

## РАЗРАБОТКА ЛОГИЧЕСКОЙ ИГРЫ «ИГРА В СЛОВА» СРЕДСТВАМИ C++BUILDER

**Введение.** В наше время стали набирать обороты видеоигры и компьютерные игры. В отличие от тех же игр на свежем воздухе, видеоигры обладают своими различными жанрами и сеттингом [1]. Все, кто имеет дело с компьютером, так или иначе сталкивались с компьютерными играми, и подавляющее большинство может сходу назвать несколько игр, которые им особенно понравились. Те, кто уже совсем наигрался, почти наигрался или еще не наигрался, но в процессе общения с компьютером уже начал совмещать игры с чем-нибудь более полезным, возможно, хотели бы придумать какие-нибудь свои, не похожие ни на какие другие игры. Много захватывает в таком творчестве. И не сам процесс игры, а разработка игровой вселенной, ее проектирование и реализация. Когда можно слить воедино сценарий, графику, музыку, искусно задуманный и умело запрограммированный алгоритм. Поэтому тема данной научной работы является довольно актуальной, так как вызывает интерес у большого количества людей и может быть использована в дальнейшем в качестве игрового приложения.

**Основная часть.** Предметом исследования является программное приложение, описывающее игровой процесс.

Целью проекта является разработка логической игры «Игра в слова» средствами C++ Builder.

Требуется спроектировать функционал данного приложения, взаимодействуя с файловыми потоками информации для хранения промежуточных результатов игры.

Благодаря уникальному сочетанию удобства разработки пользовательских интерфейсов, компонентной архитектуры, однотипности доступа к различным базам данным возможно упрощенное программирование в среде разработки C++Builder.

В основном программа состоит из 3 компонентов, это Edit, Memo и Button.

Для работы с файлами необходимо подключить заголовочный файл `<fstream>`. В `<fstream>` определены несколько классов и подключены заголовочные файлы `<ifstream>` — файловый ввод и `<ofstream>` — файловый вывод.

Для начала создаем форму, в которой первоначально будут находиться компоненты ввода и вывода. В ней заранее прописан набор алгоритмов, который будет определять его поведение [2]. Далее мы вписываем функции для работы с файлом, а именно: чтение файла и запись в файл. После чего начинаются проверки такие как: существует ли введенное слово (и если оно все же существует, но в нашем словаре такого нет, то должна выполняться функция добавления этого слова в наш словарь), а также совпадает ли последняя буква с первой буквой следующего слова. Как правило, все существующие классификации слов основываются на одном, или нескольких, выделенных далее, критериях: структура, функционирование в тексте, информативность [3]. По окончании должен выполняться вывод.

Также для решения поставленной задачи нам нужно создать меню, в которое будут задействованы основные компоненты программы. Меню задается с помощью компоненты `MainMenu` и включает в себя следующие пункты:

- «В меню»
- «Поменять тему»
- «Правила игры»
- «Выход»

Теперь остановимся на каждом из пунктов подробнее.

1. В меню. Здесь должен происходить возврат на первоначальную форму, в которой находится меню нашей игры.

2. Поменять тему. Служит для задания темы игры. В нашем случае стоит выбор между темами: Страны, Города или же Свободная. Реализация выполняется путем изменения форм, в которых прописаны пути на разные текстовые файлы.

3. Правила игры. При вызове этого пункта меню должно выводиться сообщение, которое содержит основные моменты правил игры.

4. Выход. Соответственно отвечает за выход из программы.

Первая форма приложения с меню, имеет вид, изображенный на рисунке 1.



