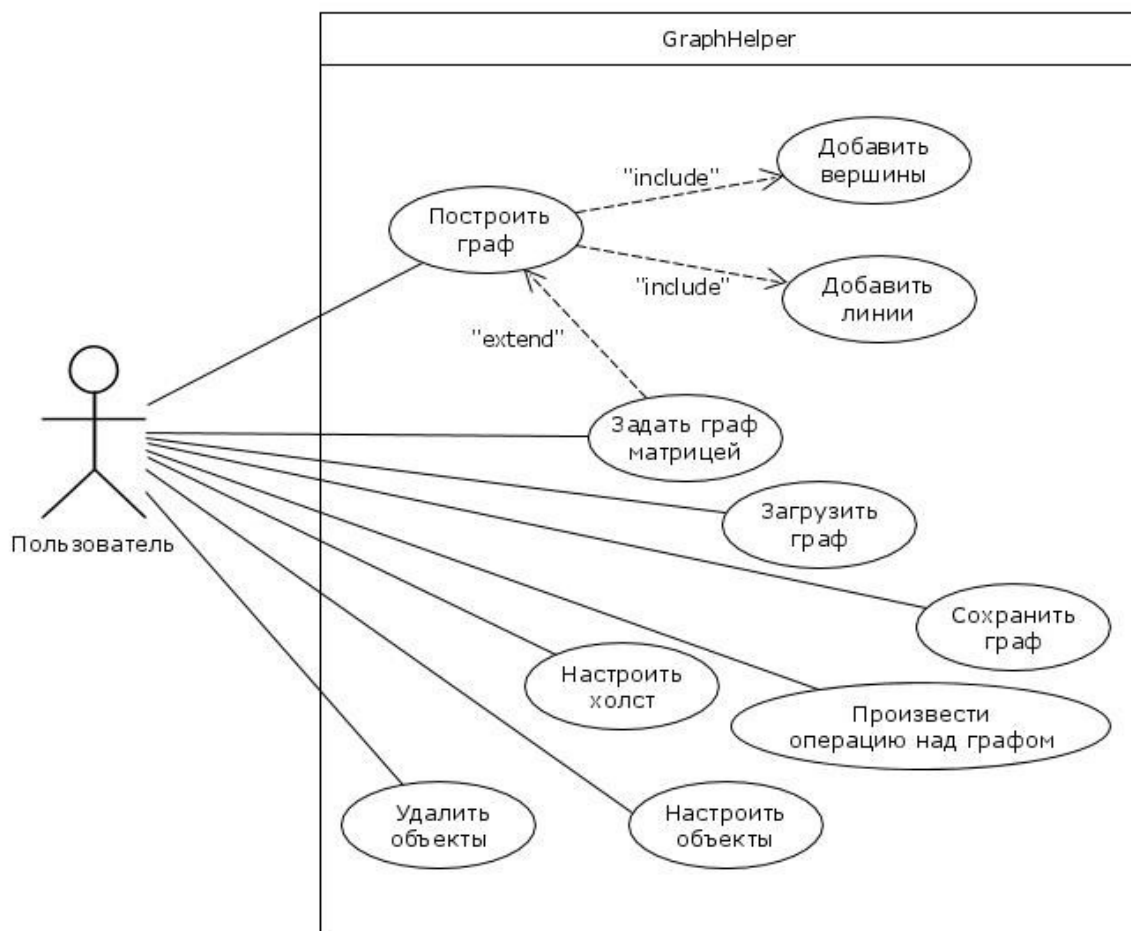


Рисунок 3 – Диаграмма прецедентов

Ниже предоставлено описание диаграммы:

3.2.1 Прецедент: Работа с графом.

3.2.2 Действующие лица: Пользователь.

3.2.3 Предусловие: Программой предусмотрено нелогичное поведение пользователя, к примеру, нельзя добавить линию, если граф не содержит рёбер, нельзя сохранить пустой холст и др.

3.2.4 Главная последовательность:

3.2.4.1 Настроить холст: С помощью инструментов, располагающихся на панели управления холстом, существует возможность изменить настройки холста, например отключить отображение имён объектов на холсте.

3.2.4.2 Построить граф: С помощью соответствующих инструментов сперва создать вершины, затем соединить их необходимым типом линии (дуга, ребро, либо петля – применение инструмента ребро по одной и той же вершине). Помимо этого существует возможность задать граф одной из характеризующих матриц, в таком случае программа сама распределит объекты на области холста.

3.2.4.3 Произвести операцию над графом: С помощью соответствующего инструмента произвести операцию над графом, в случае необходимости появляется диалоговое окно настройки операции. Все операции производятся последовательно.

3.2.4.4 Сохранить граф: Воспользоваться функцией сохранения графа, при необходимости продолжения работ над графом в дальнейшем.

С элементами пользовательского интерфейса, описание взаимодействия с которыми было предоставлено выше можно ознакомиться в Приложении А.

