

УТВЕРЖДАЮ

Руководитель практики от Университета

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(должность, ФИО)*

МП \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(Подпись)*

Дата « \_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 202\_ год

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

ПМ.05 Проектирование и разработка информационных систем

|  |  |
| --- | --- |
| Ананьин Кирилл Сергеевич | |
| *(Ф.И.О. обучающегося)* | |
| 09.02.07 Информационные системы и программирование | |
| *(специальность)* | |
|  | |
| Учебная группа | ИСПк-202-52-00 |
|  |  |
| Место прохождения практики | ФГБОУ ВО «Вятский государственный университет», |
| Колледж ВятГУ | *(наименование организации, структурного подразделения организации)* |
|  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Итоговая оценка: |  | | | | |
| Руководитель  практики от университета |  |  |  |  |  |
|  | *(дата)* |  | *(подпись)* |  | *(Ф.И.О.)* |

Киров, 2023 г.



ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРАКТИКУ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ф.И.О. обучающегося | | | Ананьин Кирилл Сергеевич | | | | | | |
| Специальность | | | 09.02.07 Информационные системы и программирование | | | | | | |
| Учебная группа | | | ИСПк-202-52-00 | | | | | | |
| Вид практики | | | учебная практика | | | | | | |
| Сроки прохождения практики с | | | | 11.01.2021 | по | 16.06.2021 | | | |
| Место прохождения практики | | | | ФГБОУ ВО «Вятский государственный университет»,  Колледж ВятГУ | | | | | |
|  | | (наименование организации, структурного подразделения организации) | | | | | | |
| № | Виды работ, выполняемых обучающимися во время практики | | | | | | Объем работ (час) | Формируемые компетенции | | |
| 1 | Пройти инструктаж по ознакомлению с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также вводный инструктаж и инструктаж на рабочем месте | | | | | | 2 | - | | |
| 2 | Постановка задачи | | | | | | 8 | ОК2, ОК6, ОК7, ОК9 | | |
| 3 | Настройка рабочего окружения | | | | | | 8 | ОК1, ОК8, ОК9 | | |
| 4 | Анализ предметной области, постановка расширенного технического задания | | | | | | 10 | ПК5.1, ОК3, ОК4 | | |
| 5 | Проектирование | | | | | | 35 | ПК5.2, ПК5.3, | | |
| 6 | Реализация программного кода | | | | | | 35 | ПК5.4, ПК5.8, | | |
| 7 | Тестирование и отладка полученного кода | | | | | | 22 | ПК5.5, ПК6.4, ОК2, ОК9 | | |
| 8 | Подготовка отчетной документации, включая детальное описание разработанных и использованных алгоритмов | | | | | | 10 | ПК5.6, ПК5.7, ОК9, ОК10, ОК11 | | |
| 9 | Подготовка к сдаче и сдача промежуточной аттестации | | | | | | 2 | ОК4, ОК5 | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Индивидуальное задание на практику разработано в соответствии с рабочей программой практики. | | | | | |
| Руководитель  практики от университета |  |  |  |  |  |
|  | (дата) |  | (подпись) |  | (Ф.И.О.) |

|  |  |
| --- | --- |
| С индивидуальным заданием ознакомлен(а) |  |
|  | (дата, подпись обучающегося) |

**ОТЗЫВ РУКОВОДИТЕЛЯ ПРАКТИКИ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ф.И.О. обучающегося | | Ананьин Кирилл Сергеевич | | | |
| Специальность | | 09.02.07 Информационные системы и программирование | | | |
| Учебная группа | | ИСПк-202-52-00 | | | |
| Вид практики | | учебная практика | | | |
| Сроки прохождения практики с | | | 11.01.2021 | по | 16.06.2021 |
| Место прохождения практики | | | ФГБОУ ВО «Вятский государственный университет», Колледж ВятГУ | | |
|  | (наименование организации, структурного подразделения организации) | | | | |