Рисунок 1 — Главное меню

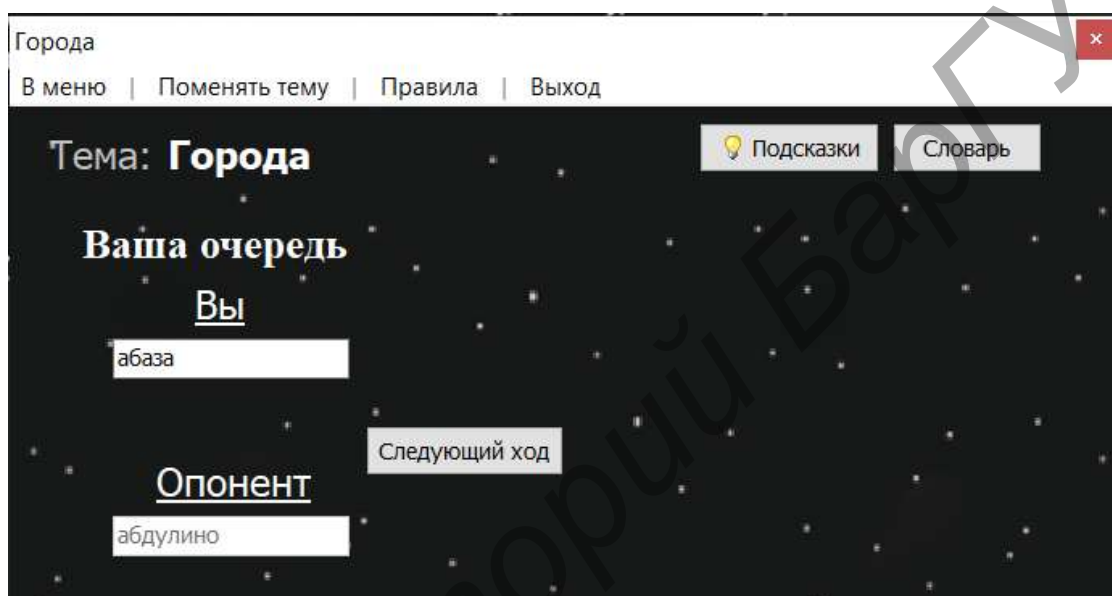


Рисунок 2 — Выведенное слово

Далее начинается игровой процесс. В результате которого, при вводе слова компьютер автоматически выдает ответ. Результат показан на рисунке 2.

Результаты тестирования программы показывают, что приложение работает корректно. Выполнение программы соответствует всем пунктам поставленной задачи. При запуске приложения все компоненты программы работают правильно. Приложение имеет красивый и удобный интерфейс, плавный и интересный геймплей [4].

**Заключение.** Разработка программы «Игра в слова» показывает, что данный процесс требует значительных творческих и умственных усилий, наличия определенного объема знаний по программированию на языке C++ и умелого использования этих знаний.

Кроме того, важными моментами при разработке программы являются описание спецификации программы, алгоритма ее реализации, хотя C++ и предоставляет необходимые средства для упрощения процесса программирования.

Проведенное тестирование программы позволяет сделать вывод о работоспособности программы и ее соответствии заданию. Выявленные недостатки предполагают дальнейшую работу над совершенствованием программы, что предоставляет начинающему программисту обширное поле деятельности для повышения знаний и совершенствовании навыков программирования в данной среде, освоении новых аспектов программирования в современной среде.

В результате учета всех отмеченных при тестировании недостатков возможно улучшение созданного программного продукта, на которое потребуется необходимое количество изменений исходного кода программы.

При наличии большего количества времени, можно добавить дополнительные функции такие как ограниченное использование подсказок (например, можно использовать подсказку лишь 3 раза), а также кнопку сдать, в случае если подсказки закончились, и игрок больше не знает подходящих слов. Несмотря на это, программа в нынешнем виде соответствует всем необходимым требованиям и условиям.

