**Содержание**

[**1. Исследование предметной области** 1](#_Toc89345149)

[**1.1 Описание программы** 2](#_Toc89345150)

[**1.2 Основные функции и подфункции** 2](#_Toc89345151)

[**1.3 Распределение ролей в команде** 2](#_Toc89345152)

[Проект был разработан одним человеком – студентом группы ПКС-1-19, Шеховцовым Глебом Евгеньевичем. 2](#_Toc89345153)

[**2. Использование система контроля версий** 5](#_Toc89345154)

[**3. Тестирование информационной системы** 8](#_Toc89345155)

[**3.1 Тестируемые функции** 8](#_Toc89345156)

[**3.2 Наборы тестов для тестируемых функций** 8](#_Toc89345157)

[**3.3 Пример фрагмента кода с использованием Юнит-тестов** 8](#_Toc89345158)

[**3.4 Пример сценария тестирования** 9](#_Toc89345159)

[**4. Кодирование** 10](#_Toc89345160)

[**4.1 Правила форматирования кода** 10](#_Toc89345161)

[**4.2 Оценка цикломатической сложности** 11](#_Toc89345162)

[**4.3 метрика Холстеда** 11](#_Toc89345163)

[**5. Результат работы программы** 12](#_Toc89345164)

# **1. Исследование предметной области**

***Цель работы*** – необходимо проанализировать предметную область, а именно описание программы, состав участников, определения функций и подфункций, макет программы.

## **1.1 Описание программы**

***Угадай число*** – логическая головоломка на скорость, которая предназначена для развития логического мышления и умения анализировать ситуацию у студентов Волгоградского Технического Колледжа. Графическое приложение содержит интерфейс угадывания числа с подсказками. После завершения результат студента будет сохранен в базе данных. Результаты всех студентов можно посмотреть при запуске приложения.

## **1.2 Основные функции и подфункции**

* *Ввод имени* – для начала игры приложение предоставляет студенту возможность ввести своё имя, которое сохраняется в базе данных.
* *Начало игры* – после ввода имени, у студента появляется возможность начать игру по нажатию кнопки. Активация игры запускает таймер и основной функционал.
* *Нахождение* – непосредственно сама игра, которую должен пройти студент менее чем за одну минуту. **За каждое введенное число студент получает подсказку от программы.**
* *Просмотр рекордов студентов* – программа дает возможность преподавателю посмотреть список результатов всех студентов, проходивших игру.
* *Запись рекорда* – после нахождения числа, студент имеет возможность записать свой рекорд в базу данных.

## **1.3 Распределение ролей в команде**

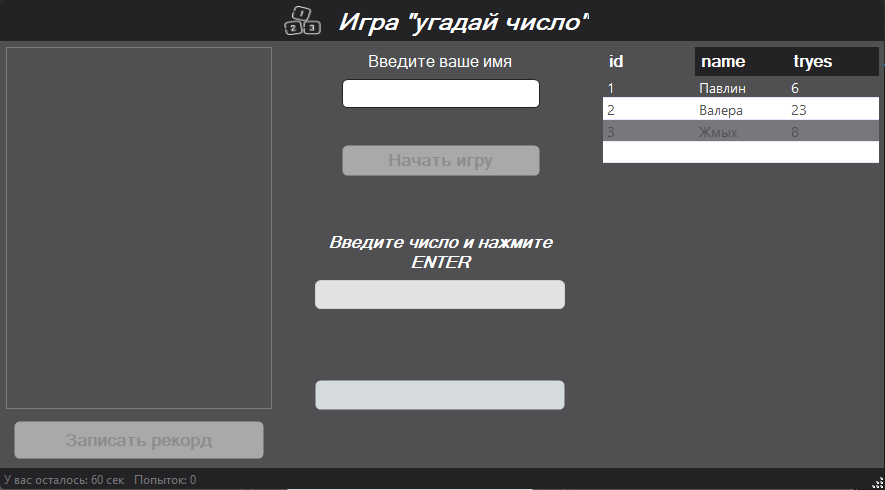
### Проект был разработан одним человеком – студентом группы ПКС-1-19, Шеховцовым Глебом Евгеньевичем.

В обязанности входили: разработка функционала программы, разработка графического интерфейса, работа с репозиторием github, создание базы данных.