ВИДЫ И КАЧЕСТВО ВЫПОЛНЕНИЯ РАБОТ

|  |  |  |  |
| --- | --- | --- | --- |
| Вид работ | Критерий выполнения работ | | |
| Выполнены полностью самостоятельно | Выполнены с незначительной помощью наставника | Выполнены с помощью наставника |
| Постановка задачи | V |  |  |
| Настройка рабочего окружения | V |  |  |
| Анализ предметной области, постановка расширенного технического задания | V |  |  |
| Проектирование | V |  |  |
| Реализация программного кода | V |  |  |
| Тестирование и отладка полученного кода | V |  |  |
| Подготовка отчетной документации, включая детальное описание разработанных и использованных алгоритмов | V |  |  |
| Подготовка к сдаче и сдача промежуточной аттестации | V |  |  |

Обучающийся ознакомлен с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также прошел вводный инструктаж и инструктаж на рабочем месте.

Во время прохождения учебной практики обучающимся освоены следующие профессиональные и общие компетенции:

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование компетенции | Показатели оценки | Оценка | |
| Освоена | Не освоена |
| ПК 5.1. Собирать исходные данные для разработки проектной документации на информационную систему. | Способен выполнять анализ предметной области, выявляя существенные элементы, оказывающие влияние на проектируемую систему | V |  |
| ПК 5.2. Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика. | Способен выполнять декомпозицию сущностей с целью получения наиболее полной картины о целесообразной структуре разработки | V |  |
| ПК 5.3. Разрабатывать подсистемы безопасной информационной системы в соответствии с техническим заданием. | Способен прогнозировать потенциально некорректные действия пользователя и предусматривать соответствующие реакции со стороны системы | V |  |
| ПК 5.4. Производить разработку модулей информационной системы в соответствии с техническим заданием. | Способен создавать программный код, отвечающий предъявляемым требованиям | V |  |
| ПК 5.5. Осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы. | Способен выполнять оценку корректности функционирования системы | V |  |
| ПК 5.6. Разрабатывать техническую документацию на эксплуатацию информационной системы. | Способен разрабатывать проектную, техническую и пользовательскую документации | V |  |
| ПК 5.7. Производить оценку информационной системы для выявления возможности ее модернизации. | Способен оценивать перспективы дальнейшего развития программной системы | V |  |
| ПК 5.8. Осуществлять интеграцию информационной системы с другими программными продуктами. | Способен повторно использовать готовые шаблонные решения при разработке программного продукта | V |  |
| ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. | Способен анализировать специализированную информацию и находить оптимальные пути решения задач профессиональной деятельности | V |  |
| ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности. | Способен анализировать специализированную информацию и находить оптимальные пути решения задач профессиональной деятельности | V |  |
| ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие. | Способен находить пути улучшения имеющихся решений, позволяющих повысить их общий качественный уровень | V |  |
| ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами. | Способен конструктивно обмениваться информацией с коллегами, грамотно формулировать запросы в целях получения разъясняющей информации | V |  |
| ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста. | Способен конструктивно обмениваться информацией с коллегами, грамотно формулировать запросы в целях получения разъясняющей информации | V |  |
| ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей. | Способен конструктивно обмениваться информацией с коллегами, грамотно формулировать запросы в целях получения разъясняющей информации | V |  |
| ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях. | Способен эффективно использовать компьютерное время, а также материальные ресурсы, необходимые для решения поставленных задач | V |  |
| ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности. | Способен соблюдать требования внутреннего трудового распорядка организации, охраны труда и техники безопасности в целях сохранения собственного здоровья | V |  |
| ОК 09. Использовать информационные технологии в профессиональной деятельности. | Способен применять стек современных средств разработки ПО для решения задач профессиональной деятельности в заданном контексте | V |  |
| ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке. | Способен использовать в своей работе специализированную документацию | V |  |
| ОК 11. Планировать предпринимательскую деятельность в профессиональной сфере. | Способен проектировать алгоритмические решения, принимая во внимание имеющиеся ресурсные ограничения | V |  |