#### Список использованных источников

1. Матыборский, А. В. Разработка игрового приложения «Змейка» в среде разработки UNITY / А. В. Матыборский, А. И. Калько // Наука — практике : материалы II Междунар. науч.-практ. конф., Барановичи, 13 мая 2021 г. : в 3 ч. / М-во образования Респ. Беларусь, Баранович. гос. ун-т, редкол.: В. В. Климук (гл. ред.) [и др.]. — Барановичи : БарГУ, 2021. — Ч. 2. — С. 71—72.
2. Конопля, Е. А. Возможности искусственного интеллекта в играх / Е. А. Конопля, В. Д. Григорович, А. И. Калько // Содружество наук. Барановичи-2018 : материалы XIV Междунар. науч.-практ. конф. молодых исследователей, Барановичи, 17 мая 2018 года : в 3 ч. / М-во образования Респ. Беларусь, Баранович. гос. ун-т, Студенч. науч. о-во БарГУ ; редкол. : В. В. Климук (гл. ред.) [и др.]. — Барановичи : БарГУ, 2018. — Ч. 3. — С. 36—37.
3. Васюк, В. В. Определение игры слов как явления классификация игры слов / В. В. Васюк, О. В. Егорова // Язык и культура : вопросы современной филологии и методики обучения языкам в вузе : материалы VII Всероссийской науч.-практ. конф. с междунар. участием, Хабаровск, 21 мая 2020 года / ред. кол. : И. Ф. Уманец, отв. ред. М. В. Рябинина, Д. В. Богданова. — Хабаровск : Тихоокеанский гос. ун-т, 2020. — С. 83—87.
4. Мазур, С. А. Разработка приложения «SQUARES» в среде разработки Unity / С. А. Мазур, С. Н. Шапутько, А. И. Калько // Техника и технологии : инновации и качество : материалы V Междунар. науч.-практ. конф. (Барановичи, 20 дек. 2018 г.) / М-во образования Респ. Беларусь, Баранович. гос. ун-т ; редкол. : В. В. Климук (гл. ред.) [и др.]. — Барановичи, 2019. — С. 15—16.

УДК 004.047

И. А. Бобровник, Г. М. Раковцы

Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

### АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА «МЕТОДИЧЕСКИЙ ОТДЕЛ УНИВЕРСИТЕТА»

**Введение.** В информационном обществе доминирует производство информационного продукта, а материальный продукт становится более информационно емким. Изменяется весь уклад жизни, возрастает значимость культурного досуга, возрастает спрос на знания. В результате появились противоречия между ограниченными возможностями человека по восприятию и переработке информации и существующими массивами хранящейся и передаваемой информации. Возникло большое число избыточной информации, в которой иногда трудно сориентироваться и выбрать нужные сведения [1].

Для решения подобных проблем применяются автоматизированные базы данных. Базы данных — это совокупность структур, предназначенных для хранения больших объемов информации и программных модулей, осуществляющих управление данными, их выборку, сортировку и другие подобные действия [2].

Методический отдел является функциональным структурным подразделением университета, осуществляющим планирование, организацию, координацию, учет и контроль деятельности кафедр согласно требованиям высшей школы. Методический отдел подчиняется первому проректору, проректору по учебной работе [3].

С каждым годом количество студентов, кафедр и дисциплин в университетах стремительно растет и, во избежание различных казусов, таких как потеря данных о студентах, дисциплинах или кафедрах, появилась необходимость в централизованном контроле работы университета.

**Основная часть.** Целью данного исследования является разработка автоматизированной информационной системы «Методический отдел университета». Предметом исследования является освоение методов проектирования баз данных и работы с базами данных с помощью SQL.

В качестве среды разработки выступает Microsoft Visual Studio. Microsoft Visual Studio глубоко анализирует структуру кода, поддерживая все возможности языка C#. Редактор поддерживает авто дополнение кода и рефакторинг, предотвращает ошибки на лету [4].

В качестве системы управления базами данных будет использована MS SQL.

Функционал приложения:

1. Вывод дисциплин, по которым подготовка обеспечивается более, чем одной кафедрой.
2. Отображение списка дисциплин, которые читаются более одного семестра.
3. Постройка диаграммы: количество часов на лабораторные по каждой кафедре.
4. Сортировка кафедры по количеству преподаваемых дисциплин.
5. Вывод на печать информацию о кафедрах (Excel).
6. Вывод на печать справку о специальностях университета и отчет по кафедрам о дисциплинах, 7) обеспечиваемых кафедрой (название, семестры и специальности, в которых и для которых читается дисциплина, часы, отведенные на каждый вид занятий).
7. Вывод на печать для каждой кафедры и университета в целом должно быть подсчитано суммарное количество лекционных часов и часов на лабораторные и практические занятия, курсовые работы (Word).
8. Вывод разницы в часах, отведенных по каждой дисциплине на лабораторные и практические занятия в одном из семестров на заданном факультете.
9. Удаление данных в базе данных.
10. Добавление данных в базе данных.
11. Изменение данных в базе данных.