Отчет по лабораторной работе: Игра «Быки и коровы»

1. Постановка задачи

Целью данной лабораторной работы является разработка консольного приложения на языке C++, реализующего игру «Быки и коровы». В игре:

- Компьютер загадывает случайное число, состоящее из n различных цифр (от 0 до 9), причем первая цифра не может быть нулём.
- Игрок (пользователь) вводит предположения о загаданном числе.
- После каждой попытки программа сообщает количество «быков» (цифры на правильных местах) и «коров» (цифры, присутствующие в числе, но на других позициях).
- Игра продолжается до тех пор, пока игрок не угадает всё число полностью.

2. Описание программной реализации

Программа реализована с использованием четырёх классов. Основные классы: Player (абстрактный), HumanPlayer (наследует Player), ComputerPlayer, Game.

Класс Player

Абстрактный базовый класс, содержащий чисто виртуальную функцию makeGuess(). Она реализована в классах-наследниках и возвращает строку — предположение игрока.

Класс HumanPlayer

Реализует ввод числа от пользователя. Метод makeGuess() запрашивает ввод строки с консоли и возвращает эту строку как предположение игрока.

Класс ComputerPlayer

Содержит метод generateSecretNumber(int length), который генерирует случайное число заданной длины из различных цифр в виде строки. Первая цифра выбирается от 1 до 9, чтобы избежать ведущего нуля, остальные выбираются случайно без повторений. Метод возвращает строку. Сложность алгоритма — линейная по длине числа, O(n).

Класс Game

Реализует основную игровую логику. В конструкторе инициализируется длина числа, игрок и создаётся secretNumber(загадываемое число) с помощью ComputerPlayer.

Метод isValidGuess(const string& guess) проверяет:
- что длина строки соответствует заданной длине числа,
- что все символы — цифры,
- что цифры не повторяются.

Если какое-либо из условий не выполняется программа попросит ввести число заново (Incorrect input! Enter a number with " << numberLength << " different dijits instead!), где numberLength – длина числа.
Сложность проверки — линейная, O(n).

Метод play() реализует цикл игры:
- Получение ввода.
- Проверка корректности*.
- Подсчёт количества быков и коров.
- Завершение игры при полном совпадении.

*Отдельно проверяется, что вводимая строка не начинается с нуля. В противном случае программа попросит ввести число заново, и напомнит, что число не должно начинаться с нуля("Incorrect input! Enter a number that doesn't start with 0 instead!")

Подсчёт быков и коров выполняется двумя вложенными циклами перебора for, сложность — O(n^2) на одну попытку, впрочем, такая сложность не критична для программы, ввиду того, что длина числа(n) ограничена и не может быть очень большим числом.

Функция main

Запрашивает у пользователя длину числа(n), проверяет, что она находится в пределах от 1 до 10. В случае корректного ввода создаёт объект класса Game и запускает игру.

3. Примеры тестирования и крайние случаи

Тест 1: Корректная работа

При длине числа 4 и загаданном числе 2076, ввод 1256 должен дать результат: 1 бык (6 на своем месте) и 1 корова (2 на неправильной позиции).

Тест 2: Победа

При загаданном числе 574 и вводе 574 результат: 3 быка, 0 коров. Игра завершается.

Тест 3: Повторяющиеся цифры

При вводе 1223 (при n = 4) программа выводит сообщение об ошибке из-за повторяющихся символов.

Тест 4: Неправильная длина

При n = 4 и вводе 123 программа выводит сообщение об ошибке из-за несоответствия длины.

Тест 5: Число начинается с нуля

При вводе 0123(при n=4) программа сообщает об ошибке, так как число не должно начинаться с нуля.

Тест 6: Крайние значения длины

При n = 1 программа работает корректно, генерируя одну цифру от 1 до 9. При n = 10 используется весь диапазон цифр от 0 до 9 без повторений, программа также работает корректно. Такие значения установлены, поскольку число не может состоять менее чем из одной цифры, и в то же время оно не может содержать более 10 цифр (т.е. n>10), потому что цифр всего 10, а по правилам игры все они должны быть разными.