

## ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ОБУЧЕНИЯ РЕШЕНИЮ ПРИКЛАДНЫХ ЗАДАЧ НА ГРАФАХ

*Т.С. Рудькова, А.В. Лобанов*

*Статья посвящена вопросу разработки программного обеспечения, созданное для обучения решению задач на графах. Разработанный программный продукт предназначен для применения в школах и в университетах с целью обучения основам теории графов. Данный программный продукт позволяет быстро выучить любой из перечисленных алгоритмов, обучение происходит на практике – что более эффективно. Пользователь получает практические навыки в решении реальных задач на графах, что повышает уровень его интеллекта и воображения, а также развитие математических и логических способностей.*

Теория графов это один из значимых разделов дискретной математики. Она имеет огромное практическое значение, к примеру, маршрутизация данных, замена оборудования, организация движения в динамической сети, распределение работ, памяти в компьютере и т.п. [1]. Поэтому в настоящее время актуальным является разработка программного продукта для обучения решению задач на графах.

В данной статье рассматривается вопрос разработки интуитивно понятного интерфейса и пошаговых подсказок, которые делают процесс обучения проще. Интерфейс в данном случае – одно из самых важных составляющих. Пользователь крайне заинтересован в высоком качестве интерфейса приложения и его интуитивно понятном построении. Главной задачей при разработке интерфейса является его проектирование.

При разработке программного продукта реализованы следующие функции: отображение узлов графа, отображения вершин графа, генерация графа, выбор алгоритма для решения, история последних действий, отображение подсказок.

При разработке интерфейса необходимо сделать его максимально понятным простому пользователю. Все кнопки и пункты меню должны находиться на привычных местах. Чтобы можно было эффективно использовать разработанное обеспечение, пользователь должен понимать, что оно из себя представляет, зачем его использовать, какие задачи можно выполнять, к чему приведет то или иное действие. Немаловажным является удобство интерфейса [5].

Расположение элементов не должно предоставлять затруднений при работе с приложением. Для создания удобного интерфейса необходимо придерживаться четкой визуальной иерархии, которая достигается путем расположения элементов на экране в определенном порядке, то есть одни и те же элементы должны отображаться в одном и том же порядке каждый раз. Плохо проработанная визуальная иерархия не приносит никакой пользы и только запутывает пользователей. Грамотная организация элементов интерфейса позволяет придать экрану менее загруженный вид. С помощью

продуманной организации элементов продемонстрируем связи между ними, и освоить такой интерфейс пользователям будет намного проще. Схожие элементы сгруппируем и разместим их на экране таким образом, чтобы пользователям стало понятно, как они связаны между собой. Благодаря грамотной организации контента можно значительно снизить когнитивную нагрузку пользователей [2].

В качестве среды разработки приложения была выбрана среда Microsoft Visual C# 2015. Данный выбор можно обосновать свободным распространением этой среды разработки – в учебных целях, удобством использования, функциональностью [2].

При запуске программного комплекса, на главном окне в пункте меню «Файл», есть подпункты «Открыть», «Сохранить» и «Новый», которые позволяют открыть и просмотреть ход решения одной из предыдущих задач, сохранить текущую, либо начать новую.

При выборе пункта меню «Граф» выводится окно создания графа, графический интерфейс которого представлен на рисунке 1. В окне существует возможность выбора настроек генерируемого графа:

- ориентированный или неориентированный граф;
- количество вершин и ребер;
- алгоритм, с помощью которого будет происходить решение поставленной задачи (алгоритм Прима, Краскала, Флери, Литла, Дейкстры, Форда-Фалкерсона, укладки графа на плоскости и т.д.).