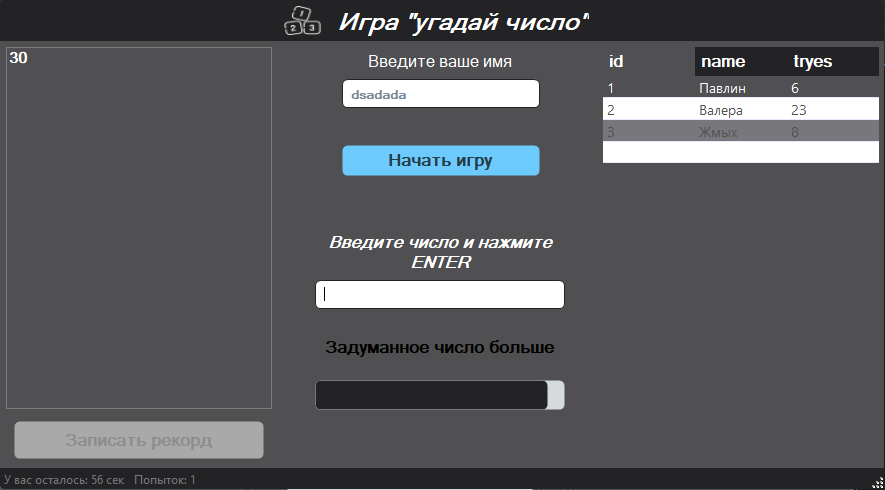
**1.4 Макет программы**

Ниже будет представлен макет программы «Угадай число».

1. Макет Главного меню



2. Макеты программы после нажатия на кнопку «Начать игру»



3. Макет тестирования

4. Макет списка результатов

Вывод: предложенный нами интерфейс программы удовлетворяет фактору качества разработки игрового приложения, как удобство пользования потому что он выполнен в минималистическом стиле и поэтому такой вид интерфейса не будет нагружать студента лишней информацией и «простой вид» интерфейса позволяет легче его понять

# **2. Использование система контроля версий**

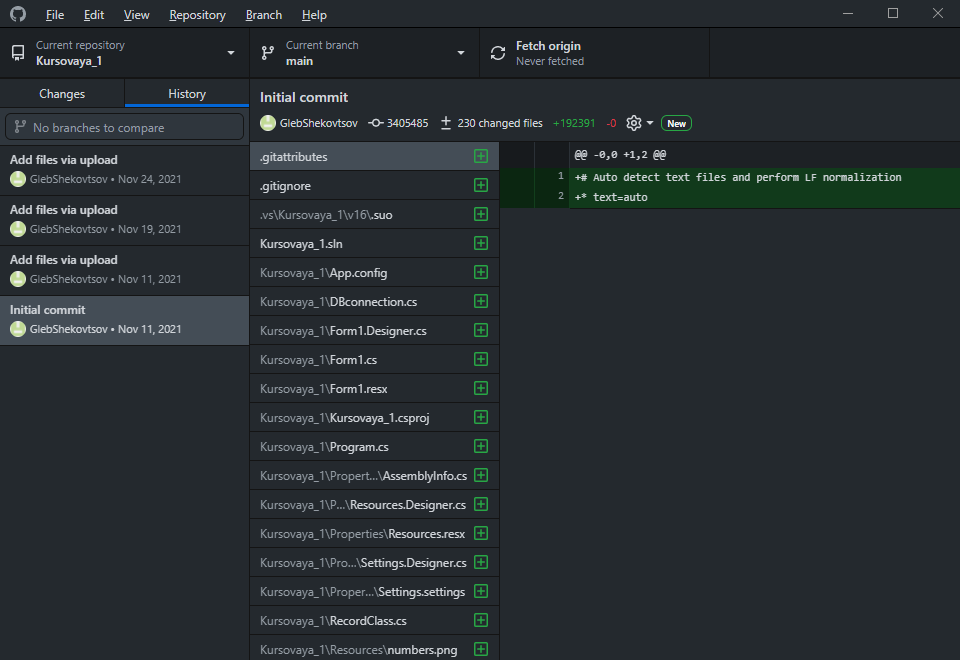
***Цель работы*** – работа и использование системы контроля версий, привести пример коммита, дерево ведения и ветки проекта.



Любая программа должна реализовываться с использованием распределенной системы контроля версий и наша не исключение. Для нашего приложения мы используем бесплатный и крупнейший веб-сервис GitHub (<https://github.com>).

