Отчёт по лабораторной работе №5-6, вариант 1

Выполнил: Флигинский Виктор Михайлович, СКБ242.

Цель

Разработка консольного приложения для управления банковскими счетами, включающего функционал создания счетов, выполнения операций (пополнение, снятие средств, начисление процентов), сохранения данных в файл и загрузки из файла, а также сравнения счетов по балансу.

Задачи

1. Разработать классы для реализации системы банковских счетов:
    1. **Базовый класс** BankAccount для общих свойств и методов счетов.
    2. **Производные классы** SavingsAccount и CheckingAccount для различных типов счетов.
2. Создать класс Customer для управления клиентами и их счетами.
3. Реализовать функционал для выполнения транзакций (пополнение, снятие, начисление процентов) через класс Transaction.
4. Добавить возможность сохранения данных о счетах в файл и их загрузки.
5. Реализовать удобный пользовательский интерфейс для взаимодействия с системой.

Классы и их описание

1. BankAccount
    1. Базовый класс для всех типов счетов.
    2. Содержит информацию о номере счета, балансе и типе счета.
    3. Включает виртуальный метод processTransaction для обработки транзакций.
    4. Поддерживает перегрузку операторов сравнения (=, >, <) для сравнения балансов счетов.
2. SavingsAccount
    1. Производный класс для сберегательных счетов.
    2. Содержит дополнительное поле interestRate для процентной ставки.
    3. Реализует метод addInterest для начисления процентов.
3. CheckingAccount
    1. Производный класс для расчетных счетов.
    2. Содержит дополнительное поле overdraftLimit для лимита на овердрафт.

    3. Реализует проверку на превышение лимита при снятии средств.
4. Customer
    1. Представляет клиента банка.
    2. Управляет массивом указателей на банковские счета.
    3. Содержит методы для добавления счетов, отображения их информации и поиска счета по номеру.
5. Transaction
    1. Содержит статические методы для выполнения операций с банковскими счетами:
        1. Внесение депозита.
        2. Снятие средств.
        3. Начисление процентов на сберегательные счета.

Функционал программы

1. **Создание счетов:**
    1. Поддержка двух типов счетов: сберегательный (SavingsAccount) и расчетный (CheckingAccount).
    2. Ввод данных (номер счета, начальный баланс, процентная ставка/лимит овердрафта).
2. **Транзакции:**
    1. Внесение средств на счет.
    2. Снятие средств с проверкой на доступный баланс/лимит.
    3. Начисление процентов для сберегательных счетов.
3. **Работа с файлами:**
    1. Сохранение информации о счетах в файл.
    2. Загрузка данных о счетах из файла.
4. **Пользовательский интерфейс:**
    1. Простой текстовый интерфейс с меню для выбора операций.
5. **Сравнение счетов:**
    1. Возможность сравнения двух счетов по балансу с использованием операторов >, <, =.

Структура программы

1. **Заголовочные файлы (*.h)**: BankAccount.h, SavingsAccount.h, CheckingAccount.h, Customer.h, Transaction.h
2. **Файлы реализации (*.cpp)**: Реализация классов и их методов.

3. **Основной файл (main.cpp)**: Реализация пользовательского интерфейса, организация работы программы.

Результаты

• Реализована система банковских счетов с основными функциями.
• Программа успешно поддерживает операции над счетами, сохранение/загрузку данных и сравнение балансов.
• Код структурирован по принципам объектно-ориентированного программирования.

Выводы

1. В ходе выполнения лабораторной работы были изучены и применены:
   • Принципы ООП: наследование, полиморфизм, инкапсуляция.
   • Работа с массивами указателей.
   • Функционал для работы с файлами.
2. Программа демонстрирует практическое применение ООП для создания гибкой и расширяемой системы.

Код

Исходный код размещён на GitHub.