

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.

2. Класс Transaction является простым и удобным способом хранения информации о финансовых операциях, он позволяет легко создавать объекты транзакций с необходимыми данными и может быть использован в различных приложениях для управления личными финансами.

3. Класс Card представляет собой модель для хранения информации о банковской карте, он включает в себя поля, которые описывают основные характеристики карты, такие как номер карты, имя владельца, срок действия и код безопасности (CVS). Также класс содержит ссылку на объект бюджета, что позволяет связать карту с финансовыми данными.

4. Класс Form1 представляет собой часть приложения на C#, использующего Windows Forms для управления пользователями и их данными, хранящимися в XML-файле. Он содержит методы для создания, чтения и записи данных пользователей, а также для проверки логина и пароля.

5. Класс Calculator представляет собой простое приложение для калькулятора на языке C#, использующее Windows Forms.

6. Класс Calculator представляет собой простое приложение для калькулятора на языке C#, использующее Windows Forms.

7. Классы Year, Month, Week отображают и управляют интерфейсом формы с отчётом о финансах за год, месяц и неделю соответственно.

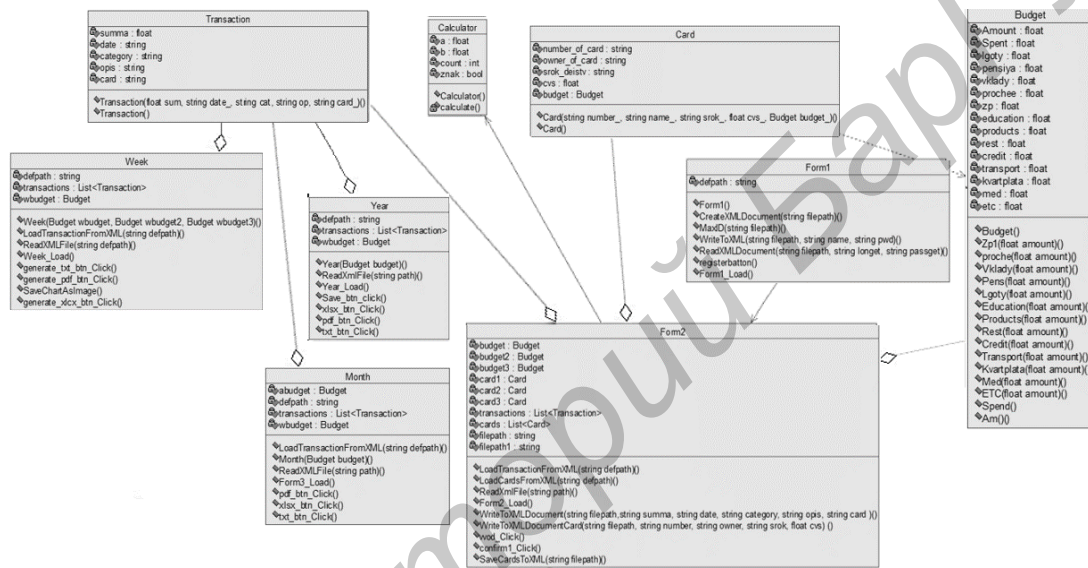


Рисунок 1 — Диаграмма классов приложения по управлению финансов

После запуска программы пользователю отобразится окно «Вход/Регистрация». Оно содержит в себе 2 кнопки: «Вход», «Регистрация», а также два текстовых поля: «Логин», «Пароль». Если пользователь впервые пользуется приложением, то ему необходимо ввести данные в текстовые поля и нажать на кнопку «Регистрация», чтобы его данные были сохранены и использованы для входа. При нажатии на кнопку «Вход» при корректности введенных данных откроется главная форма программы, она изображена на рисунке 2.



Рисунок 2 — Главная форма приложения управления финансами

Пользователь должен добавить карту, для хранения состояния своего нынешнего бюджета и осуществления транзакций. Для этого необходимо нажать на кнопку «Добавить карту», после заполнения данных, карта отобразится на форме. После добавления карты пользователь получает возможность добавлять новые транзакции. При добавлении транзакции пользователь должен помнить, что бюджет не должен быть отрицательным, иначе транзакция не выполнится. История транзакций сохраняется в файл.

При выборе кнопок «Неделя», «Месяц», «Год» пользователю отображается форма с данными о доходах и расходах за выбранный период в графическом виде. Вид формы за неделю изображён на рисунке 3. При работе с данной формой полученный отчет можно сохранить в файлы разных форматов (.txt, .pdf, .xls), нажав на одноименные кнопки.



Рисунок 3 — Окно отчетов за неделю

При нажатии на кнопку «Сохранить» на форме, показанной на рисунке 3, программа создаёт файлы: «imadedm.bmp» — график доходов за месяц, «imaderm.bmp» — график расходов за месяц, «imaded.bmp» — график доходов за неделю, «imader.bmp» — график расходов за неделю. Они разместятся в папке проекта.

Используя кнопку «Калькулятор», пользователь имеет возможность воспользоваться встроенным калькулятором, позволяющим производить расчёты параллельно с добавлением информации о транзакции.

Заключение. В процессе разработки приложения были изучены основные аспекты программирования на языке C#, а также возможности среды Visual Studio. Результаты тестирования показали, что программа работает корректно и стабильно, выполняя свою задачу. Это приложение предоставляет пользователям возможность эффективно отслеживать свои доходы и расходы, планировать бюджет и анализировать финансовые потоки.

Практическая значимость разработанного приложения по управлению личными финансами заключается в: упрощении финансового учёта, планировании бюджета, анализе финансовых потоков, повышении финансовой грамотности.

Список цитируемых источников

1. *Либерти, Дж.* Изучаем C# / Дж. Либерти, Э. Стеллман. — 5-е изд. — СПб. : Питер, 2021. — 800 с.
2. *Арнольд, Н.* Visual Studio 2017. Руководство разработчика / Н. Арнольд, Б. Наггет. — СПб. : Питер, 2018. — 624 с.

УДК 004.514

Э. А. Веракса, Г. М. Раковцы

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

РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЕТА ПОСТАВОК ЗАПЧАСТЕЙ И РЕАЛИЗАЦИИ ГОТОВОЙ ПРОДУКЦИИ НА ПРЕДПРИЯТИИ «558 АВИАЦИОННЫЙ РЕМОНТНЫЙ ЗАВОД»

Введение. Предприятия, занимающиеся производством и ремонтом техники, сталкиваются с необходимостью учета большого объема номенклатуры, координации поставок и контроля за движением готовой продукции. Ручное ведение таких процессов часто приводит к ошибкам, задержкам и снижению общей эффективности работы. Автоматизация учета позволяет значительно сократить трудозатраты, повысить точность данных и обеспечить прозрачность всех этапов производственно-логистической цепочки.

В условиях постоянного роста требований к оперативности и надежности поставок внедрение современных информационных систем становится необходимым шагом для повышения эффективности управления и оптимизации производственных процессов.