Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный технологический университет»

Факультет информационных технологий

Отчет по лабораторной работе №8

«Технологии разработки ПО. Командная работа над проектом»

Дисциплина: «Введение в разработку программного обеспечения»

Специальность: «Информационные системы»

Выполнили:

Студенты 1 курса 3 группы 1 подгруппы

Шманай Виктория Александровна

Адамович Карианна Павловна

Собаль Анастасия Вадимовна

2024

Минск

**Введение:**

**Цель:** Написание консольной программы на С++, программа должна быть реализована наподобие казино, имея в себе азарт и риски, должны быть реализованы 4 режима игры. При создании проекта должны быть использованы навыки полученные во время учебы.

**Задачи:**

1. Разработать четыре режима игры.
2. Разработать авторизацию пользователя.
3. При разработке использовать известные для нас команды ЯП С/С++;
4. При необходимости, дополнительно изучить команды, которые помогут реализовать приложение.

**План разработки приложения:**

1. Создать консольный интерфейс, который будет понятен и удобен пользователю.
2. Разработать
   1. Правила игры
   2. Графическое представление в консоли
   3. Логику приложения
   4. Выход из приложения

**Меню:**

Меню представляет собой стандартную консоль Windows. При запуске представлен список:

1)Баланс

2)Таблица лидеров

3)Режимы

4)О проекте

5)Выход

Данные пункты меню разработаны с помощью switch.

**Casino**

**Игровой процесс**

При запуске приложения перед нами появляется консоль. Для начала нас попросят ввести свой никнейм. Потом нам предоставляется возможность над работой с главным меню которое содержит: баланс, таблицу лидеров, режимы, о проекте, выход.

На этом этапе пользователю при входе под новым никнеймом дается бонусных 500$ для начальной игры. Если пользователь выберет 1 пункт(Баланс), то ничего не произойдет, этот пункт меню используется для вывода баланса пользователя. Если пользователь выберет 2 пункт(Таблица лидеров), то ему предоставиться текстовая таблица в которой содержится никнейм и баланс пользователя, топ располагается сортировкой по балансу.

**Правила игры:**

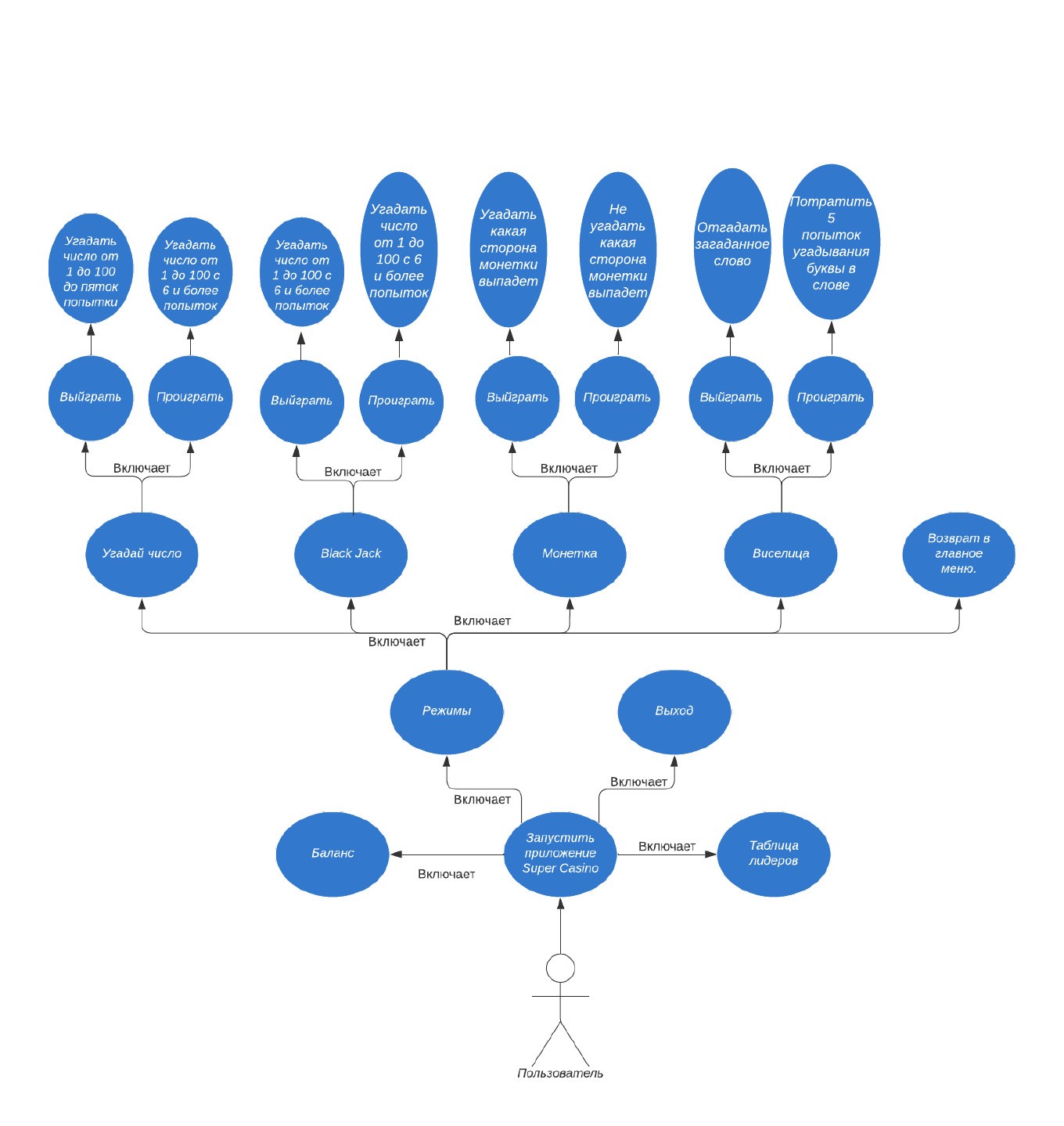
1. Пользователь играет в режимы чтобы преумножить или преуменьшить свой баланс.
2. При выборе любой игры пользователь должен ввести ставку.
3. После результата игры баланс пользователя изменяется, после выхода из игры пользователь сохраняется в таблицу лидеров как и его баланс.