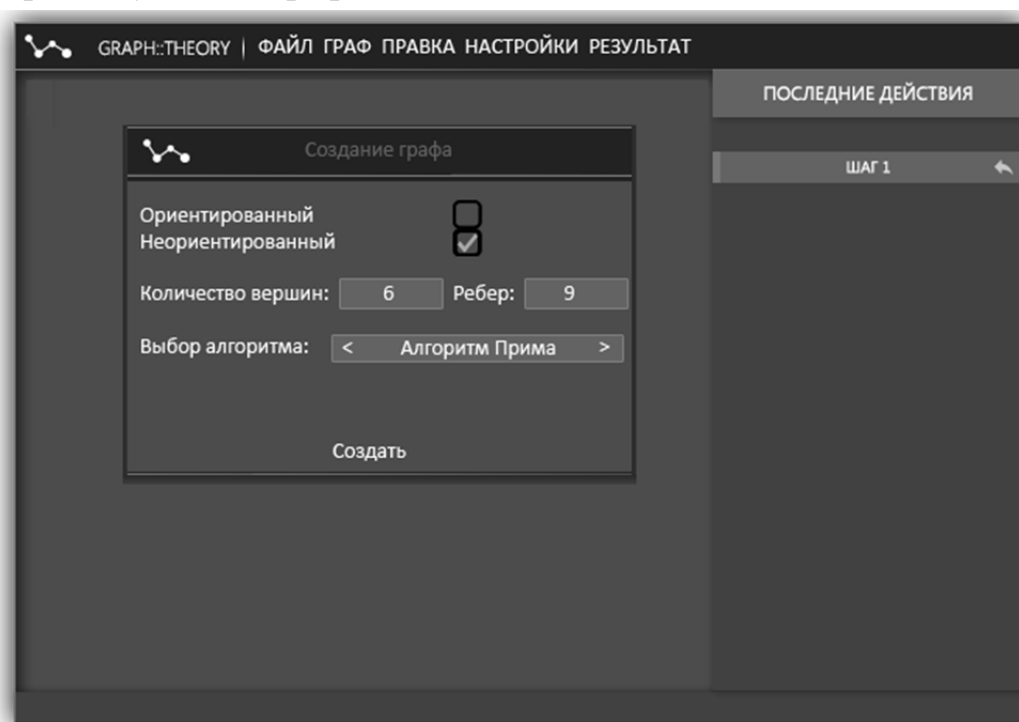


Рис.1. Графический интерфейс окна генерации графа

При успешной генерации графа выводится основное окно, представленное на рисунке 2, в котором производятся все дальнейшие действия по решению задачи.



Рис.2. Основное окно рабочей программы

А также выводятся информативные окна, которые помогают в решении задачи. Интерфейс программы во время решения задачи представлен на рисунке 3.

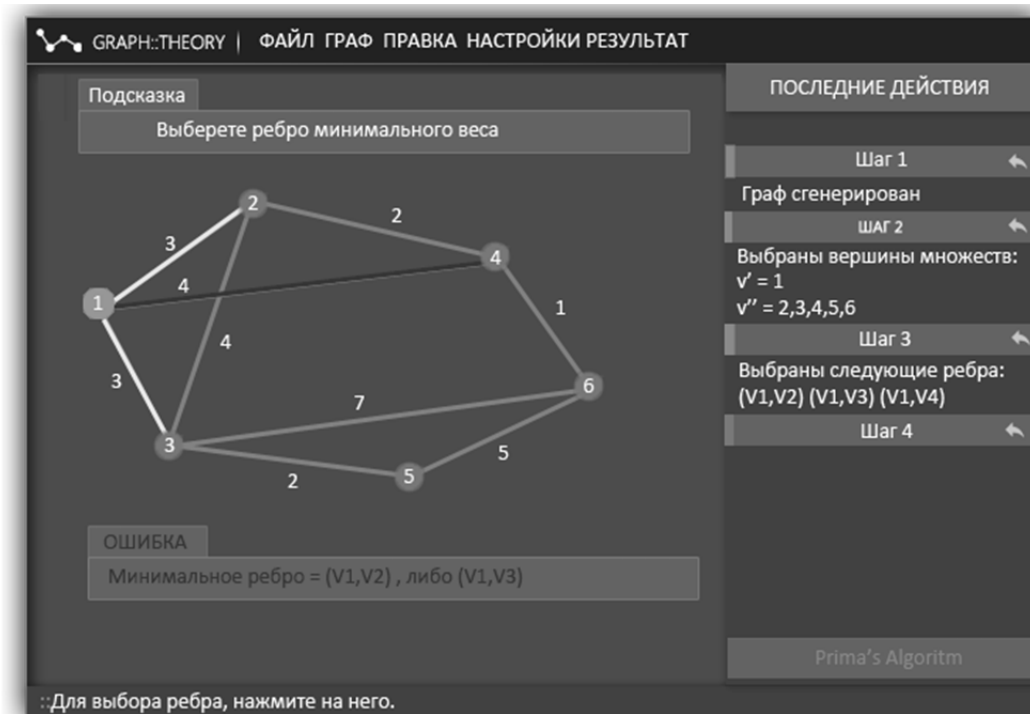


Рис.3. Интерфейс программы во время решения задачи

При успешном решении задачи, представленном на рисунке 4:

- выводится окно, в котором приводится ответ в той форме, которую предполагает выбранный пользователем алгоритм;
- вспомогательная информация: время, затраченное на решение, количество ошибок, советы для повышения уровня знаний пользователя.