**Краткая характеристика работы обучающегося**

|  |
| --- |
| Программа практики выполнена в полном объеме. Все виды работ выполнялись в срок, |
| без существенных замечаний. В достаточной степени была проявлена самостоятельность |
| и умение грамотно пользоваться источниками информации, находящимися в свободном |
| доступе. |

|  |  |  |
| --- | --- | --- |
|  |  | Руководитель практики  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Подпись ФИО  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (должность)  Дата «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 202\_\_ год |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc136806769)

[1. СВЕДЕНИЯ О РАБОТЕ, ВЫПОЛНЕННОЙ В ПЕРИОД ПРОХОЖДЕНИЯ УЧЕБНОЙ ПРАКТИКИ 4](#_Toc136806770)

[2. НАСТРОЙКА РАБОЧЕГО ОКРУЖЕНИЯ 5](#_Toc136806771)

[3. ФОРМУЛИРОВКА ИНДИВИДУАЛЬНОГО ЗАДАНИЯ 8](#_Toc136806772)

[4. ОПИСАНИЕ ВЫПОЛНЕНИЯ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ 9](#_Toc136806773)

[ЗАКЛЮЧЕНИЕ 45](#_Toc136806774)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 46](#_Toc136806775)

[ПРИЛОЖЕНИЕ А 47](#_Toc136806776)

[ПРИЛОЖЕНИЕ Б 56](#_Toc136806777)

[ПРИЛОЖЕНИЕ В 57](#_Toc136806778)

[ПРИЛОЖЕНИЕ Г 58](#_Toc136806779)

# **ВВЕДЕНИЕ**

Учебная практика ПМ.05 проходила на базе Колледжа ФГБОУ ВО «Вятский государственный университет» в период с 11.01.2021 г. по 16.06.2021 г. по субботам.

Цель практики: сформировать у обучающихся представление об алгоритмах обработки данных, а также современных библиотечных реализациях алгоритмов и структур данных и их использовании при решении прикладных задач.

Задачи практики:

– дать представление о различных структурах данных;

– показать различные методы решения алгоритмических задач;

– дать представление о сложных алгоритмах, используемых при решении алгоритмических задач.

# **1. СВЕДЕНИЯ О РАБОТЕ, ВЫПОЛНЕННОЙ В ПЕРИОД ПРОХОЖДЕНИЯ УЧЕБНОЙ ПРАКТИКИ**

В период с 11.01.2021 по 16.06.2021 при прохождении учебной практики ПМ.05 на базе Колледжа ФГОБУ ВО «Вятский государственный университет» был выполнен следующий перечень работ, представленный в таблице 1.

Таблица 1 – Сведения о работе, выполненной в период практики

| Дата | Краткое содержание выполненных работ |
| --- | --- |
| 11.01.2021 | Ознакомление с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также вводный инструктаж и инструктаж на рабочем месте |
| 12.01.2021-25.01.2021 | Постановка задачи |
| 26.01.2021-08.02.2021 | Настройка рабочего окружения |
| 09.02.2021-22.02.2021 | Анализ предметной области, постановка расширенного технического задания |
| 23.02.2021-29.03.2021 | Проектирование |
| 30.03.2021-03.05.2021 | Реализация программного кода |
| 04.05.2021-24.05.2021 | Тестирование и отладка полученного кода |
| 25.05.2021-13.06.2021 | Подготовка отчетной документации, включая детальное описание разработанных и использованных алгоритмов |
| 14.06.2021-16.06.2021 | Подготовка к сдаче и сдача промежуточной аттестации |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата) (подпись)

# **2. НАСТРОЙКА РАБОЧЕГО ОКРУЖЕНИЯ**

Python – это универсальный современный ЯП высокого уровня, к преимуществам которого относят высокую производительность программных решений и структурированный, хорошо читаемый код.  Синтаксис Питона максимально облегчен, что позволяет выучить его за сравнительно короткое время. Ядро имеет очень удобную структуру, а широкий перечень встроенных библиотек позволяет применять внушительный набор полезных функций и возможностей. ЯП может использоваться для написания прикладных приложений, а также разработки WEB-сервисов.

