

ЛИТЕРАТУРА

1. Естественное и искусственное освещение. Строительные нормы проектирования : ТКП 45-2.04-153-2009 (02250). – Введ. 2010-01-01. – Минск : Минстройархитектуры, 2010. – 110 с.
2. Здания и сооружения. Методы измерения освещенности : ГОСТ 24940-96. – Введ. 1997-01-01. – М. : Минстрой России, ГУП ЦПП, 1997. – 24 с.

УДК 004.45:519.178

**РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА
ДЛЯ ОБУЧЕНИЯ РЕШЕНИЮ ЗАДАЧ НА ГРАФАХ****Т.С. РУДЬКОВА, А.В. ЛОБАНОВ, Е.И. ПОГРЕБНЯК***(Представлено: М.В. ДЕКАНОВА)*

Рассмотрен вопрос разработки программного обеспечения, созданного для обучения решению задач на графах.

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.

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 появляется окно выбора характеристик графа, указывается алгоритм, с помощью которого производится решение, а также количество вершин и ребер, потом появляется главное окно программы, которое показано на рис. 1.

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. По нажатию на пункт «Граф», всплывает окно с параметрами генерации задачи, с возможностью выбора количества ребер. Пункт «Правка» включает в себя подпункты «Отменить» и «Повторить ввод», которые выполняют функции отмены или повтора последних действий. В настройках имеется возможность изменить язык программы. По нажатию на пункт меню «Результат» программа предлагает сохранить решение задачи в текстовый файл.

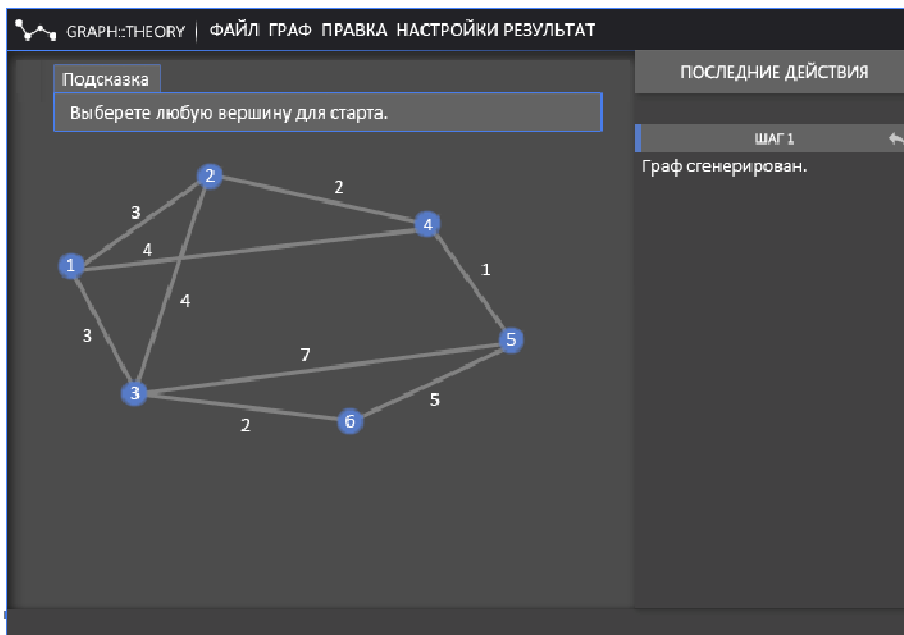


Рис. 1. Главное окно программы

Подсказки помогают пользователю сделать нужный выбор, в выборе вершины или ребра. Так же есть блок «INFO» – в котором выводятся вычисления по выбранному алгоритму. В блоке «Ошибка» – показываются ошибки в выборе вершины либо ребра, в случае если выбранный элемент не подходит под оптимальное решение выбранного алгоритма.

Пример работы программы показан на рис. 2.