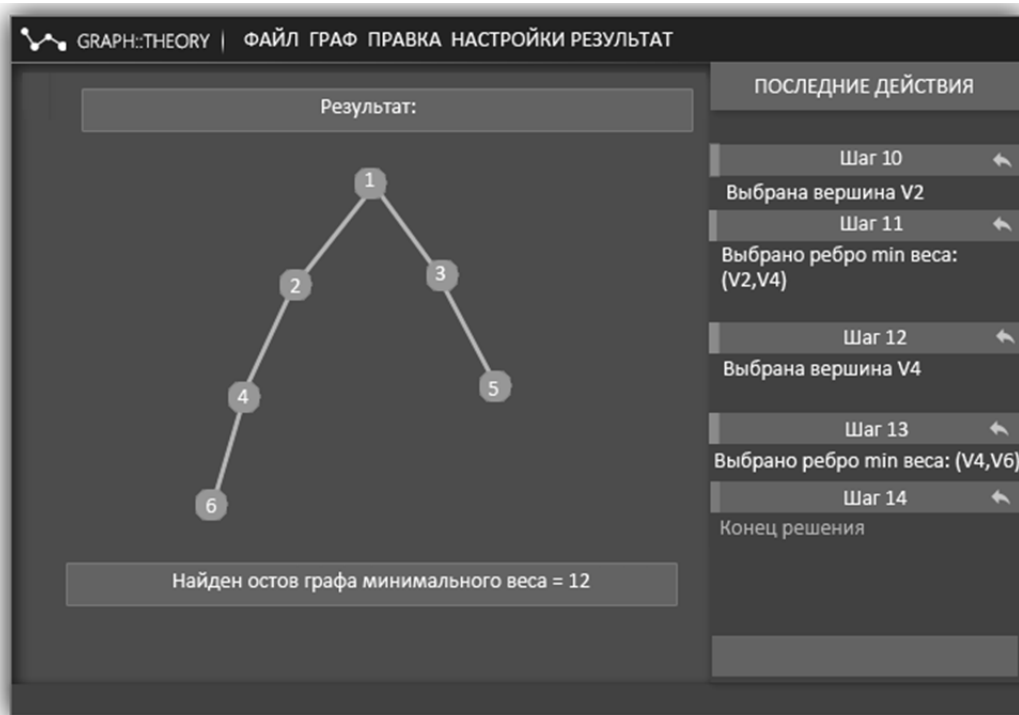


Рис.4. Интерфейс программы при успешном решении задачи

На рисунке 3 представлено окно программного комплекса во время решения задачи с помощью выбранного алгоритма «Прима». Рассмотрим более подробно это окно. Программа разбита на несколько логических блоков:

- Блок «Подсказки»;
- Блок интерактивных вычислений, с помощью которого можно проверить правильность решения;
- Блок «Ошибки»;
- Блок «Последние действия»;
- Блок с основным графом, в котором ребра и вершины подсвечиваются разным цветом, что создаёт большую информативность.

Блок «Подсказки», выводит информацию, которая помогает пользователю сделать правильный выбор, ребра или вершины. Так же показывает различные аспекты, которые присуще данной итерации алгоритма, например, выбор ребра минимального веса. Своеобразной подсказкой является подсветка различными цветами вершин и рёбер. Вершины могут подсвечиваться двумя цветами: красным или синим. Красный цвет вершины указывает на то, что вершина уже выбрана, синий цвет – данная вершина еще не выбрана. Ребра могут быть следующих цветов: зеленый, белый, серый или желтый. Зелёный цвет ребра указывает на то, что выбрано правильное ребро, белый цвет у ребра показывает то ребро, которое можно выбрать на текущем шаге. Красный цвет – на то, что ребро выбрано не верно, в этот момент появляется блок «Ошибки», где

объясняется, почему это ребро выбрано не верно. Желтый цвет у ребер появляется в тот момент, когда пользователь выбрал неправильное ребро, а те ребра, которые потенциально являются правильными, подсвечиваются желтым цветом. Для включения подсказок предусмотрен режим обучения. Для выключения – режим экзамена, когда отключены не только все подсказки по ходу решения задачи, но и блок ошибок. В данном режиме остаются доступными только блоки с вычислениями и последними действиями.