Python может поддерживать широкий перечень стилей разработки приложений, в том числе, очень удобен для работы с ООП и функционального программирования.

PyCharm – это кроссплатформенная интегрированная среда разработки для языка программирования Python, разработанная компанией JetBrains на основе IntelliJ IDEA. Предоставляет пользователю комплекс средств для графического отладчика и работы с кодом.

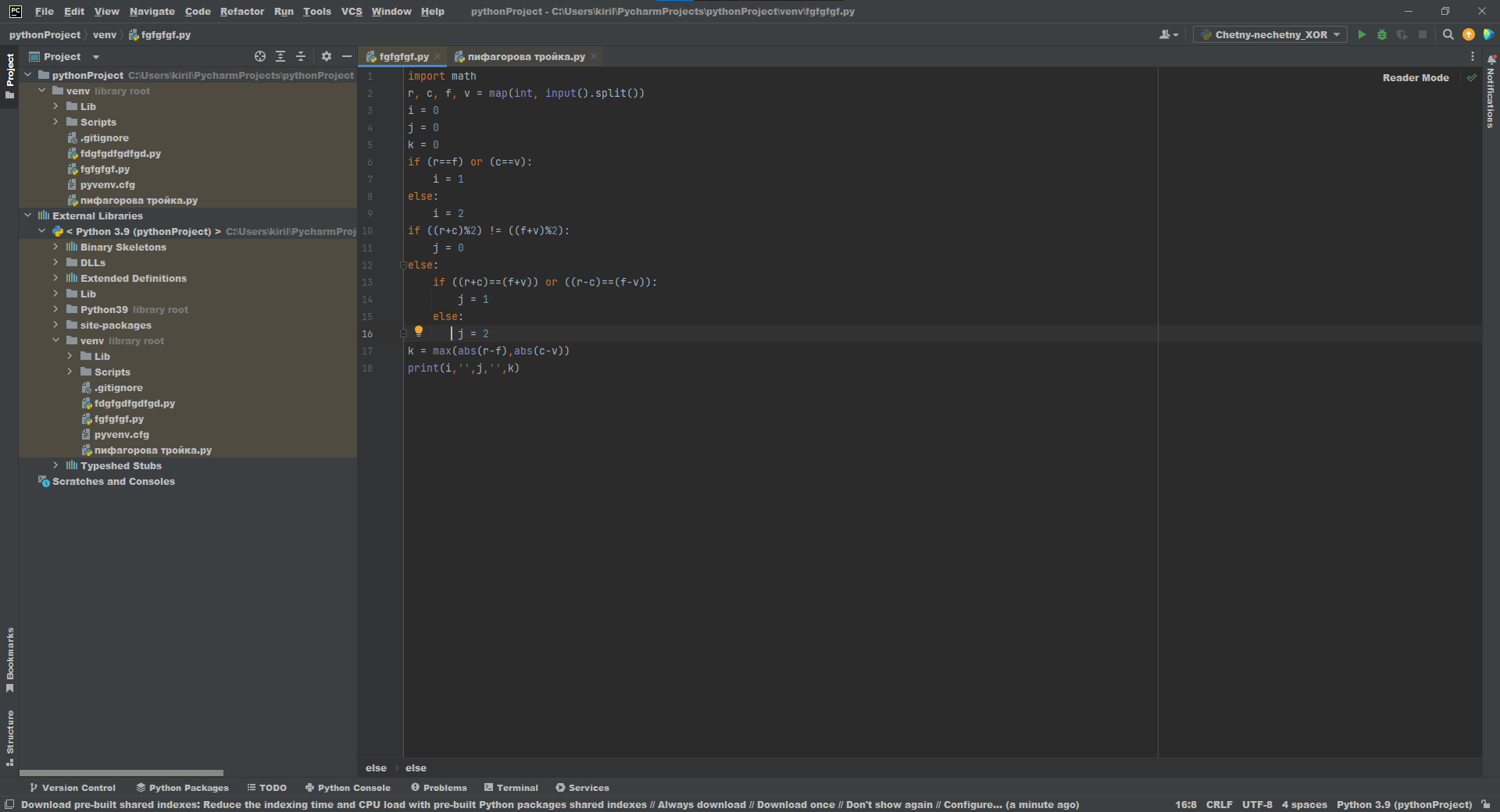


Рисунок 1 - PyCharm

Помощь в разработке программы оказали такие библиотеки и модули как:

