

## РАЗРАБОТКА ИНТЕРАКТИВНОГО ПРИЛОЖЕНИЯ ИГРЫ «ЛИНИИ» С ИСПОЛЬЗОВАНИЕМ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ

**Введение.** В современном мире, где информационные технологии играют ключевую роль в различных сферах деятельности, эффективные программные решения становятся все более востребованными. Одним из таких решений является разработка компьютерных игр, которые не только развлекают пользователей, но и служат важным инструментом для обучения и отработки навыков программирования.

Данное исследование посвящено разработке игры «Линии» с использованием объектно-ориентированного подхода. Для написания игры будет использоваться язык программирования C++. У многих программистов при необходимости разработки игры выбор падает именно на язык C++, так как он сочетает в себе принципы как обобщённого (процедурного), так и объектно-ориентированного программирования. Благодаря принципам процедурного программирования, языку свойственны такие характеристики, как эффективность, компактность, быстродействие и переносимость. А благодаря принципам объектной ориентации, язык программирования C++ предлагает новую методологию программирования, которая позволяет решать современные задачи, степень сложности которых постоянно растёт [1].

В качестве среды разработки выбрана Visual Studio с поддержкой WinForms. WinForms позволяет быстро создавать и настраивать графические интерфейсы, что позволит сконцентрироваться на логике игры.

**Основная часть.** Целью данного исследования является создание игры «Линии». Игра «Линии» представляет собой логическую головоломку, в которой игрок строит линии из одноцветных шаров на поле размером 10x10. При создании линии из пяти и более шаров одного цвета, они исчезают. Игра завершается, когда на поле не остаётся свободных позиций для новых шаров. Основным функционалом включает:

1. Поле размером 10x10, на котором размещаются шары.
2. Добавление шаров. Перед каждым ходом игрока на поле случайным образом появляются три разноцветных шара. Шары добавляются в случайные пустые позиции на поле.
3. Перемещение шаров. Игрок может перемещать шары по полю с помощью мыши. Перемещение осуществляется путем перетаскивания шара на новую позицию.
4. Удаление линий. Если удастся построить линию из четырёх или более шаров одного цвета (по горизонтали или вертикали), они исчезают.
5. Игра заканчивается, когда на поле не остаётся свободных позиций для добавления новых шаров.
6. Подсчет и отображение статистики игры, включая количество удаленных линий, время и общие очки игрока.

Когда на поле не остаётся свободных клеток для новых шаров, игра завершается.

Разработка игр, таких как «Линии», является важной задачей в контексте обучения программированию, особенно с применением объектно-ориентированного подхода. Игры служат отличной иллюстрацией применения различных парадигм и принципов разработки программного обеспечения.

В данном исследовании применен объектно-ориентированный подход, основанный на следующих принципах:

1. Инкапсуляция: данные и методы, работающие с этими данными, объединены в единые объекты (классы). Внутренние детали реализации скрыты, что обеспечивает защиту данных от прямого доступа извне и упрощает поддержку кода.
2. Абстракция: выделены основные характеристики объектов, скрывая сложные детали реализации. Это позволяет сосредоточиться на важной информации и упрощает взаимодействие с объектами.

Что бы показать, как пользователи взаимодействуют с системой, была разработана диаграмма вариантов использования. Она описывает функциональность системы с точки зрения того, что система делает. Диаграмма представлена на рисунке 1.



Рисунок 1 — Диаграмма вариантов использования приложения игры «Линии»

Диаграмма классов описывает структуру системы с точки зрения её классов, атрибутов и методов. Она показывает отношения между классами (наследование, ассоциация, агрегация, композиция). Полезна для проектирования структуры данных и архитектуры системы. Полная диаграмма классов представлена на рисунке 2.

Описание основных классов:

3. Images: класс для хранения и управления изображениями. Отвечает за хранение всех изображений, используемых в игре. Не содержит методов, только объекты Image.