Блок интерактивных вычислений – помогает пользователю понять, как каждый из шагов решения задачи изменяет некоторые переменные в алгоритме, оценить правильность решения. В этом блоке выводятся значения ключевых переменных в выбранном пользователем алгоритме.

Блок «Ошибки» – данный блок появляется в случае, если пользователь сделал неправильный шаг в решении задачи, он выводится на экран пользователю только в режиме обучения. В этом блоке можно увидеть и понять, почему программа сделала вывод, что ход решения пользователя ошибочный. В блоке «Ошибки» выводятся значения ключевых переменных и пользователю разъясняется, почему ответ не верный. Например, на рисунке 3 показано, что минимальное ребро выбрано не верно, и программа предоставляет выбор для правильного решения «Минимальное ребро равно  $(v_1, v_2)$  либо  $(v_1, v_3)$ ». Поэтому для выбора правильного ребра пользователю осталось только нажать на него.

Блок «Последние действия» – в данном блоке пользователь может отслеживать количество итераций, которые он совершил, вернуться к любому своему шагу, чтобы исправить решение. В данном блоке выводится ключевая информация присущая данной итерации в выбранном алгоритме, например, как приводится на рисунке 3 «Шаг 3. Выбраны следующие ребра:  $(v_1, v_2)$ ,  $(v_1, v_3)$ ,  $(v_1, v_4)$ ». В самом низу этого блока, выводится выбранный алгоритм для решения поставленной задачи.

Блок «Основной граф» – в данном блоке, пользователь наглядно видит структуру графа, а также в этом блоке пользователь выбирает элементы для решения поставленной задачи: ребра и/или вершины.

На рисунке 4 представлено окно, в котором показан результат правильного решения поставленной задачи. Результат выведен в такой форме, в которой требует выбранный алгоритм «Прима» – в данном случае построено дерево решений и найден остов графа минимального веса.

В программном комплексе в пункте меню «Настройки» можно задать сложность генерируемых задач, режим обучения, указать язык, на котором потребуются работать, создать журнал – где будут сохраняться все решения задач.

С помощью этих действий, пользователи изучают структуру алгоритмов, представленных в программе, и практическим путём изучают их. Разработанный интуитивно понятный интерфейс способствует этому.

Таким образом, разработанный программный продукт с интуитивно понятным интерфейсом позволит освоить выбранный алгоритм и получить практические знания в решении выбранных задач.

## Список литературы

1. Теория графов [Электронный ресурс] – Режим доступа: [https://ru.wikipedia.org/wiki/Graph\\_theory](https://ru.wikipedia.org/wiki/Graph_theory). – Дата доступа: 19.02.2016.
2. Шилдт Г. C# 3.0: полное руководство. – М.: ООО «И.Д. Вильямс», 2010. – 992 с.
3. Richard Johnsonbaugh. Discrete Mathematics. — 7th edition. — Prentice Hall, 2008.
4. Конечная математика // Большая советская энциклопедия: в 30 т. / гл. ред. А. М. Прохоров. — 3-е изд. — М.: Советская энциклопедия, 1969—1978.
5. Разработка программного комплекса для обучения решению задач на графах: электронный сборник трудов молодых специалистов Полоцкого государственного университета. Сер. Промышленность. Информационные технологии. N10(80); Т.С. Рудькова, А.В. Лобанов, Е.И. Погребняк. – Новополоцк, – 2015. – 113-115 с.

*Рудькова Татьяна Сергеевна, ассистент кафедры технологий программирования факультета информационных технологий Полоцкого государственного университета, t.s.rudkova@pdu.by*

*Лобанов Александр Валерьевич, студент третьего курса факультета информационных технологий Полоцкого государственного университета*