Tkinter – это кроссплатформенная библиотека для разработки графического интерфейса на языке Python (начиная с Python 3.0 переименована в tkinter). Tkinter расшифровывается как Tk interface, и является интерфейсом к Tcl/Tk. Tkinter входит в стандартный дистрибутив Python.

Модуль random — это модуль, который позволяет генерировать псевдослучайные числа и таким образом имитировать случайные события.

Модуль Python Datetime предоставляет классы для работы с датой и временем. Эти классы предоставляют ряд функций для работы с датами, временем и временными интервалами.

Для системы контроля версий в соответствии с заданием использована система GitHub.

Ссылка, скриншот профиля (см. рис 2) и репозиторий хранящий файлы представлены ниже:

Ссылка на профиль: [Ett1s (Kirill) (github.com)](https://github.com/Ett1s)

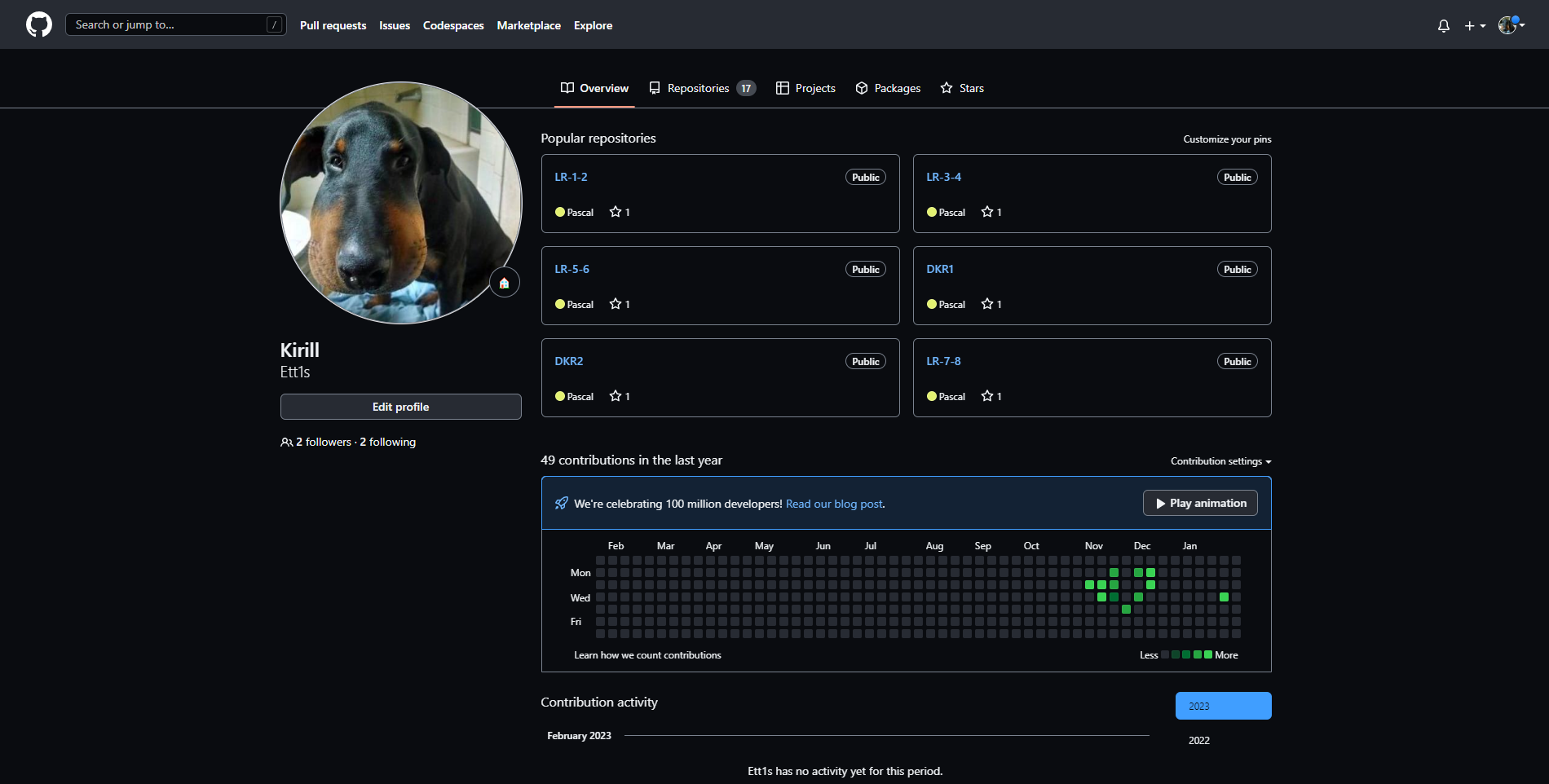
****

Рисунок 2 – GitHub

Ссылка на репозиторий: <https://github.com/Ett1s/Biki_I_Korovi_2>