Логотип GitHub

1. Пример коммита программы:



Коммит

Какие файлы были изменены

Что было изменено

Репозиторий

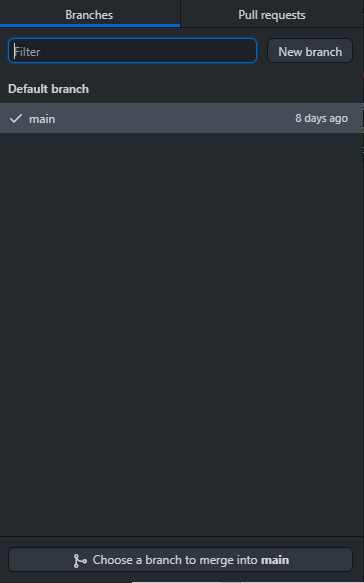
Ветка

Название Коммита

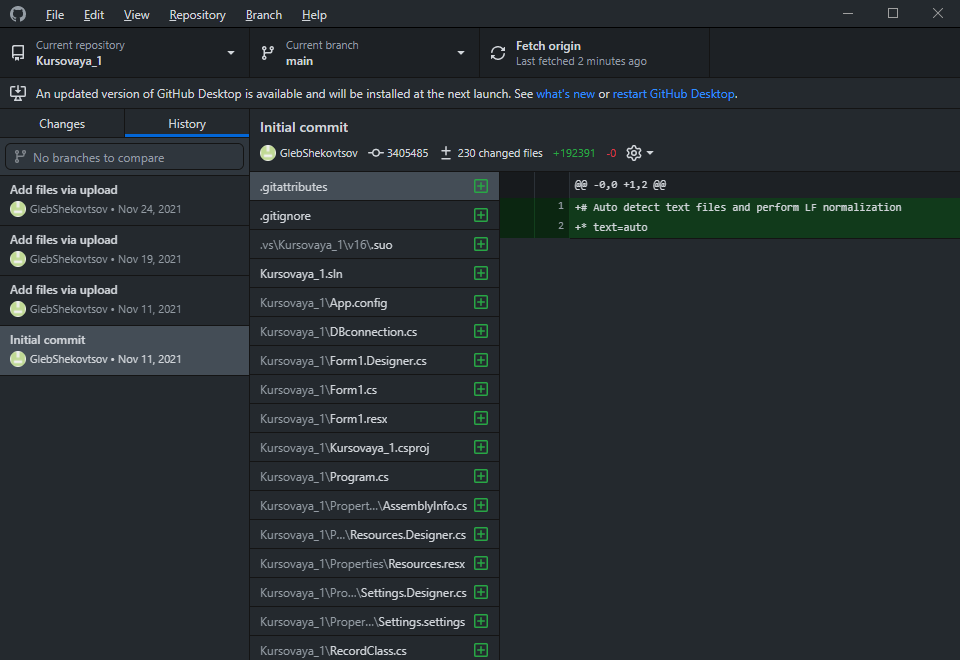
Описание Коммита

Ниже представлены ветки проекта

* Ветка main – основная ветка



2. Пример веток

2.1 Ветка ***main –*** основная ветка

# **3. Тестирование информационной системы**

***Цель работы*** – Протестировать программный код с помощью технологий автоматизированного и ручного тестирования.

## **3.1 Тестируемые функции**

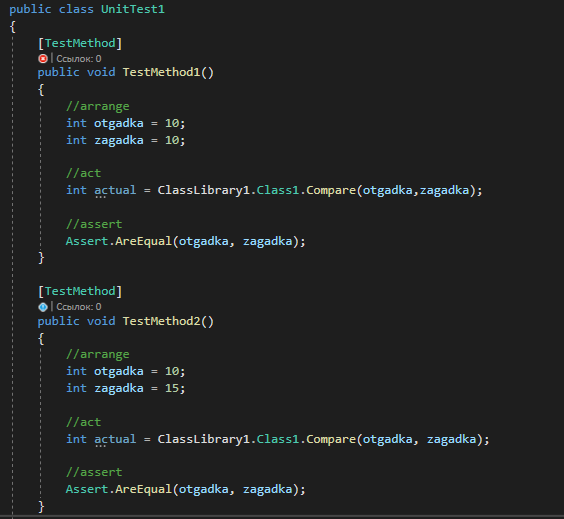
* Compare check – check

## **3.2 Наборы тестов для тестируемых функций**

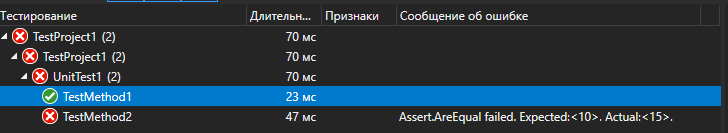
* check – Проверка равны ли числа
* check – Проверка не равны ли числа