4. CellChecker: класс для проверки возможности перемещения шара. Реализует логику проверки возможности перемещения шара на игровом поле. Методы: CellChecker() — конструктор класса, инициализирует объект; check() — метод проверки достижимости ячейки; resetChecks() — метод сброса массива проверок.

5. BallManager: класс для управления шарами. Отвечает за создание, перемещение шаров и проверку линий. Методы: CreateBalls() — метод создания новых шаров; MoveBalls() — метод перемещения шара; CheckBalls() — метод проверки линий.

6. Board: класс для управления игровым процессом. Создает игровое поле, управляет ходами игрока, обрабатывает перемещения и контролирует статистику. Методы: Start() — метод инициализации игрового поля; MoveBalls() — метод перемещения шаров; CheckBalls() — метод проверки линий.

7. MyForm: класс для представления главного окна приложения. Обеспечивает пользовательский интерфейс. Методы: содержит элементы управления, события и обработчики UI.

Главная форма игры представлена на рисунке 3. В процессе игры отображается статистика, содержащая количество собранных линий и шаров, и время игры. Когда на поле не останется места, время игры остановится, и игра завершится.

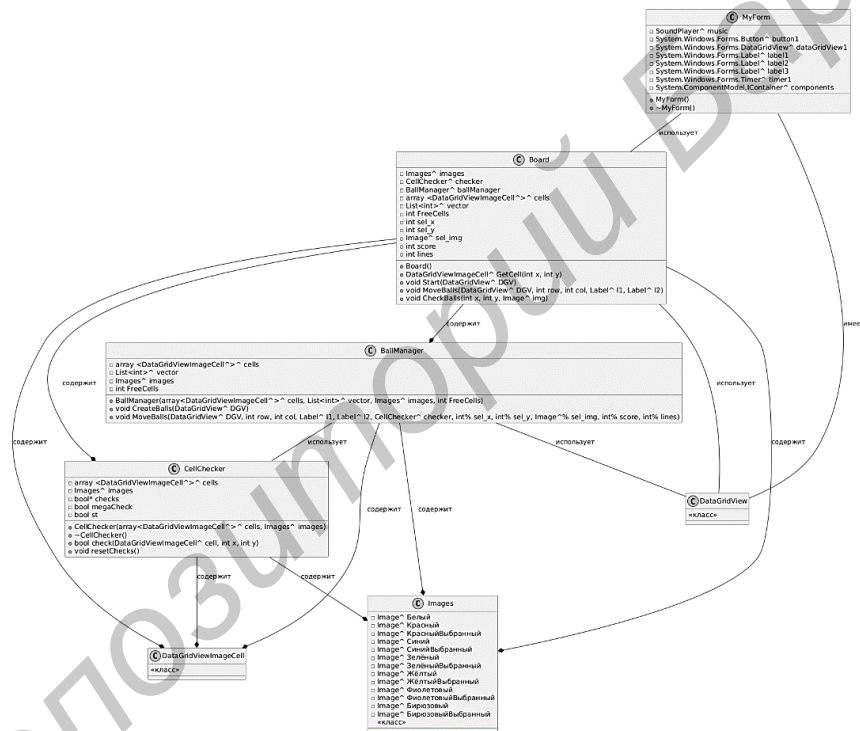


Рисунок 2 — Диаграмма классов приложения игры «Линии»

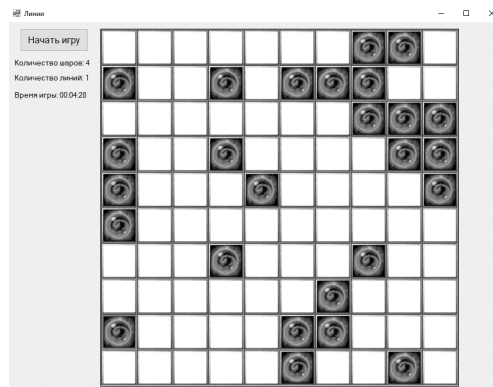


Рисунок 3 — Главная форма приложения игры «Линии»

**Заключение.** В рамках данного исследования была разработана игра «Линии» с применением объектно-ориентированного подхода. Данный проект позволил не только углубить теоретические знания в области объектно-ориентированного программирования, но и получить ценный практический опыт разработки полноценного программного продукта.