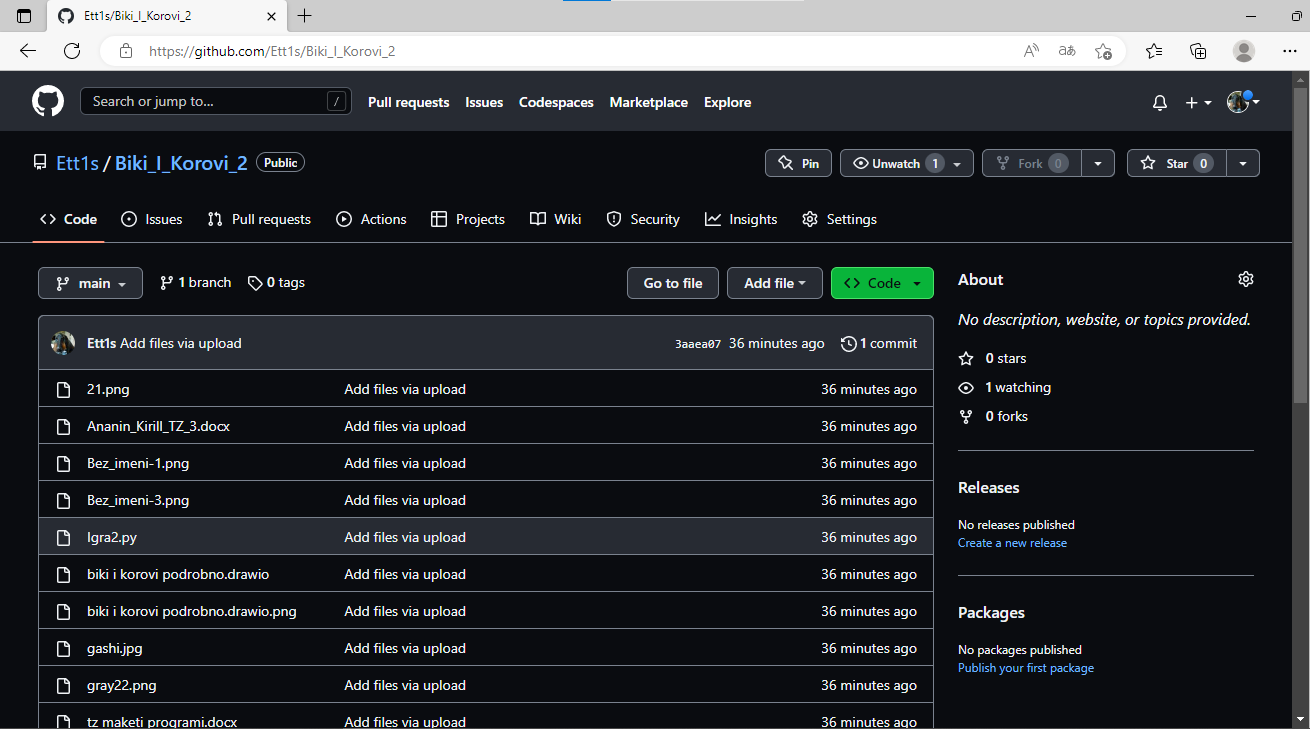
****

Рисунок 3 – Репозиторий на GitHub

# **3. ФОРМУЛИРОВКА ИНДИВИДУАЛЬНОГО ЗАДАНИЯ**

В соответствии с индивидуальным заданием на учебную практику необходимо:

* Проанализировать предметную область
* Реализовать программный код приложения, реализующего игровой процесс игры «Быки и коровы»
* Провести тестирование и отладку полученного кода
* Подготовка отчетной документации, включая детальное описание разработанных и использованных алгоритмов

Подготовить отчет к сдаче

# **4. ОПИСАНИЕ ВЫПОЛНЕНИЯ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ**

**4.1 Анализ предметной области и обзор аналогов**

Быки и коровы — логическая игра, в ходе которой за несколько попыток один из игроков должен определить, что задумал другой игрок (компьютер). Варианты игры могут зависеть от типа отгадываемой последовательности — это могут быть числа, цвета, пиктограммы или слова. После каждой попытки задумавший игрок (компьютер) выставляет «оценку», указывая количество угаданного без совпадения с их позициями (количество «коров») и полных совпадений (количество «быков»).

Игра «Быки и коровы» может применяться в повседневной жизни, например: скоротать время или же провести время с друзьями. Также она может быть использована в разных детских учреждениях для общего образования детей, так как она помогает в развитии логического мышления и памяти. В школах или в колледжах во время различных игровых мероприятий.

Игра развивает умение сравнивать и анализировать, логичность мышления, счёт, память

Настольная игра «Быки и коровы» не имеет возрастных рамок и может быть использована людьми любого возраста.

В настоящее время существуют следующие аналоги игры «Быки и коровы».

Аналог №1

[Быки и коровы играть онлайн (xn--90aeltibbl.xn--p1ai)](https://xn--90aeltibbl.xn--p1ai/Home/BullsAndCows)

Внешний вид игры представлен на рисунке 2.