**Техническая документация по приложению**

Пользовательская история для проекта:

Как **пользователь** я хочу **использовать «Casino»**, чтобы провести время с интересом.

Диаграмма вариантов использования для проекта:



**Модуль Main.cpp**

Как **пользователь,** я хочу, чтобы при входе в приложение **было меню** содержащее в себе пункты: баланс, таблица лидеров, о проекте, режимы, выход. Также вход/выход из режимов.

**Общий обзор:**

Программа представляет собой текстовый интерфейс для набора азартных игр в "Casino".

Основные функции включают в себя управление балансом, просмотр таблицы лидеров, выбор режимов игры, а также просмотр информации о проекте.

**Библиотеки и зависимости:**

Используются стандартные библиотеки C++ (iostream, fstream, string, vector, algorithm, map, Windows.h) и пользовательские библиотеки (games.h, leaderboard.h, utils.h).

**Глобальные переменные:**

currentUsername и currentBalance хранят информацию о текущем

пользователе и его балансе.

leaderboard - вектор объектов типа LeaderboardEntry, представляющих записи в таблице лидеров.

leaderboardFilename - имя файла, в котором хранится таблица лидеров.

**Функции:**

displayMenu: Отображает основное меню с информацией о пользователе и балансе.

displayModes: Отображает меню выбора режимов игры.

AboutProject: Выводит информацию о проекте и участниках.

main: Основная функция программы. Реализует цикл взаимодействия с пользователем, включая обработку выбора опций из меню.

**Циклы и переключатели:**

Используется бесконечный цикл while (true), чтобы программа оставалась активной до выбора пользователя завершить ее. Имеется вложенный цикл для выбора режимов игры с возможностью возврата в главное меню.

**Модуль games.cpp**

**Общий обзор:**

games.cpp содержит реализации функций для игровых режимов (угадай число, блэкджек, монетка, виселица).

Взаимодействие с пользователем осуществляется через стандартный ввод/вывод.

Игры включают в себя ставки и обновление баланса игрока.

**Игра "Угадай число" (playGuessNumber):**

Игрок угадывает число от 1 до 100.

Реализована ставка, умножение выигрыша в зависимости от количества попыток.

Обработка случаев, когда пользователь вводит некорректные ставки или выбирает выход из игры.

**Игра "Блэкджек" (BlackJack):**

Реализована карточная игра блэкджек с стандартными правилами.

Используется случайное распределение карт.

Реализована ставка и обновление баланса в зависимости от результата игры.

Обработка случаев, когда пользователь вводит некорректные ставки или выбирает выход из игры.

**Игра "Монетка" (CoinFlip):**

Игрок делает ставку и выбирает орла или решку.

С использованием генератора случайных чисел определяется выпавшая сторона монетки.

Обновление баланса в зависимости от результата.

Обработка случаев, когда пользователь вводит некорректные ставки или выбирает выход из игры.

**Игра "Виселица" (Hangman):**

Игрок угадывает слово, вводя буквы.

Реализовано отображение угаданных букв и попыток.

Игра завершается, если слово угадано или закончились попытки.

Обновление баланса в зависимости от результата.

Обработка случаев, когда пользователь вводит некорректные ставки или выбирает выход из игры.

**Обработка ошибок и улучшения:**

Во всех играх реализована проверка на корректность ввода ставки.

Может быть полезно добавить дополнительные проверки и обработку ошибок для повышения устойчивости программы.

Внимание к поддержке различных языков в Hangman с использованием SetConsoleCP(1251).

**Модуль leaderboard.cpp**

**Общий обзор:**

leaderboard.cpp содержит реализации функций для работы с таблицей лидеров.

Операции включают сохранение и загрузку таблицы из файла, отображение таблицы и обновление баланса игрока.

Все функции взаимодействуют с вектором std::vector<LeaderboardEntry> leaderboard.

**Функция saveLeaderboard:**

Сохраняет таблицу лидеров в текстовый файл.

Открывает файл для записи и записывает каждую запись из вектора с именем и балансом в новую строку.

**Функция loadLeaderboard:**

Загружает таблицу лидеров из текстового файла.

Открывает файл для чтения и считывает каждую строку, разделяя имя и баланс. Создает объект LeaderboardEntry для каждой записи и добавляет его в вектор.

**Функция displayLeaderboard:**

Отображает таблицу лидеров в консоли.

Использует форматированный вывод для красивого представления данных.

**Функция getInitialBalance:**

Возвращает начальный баланс пользователя на основе таблицы лидеров.

Поиск пользователя по имени в векторе и возвращение его баланса. Если не найден, возвращает значение по умолчанию.

**Функция updateLeaderboard:**

Обновляет таблицу лидеров с учетом измененного баланса пользователя.

Поиск пользователя в векторе и обновление его баланса. Если пользователь не найден, добавляет новую запись в вектор.

После обновления сортирует вектор по убыванию баланса.

**Обработка ошибок:**

Реализованы сообщения об ошибках при неудачных попытках открыть файл для чтения или записи.