## **3.3 Пример фрагмента кода с использованием Юнит-тестов**

Для Юнит-теста мы взяли проверку функцию check класса Compare



Результат работы Юнит-теста:



## **3.4 Пример сценария тестирования**

Для сценария тестирования был взять ввод чисел и имени в главном меню приложения.

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Действия** | **Ожидание** | **Результат** |
| 1 | Ввод букв в поле с отгадкой цифр | Не даст ввести | Совпадает с ожиданием |
| 2 | Ввод только больших букв | Даёт ввести | Совпадает с ожиданием |
| 3 | Ввод латиницы | Даёт возможность ввода | Совпадает с ожиданием |
| 4 | Ввод цифр в поле имени | Даёт ввести | Совпадает с ожиданием |
| 5 | Ввод спец. символов в поля | Не даёт ввести | Совпадает с ожиданием |
| 6 | Нажать кнопку “начать игру” без ввода имени | Не позволяет | Совпадает с ожиданием |
| 7 | Нажать кнопку “начать игру” с двумя символами в поле ввода | Не позволяет | Совпадает с ожиданием |
| 8 | Нажать на кнопку “записать рекорд” до завершения игры | Не позволяет | Совпадает с ожиданием |

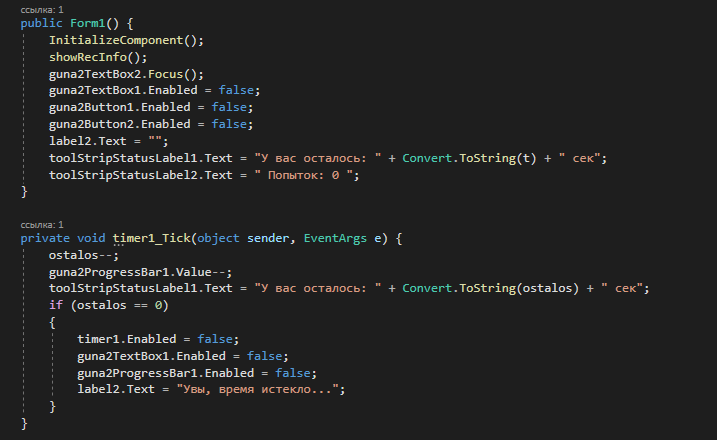
# **4. Кодирование**

***Цель работы*** – соблюдать кодирование согласно правилам форматирования кода. Выполнить оценку цикломатической сложности и использование метрики Холстеда.

## **4.1 Правила форматирования кода**

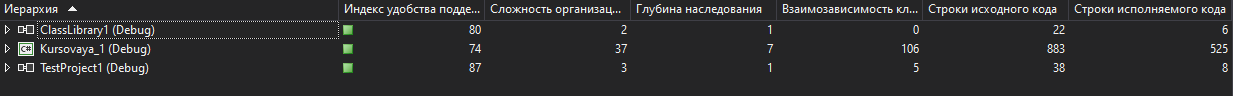
* Нотация кода – camelCase.
* Фигурные скобки на той-же строке что и функция
* Табуляция в 4 пробела
* Между разными функциями интервал 1 строка
* Стиль написания комментариев – над функциями, объединение общих элементов
* Краткое описание всех открытых функциях и входящих в них переменных через *///summary*