Проект продемонстрировал, как принципы объектно-ориентированного программирования могут быть эффективно применены для создания структурированного, модульного и легко модифицируемого программного обеспечения. Разработка игры «Линии» подтвердила важность использования инкапсуляции и абстракции для создания качественного программного продукта.

#### Список цитируемых источников

1. Новогран, А. А. Использование возможностей С++ для разработки компьютерной игры «Таблица умножения» // А. А. Новогран, М. А. Вареник // Материалы V Междунар. науч.-практ. конференции «Наука-практике», г. Барановичи, 17 мая 2024 г. / Баранович. гос. ун-т. — Барановичи, 2024. — С. 229—232.
2. Доусон, М. Изучаем С++ через программирование игр / М. Доусон. — СПб. : Питер, 2016. — 352 с.

УДК 004.432

М. А. Вареник, А. С. Стрельчик

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

## ИСПОЛЬЗОВАНИЕ ВОЗМОЖНОСТЕЙ С# ДЛЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ ДЛЯ УПРАВЛЕНИЯ ЛИЧНЫМИ ФИНАНСАМИ

**Введение.** Современный мир требует от нас не только умения зарабатывать деньги, но и умения эффективно ими управлять. Контроль за личными финансами может стать критическим фактором для финансового благополучия в долгосрочной перспективе, так как его отсутствие может вызвать серьёзные финансовые проблемы. С этим и связана актуальность разработки приложения для управления личными финансами с применением объектно-ориентированного подхода.

Целью данного исследования является создание такого приложения, которое позволит пользователям систематически вести учёт финансовых операций, контролировать свои расходы и доходы, а также получать полную картину своего финансового положения.

Приложение для управления личными финансами было разработано с использованием языка С#, для компиляции программы использовалась среда Visual Studio. Язык программирования С# был выбран благодаря его высокой производительности, что критически важно для обработки данных и сложных расчётов. Кроме того, современный язык С# и платформа .NET позволяют создавать кроссплатформенные приложения, способные работать на различных операционных системах, что является важным преимуществом. Он также обладает развитой экосистемой и множеством библиотек, значительно упрощающих разработку [1]. В качестве среды разработки выбрана среда Visual Studio, как мощная и удобная IDE, обеспечивающая все необходимые инструменты для эффективного создания, отладки и тестирования приложений [2].

**Основная часть.** Приложение должно поддерживать создание и редактирование транзакций, управление категориями, визуализацию данных и предоставление отчётов.

Основной функционал:

- добавление транзакций (доходов и расходов). Указание суммы, категории, даты и описания транзакций;
- установка бюджета по категориям (например, продукты, развлечения). Отслеживание расхода бюджета. Настройка оповещений о превышении бюджета;
- генерация графиков и диаграмм для анализа доходов и расходов;
- визуализация выполнения бюджета по категориям;
- поддержка различных типов отчётов (ежемесячные, годовые, недельные);
- экспорт данных в различные форматы для дальнейшего использования;
- защита приложения с помощью пароля.

Дополнительный функционал:

- подробные отчёты по категориям и периодам;
- реализация калькулятора внутри программы;
- исторический анализ изменений в финансовом состоянии;
- добавление и редактирование банковских карт.

Диаграмма классов, разработанного программного продукта представлена на рисунке 1.

Основные классы:

1. Класс Budget предоставляет удобный интерфейс для управления финансовыми потоками, позволяя пользователю отслеживать как доходы, так и расходы, что может помочь в планировании бюджета и финансовом управлении.