

Нижегородский государственный университет имени Н.И.
Лобачевского

Отчёт по лабораторной работе

Тема лабораторной работы: игра «Быки и коровы»

Выполнил: Щербаков Никита Андреевич
Группа: 3824Б1ФИ2

1. Постановка задачи

Разработать систему классов и реализовать с ее помощью игру «Быки и коровы» в виде законченного консольного приложения. Описание работы:

- Играют два игрока (человек и компьютер).
- Игрок выбирает длину загадываемого числа с неповторяющимися цифрами – n (от 1 до 10).
- Компьютер «задумывает» n-значное число с неповторяющимися цифрами.
- Игрок делает попытку отгадать число – вводит n-значное число с неповторяющимися цифрами.
- Компьютер сообщает, сколько цифр угадано без совпадения с их позициями в загаданном числе (то есть количество коров) и сколько угадано вплоть до позиции в загаданном числе (то есть количество быков).
- Игрок делает попытки, пока не отгадает всю последовательность.

2. Описания программной реализации

Реализация выполнена на языке C++, разбита на несколько классов, каждый из которых отвечает за отдельную часть функционала.

2.1. Класс Number_check

Используется для проверки допустимости чисел (в частности, на отсутствие повторяющихся цифр).
Используется для поддержки класса Human.

Поля класса:

- **bool** acceptable
Значение допустимости числа. Принимает значения true или false.
- **int** arr_digits[10]
Целочисленный массив для хранения количества вхождений конкретной цифры в число.

Методы класса:

- **void** set_acceptable(**bool** result)
Задаёт значение допустимости числа.
Параметры: result – значение допустимости числа.
- **bool** get_acceptable() **const**
Возвращает значение допустимости числа (true или false).
- **void** reset()
Обнуляет массив arr_digits[10] и устанавливает значение true для acceptable (все поля класса инициализируются по умолчанию).
- **void** update(int digit)
Увеличивает число вхождений в число цифры digit, увеличивая значение arr_digits[digit]. Если число вхождений становится больше 1, то устанавливает значение допустимости false, то есть обнаружено повторение цифр в числе.
Параметры: digit – конкретная цифра (от 0 до 9).

2.2. Класс Computer

Используется для генерации компьютером случайного числа заданной длины. Используется для поддержки класса Bulls_and_cows.

Поля класса:

- **int** computer_n

Длина загадываемого компьютером числа.

- **std::string** computer_number
Загадываемое компьютером число.

Методы класса:

- **void** set_computer_n(**int** _computer_n)
Задаёт длину загадываемого компьютером числа.
Параметры: _computer_n – длина загадываемого компьютером числа.
- **int** get_computer_n() **const**
Возвращает длину загадываемого компьютером числа.
- **std::string** get_computer_number() **const**
Возвращает загадываемое компьютером число.
- **void** generate_random_number()
Генерирует случайное число заданной длины computer_n с неповторяющимися цифрами.
Алгоритм, который используется в методе, перемещает элементы в статическом целочисленном массиве _arr[10] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 } таким образом:
 - Генерируется случайную цифру в диапазоне [0;i], где i = 9, которая будет являться индексом массива _arr (переменная index).
 - Затем элементы _arr[index] и _arr[i] меняются местами.
 - Цифра _arr[index] записывается в загадываемое число.
 - Алгоритм повторяется при i на единицу меньше.
 - Алгоритм заканчивается когда количество выбранных цифр становится равным computer_n.

Сложность: O(n), где n = computer_n.

2.3.Класс Human

Используется для обработки и хранения информации о числе-попытке(число введённое пользователем).
Используется для поддержки класса Bulls_and_cows.

Поля класса:

- **int** human_n
Длина числа-попытки.
- **std::string** human_number
Число-попытка.
- **Number_check** human_check
Информация о цифрах в числе-попытке и его допустимости.

Методы класса:

- **void** reset()
Обнуляет все данные о числе-попытке (все поля класса инициализируются по умолчанию).
- **void** initialization(**std::string** _human_number)
Устанавливает текущее число-попытку и проверяет его допустимость через методы объекты human_check.
Параметры: _human_number – число-попытка.
Сложность: O(n), где n – длина числа-попытки _human_number.
- **bool** get_human_number_acceptable() **const**
Возвращает допустимость числа-попытки.

2.4.Класс Bulls_and_cows

Основной класс. Организует интерфейс и логику игры «Быки и коровы».

Поля класса:

- **Computer** computer
Объект класса Computer для хранения и генерации загадываемого компьютером числа.
- **Human** human
Объект класса Human для хранения и обработки числа-попытки.
- **int** bulls
Количество «быков».
- **int** cows
Количество «коров».

Методы класса:

- **int** set_number_of_digits_to_generate_computer_number(**std::string** _str)
Используется для обработки поступающей в программу длины загадываемого компьютером числа с его последующей передачей в «генератор» числа с данной длиной.
Параметры: _str – длина загадываемого компьютером числа.
Возвращаемое значение:
 - 0 – всё прошло успешно.
 - -1 – длина числа не подходит, то есть вне диапазона [1;10].
 - -2 – не все элементы в строке _str являются цифрами, то есть пользователь ввёл не число.
- **int** human_try_guess(**std::string** _str)
Используется для обработки поступающего в программу числа-попытки и его последующего сравнения загаданным компьютером числом для подсчёта «быков» и «коров».
Параметры: _str – число-попытка.
Возвращаемое значение:
 - 0 – всё прошло успешно.
 - -1 – длина числа не подходит, то есть вне диапазона [1;10].
 - -2 – не все элементы в строке _str являются цифрами, то есть пользователь ввёл не число.
 - -3 – длина числа-попытки отличается от длины загаданного компьютером числа.
 - -4 – цифры в числе попытке повторяются.
 - -5 – число-попытка с длиной большей единицы начинается с нуля.

Сложность: $O(n^2)$.

- **void** start_game()
Основной метод. Запускает игру «Быки и коровы». Реализует консольный вывод и ввод. После отгадывания загаданного компьютером числа пользователь может или завершить программу или отгадать ещё одно число.
Сложность: зависит от количества итераций (в среднем $O(k * n^2)$, где k — количество попыток отгадать загаданное число).

3.Описание и примеры тестовых задач

- Тест №1
 - Длина загадываемого числа: 3
 - Загаданное число: 145
 - Число-попытка: 641
 - Ожидаемый результат: 1 «бык», 1 «корова».
 - Результат выполнения программы: совпадает с ожидаемым результатом.
- Тест №2
 - Длина загадываемого числа: 7

- Загаданное число: 5683924
- Число-попытка: 1234567
- Ожидаемый результат: 0 «быков» 5 «коров».
- Результат выполнения программы: совпадает с ожидаемым результатом.
- Тест №3
 - Длина загадываемого числа: 4
 - Загаданное число: 9328
 - Число-попытка: 190
 - Ожидаемый результат: ошибка
 - Результат выполнения программы: «The length of the number does not match the length of the hidden number».
- Тест №4
 - Длина загадываемого числа: 5
 - Загаданное число: 59603
 - Число-попытка: 11243
 - Ожидаемый результат: ошибка
 - Результат выполнения программы: «The digits in the number should not be repeated».
- Тест №5
 - Длина загадываемого числа: 111
 - Загаданное число: -
 - Число-попытка: -
 - Ожидаемый результат: ошибка
 - Результат выполнения программы: «Incorrect length of the number being guessed».

4.Вывод

Программа реализует все требования лабораторной работы. Игра успешно работает в консольном режиме, обеспечивая интерактивное взаимодействие с пользователем.