Прим. 1



## **4.2 Оценка цикломатической сложности**

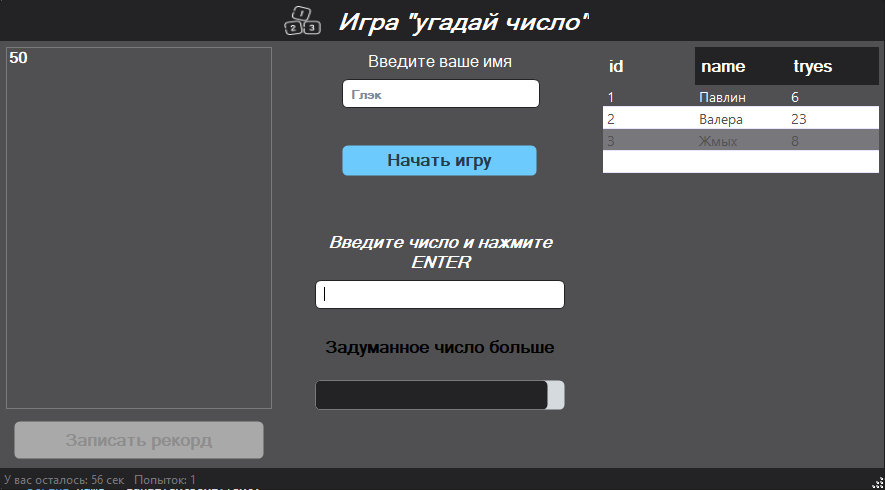
## **4.3 метрика Холстеда**



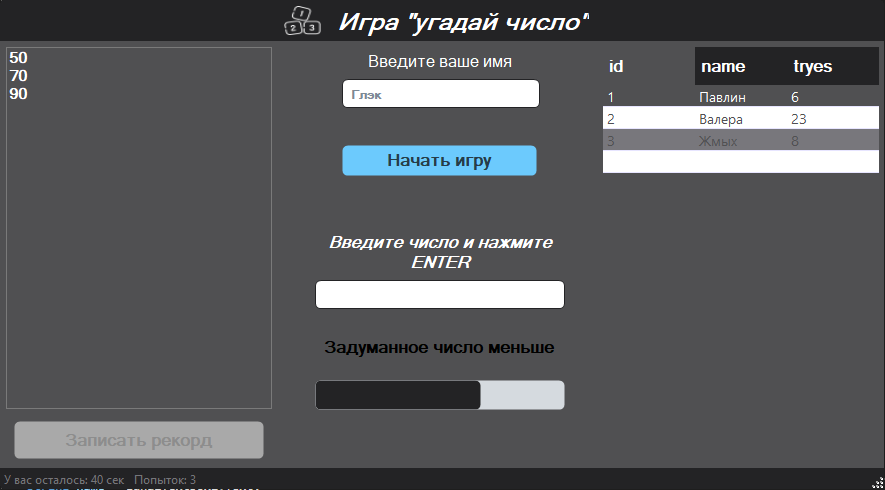
# **5. Результат работы программы**

***Цель работы*** – показать работоспособность программы в целом или отдельных функций.

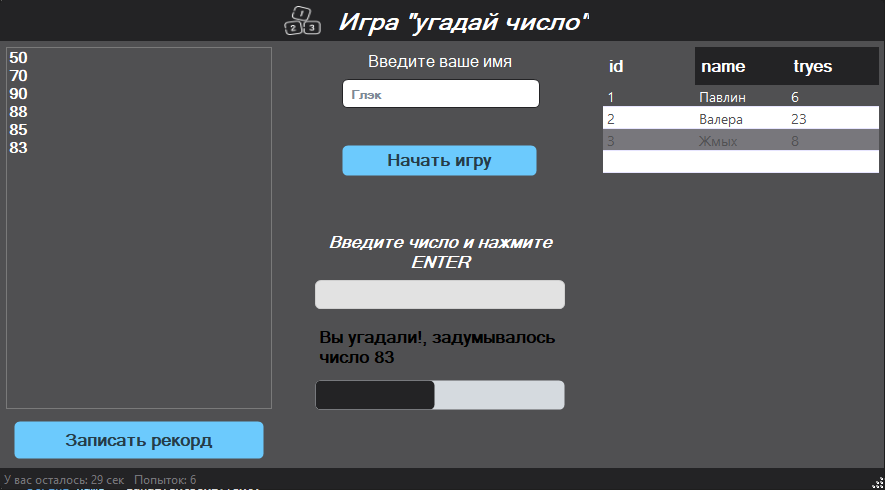
5.1. Пример работы игры:



Введенное число больше



Введенное число меньше



Число угадано

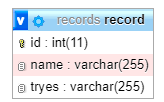
****

Таблица для рекордов