3.3 Условия эксплуатации

Эксплуатация программы требует от пользователя только наличия первичных навыков пользования персональным компьютером.

3.4 Требования к составу и параметрам технических средств

Для корректной работы программы необходимо следующее техническое обеспечение со следующими минимальными характеристиками:

- оперативная память – 256 Mb RAM;
- процессор – Intel Pentium III 1 Ghz;
- жесткий диск – 16 Gb HDD;
- GPU либо OpenGL ES 2.0.

3.5 Требования к информационной и программной совместимости

Программа разрабатывается на языке программирования python3 средствами графических библиотек tkinter и ttk.

4 Требования к программной документации

В предтворительный состав технической документации входят техническое задание и пояснительная записка.

5 Стадии и этапы разработки

Этапы разработки представлены в таблице 1, на следующей странице:

Таблица 1 – Этапы разработки

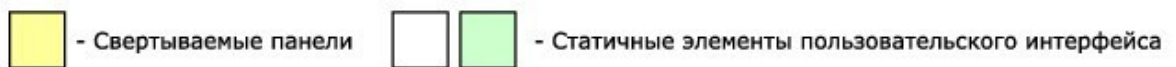
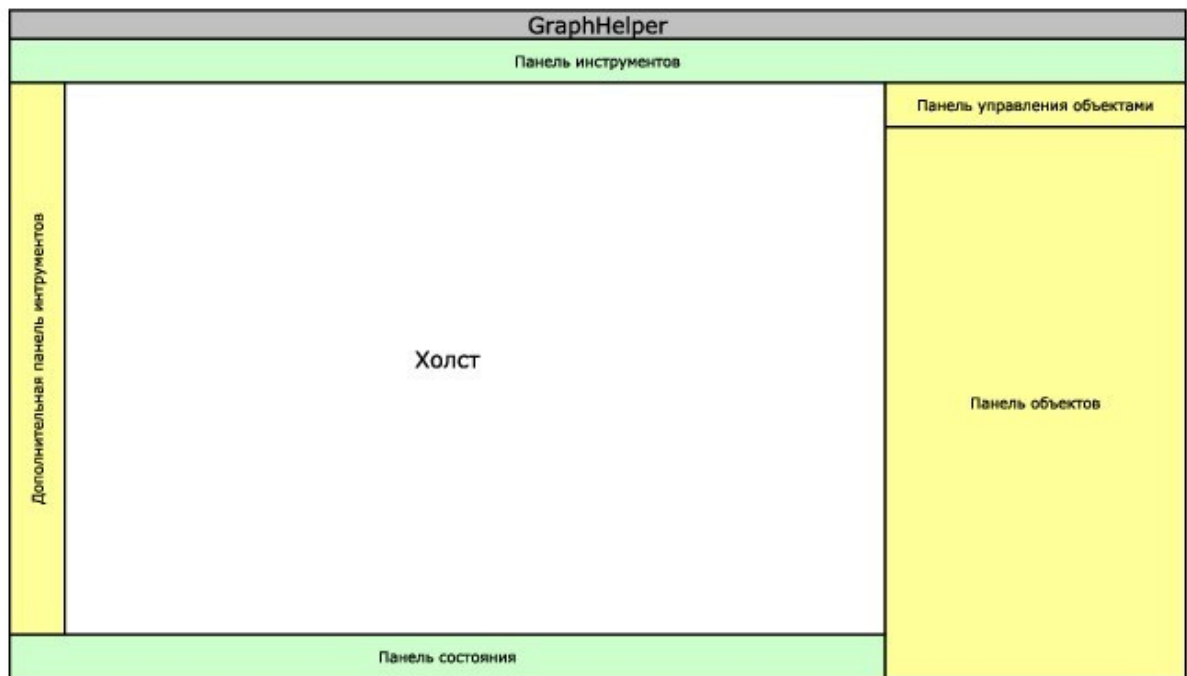
Этапы разработки	Сроки
Реализация главного меню и информационных окон	10.10 – 15.10
Реализация основных панелей	15.10 – 20.10
Реализация механики построения графов	20.10 – 1.11
Реализация алгоритмов над графом	1.11 – 20.11
Реализация механизма экспорта и импорта графа	20.11 – 5.12

6 Порядок контроля и приемки

Производится проверка корректного выполнения заложенных в программу функций, осуществляется функциональное тестирование программы и визуальная проверка интерфейса на соответствие требованиям технического задания. Прием проекта будет утвержден при корректной работе программы при различных входных данных и при предоставлении полной документации к продукту, выполненной в соответствии с требованиями технического задания.

ПРИЛОЖЕНИЕ А

Макеты пользовательского интерфейса



GraphHelper / Расщепление вершины	
Список вершин 1	список вершин 2
Cancel / Ok	



- Статичные элементы пользовательского интерфейса