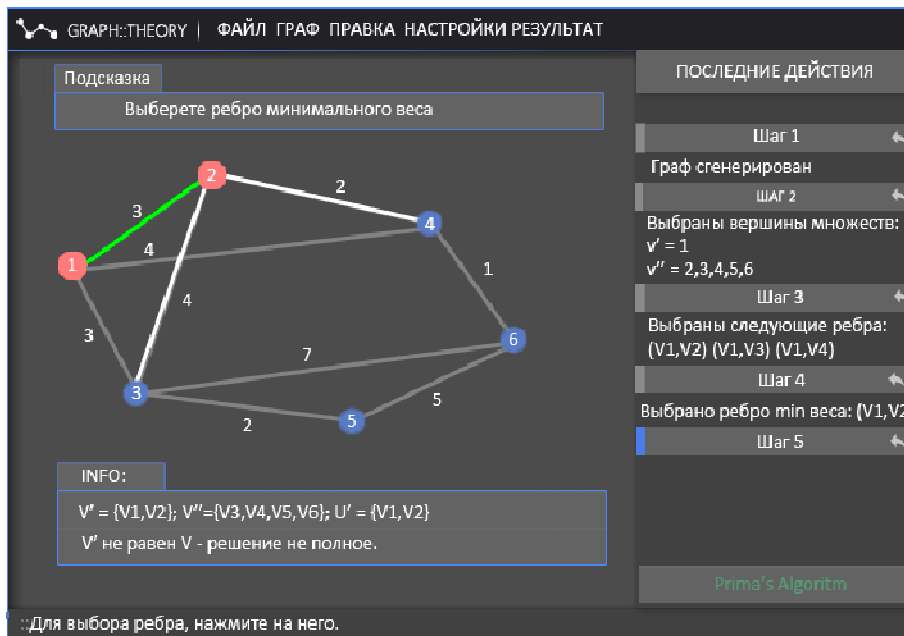


Рис. 2. Окно программы во время выполнения

Во время работы программы пользователь выбирает вершину либо ребро графа, которая была сгенерирована во время запуска программы. Для правильного выбора вершины либо ребра программа предоставляет помощь – подсказки и рекомендации показываются в блоке «Подсказки» и «Info». После каждой итерации, программа пересчитывает показатели, которые являются ключевыми в выбранном алгоритме, на рис. 2 указан пример решения по алгоритму «Прима» [1].

После выбора правильной вершины либо ребра, программа перерисовывает главное окно, в котором выбранный элемент подкрашивается розовым цветом, а правильное ребро – зеленым. На следующей итерации возможные элементы для выбора, подсвечиваются следующими цветами, синим для вершин, белым для ребер.

В результате перечисленных манипуляций, достигается правильное решение задачи. Тогда в данном случае выводится поздравление с окончанием решения и выводится отчет о проделанной работе (дерево решений), а также такая характеристика как «Остов графа минимального веса». Полученные значения и являются решением выбранного алгоритма.

В случае, если выбраны все вершины либо ребра, а правильное решение так и не получено, программа выдает результат о неправильных действиях и разъяснение ошибок. Существует возможность вернуться на любой из проделанных итераций, с помощью блока «Последние действия», в нем есть элемент «стрелка», которая переносит пользователя на выбранную итерацию.

С помощью этих действий, пользователи изучают структуру алгоритмов, представленных в программе и практическим путем 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.

ЛИТЕРАТУРА

1. Graph_theory [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Graph_theory.
2. Шилдт, Г. С# 3.0: полное руководство / Г. Шилдт. – М. : И.Д. Вильямс, 2010. – 992 с.
3. Johnsonbaugh, Richard. Discrete Mathematics / Richard Johnsonbaugh. – 7th edition. – Prentice Hall, 2008.
4. Конечная математика // Большая совет. энцикл. : в 30 т. / гл. ред. А.М. Прохоров. – 3-е изд. – М. : Совет. энцикл., 1969 – 1978.

УДК 004.9

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СУЩЕСТВУЮЩИХ КОМПЛЕКСОВ ДЛЯ РЕШЕНИЯ ПРИКЛАДНЫХ ЗАДАЧ НА ГРАФАХ

Т.С. РУДЬКОВА, А.В. ЛОБАНОВ, Е.И. ПОГРЕБНЯК
(Представлено: *М.В. ДЕКАНОВА*)

Рассматриваются основные программные продукты, для решения прикладных задач на графах и обоснование необходимости создания новой, которая оказывала помощь в усвоении материала по теории графов. В статье также содержится краткая информация о графах и некоторых алгоритмах в теории графов.

В настоящее время исследования в областях, традиционно относящихся к математике, занимают все более заметное место. Проблема выбора оптимального варианта решения различного рода прикладных задач, относится к числу наиболее актуальных технико-экономических проблем [1].

Развитие теории графов в основном обязано большому числу всевозможных приложений, таких как, составление расписания движения транспортных средств, размещение пунктов скорой помощи или телефонных станций, оптимальный подбор интенсивностей выполнения работ, задача о распределении работ, размещение диспетчерских пунктов городской транспортной сети, а также проектирование сетей телевизионного вещания и других [2, 4]. Поэтому графы нашли применение практически во всех отраслях научных знаний: физике, биологии, химии, математике, истории, лингвистике, социальных науках, технике и т.п. Наибольшей популярностью теоретико-графовые модели пользуются при исследовании коммуникационных сетей, систем информатики, химических и генетических структур, электрических цепей и других систем сетевой структуры. Например, структура ссылок веб-сайта может быть представлена в виде ориентированного графа, в котором вершины представляют веб-страниц и направленные ребра представляют ссылки с одной страницы на другую [2].

Вопрос использования теории графов можно разбить на несколько тесно связанных между собой задач. Первая: изучение элементов теории, ее основных понятий с целью получить математические знания. Вторая: использование графовых задач для развития математических и логических способностей и воображения. Третья: применение графов в служебных целях в качестве некоторого вспомогательного средства, позволяющего облегчить процесс обучения математике.