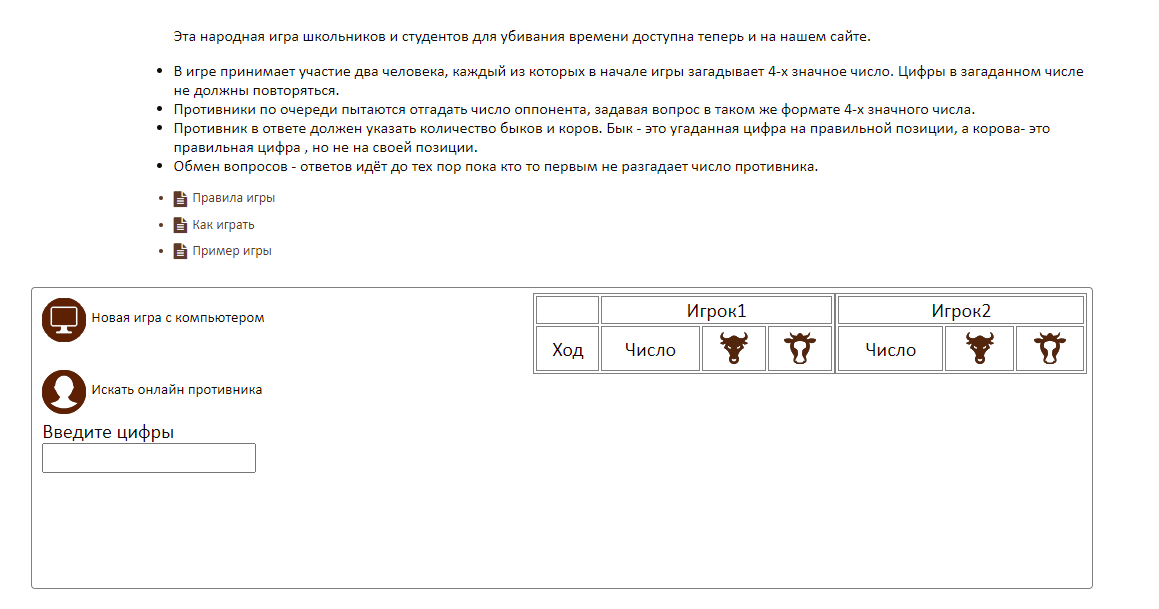


Рисунок 4 – Скриншот внешнего вида игры

К преимуществам данного аналога можно отнести возможность игры как с компьютером, так и с другими пользователями и ведение записи результатов хода в таблицу. Недостатками являются невозможность игры без доступа в интернет

Аналог №2

[Быки и коровы - Онлайн игра (vladimirus-team.top)](https://vladimirus-team.top/byki-i-korovy?ysclid=ld35hzxhdy483898306)

Внешний вид игры представлен на рисунке 3.

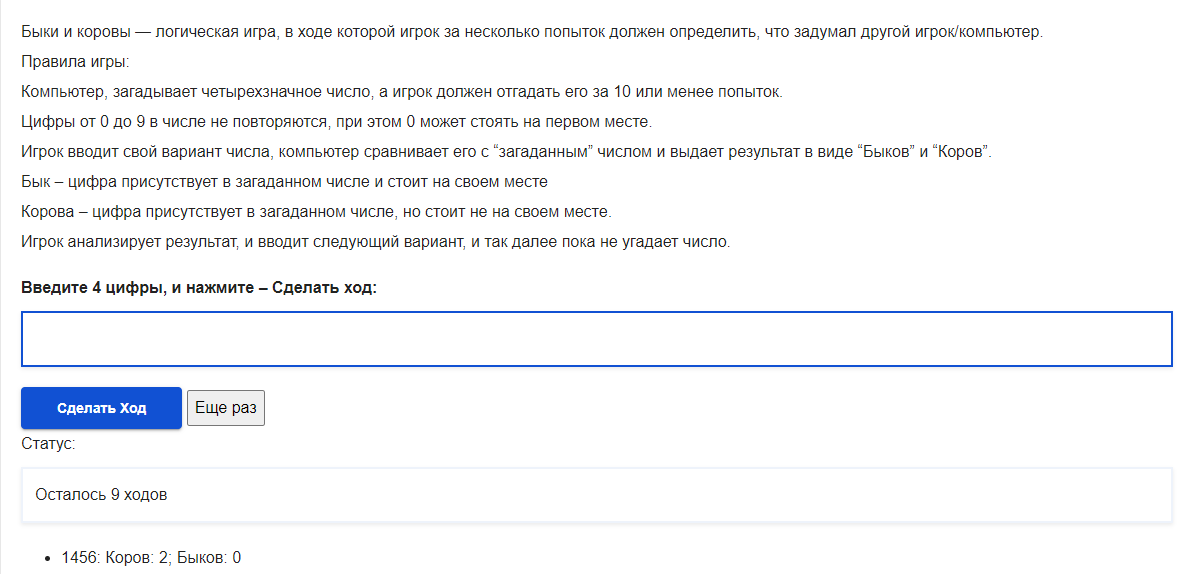


Рисунок 5 – Скриншот внешнего вида игры №2

К преимуществам данного аналога относятся наличие кнопки «Еще раз», которая позволяет начать новую игру, а также ведение записи результатов хода.

К недостаткам можно отнести невозможность игры без доступа в интернет, а также отсутствие визуализации кол-ва «Быков» и «Коров» в веденом числе

Аналог №3

[МетаШкола - Игра - Быки и коровы (metaschool.ru)](https://metaschool.ru/pub/games/bulls-and-cows/bulls.php?ysclid=lcvlqsy2gx661199465)

Внешний вид игры представлен на рисунке 4.

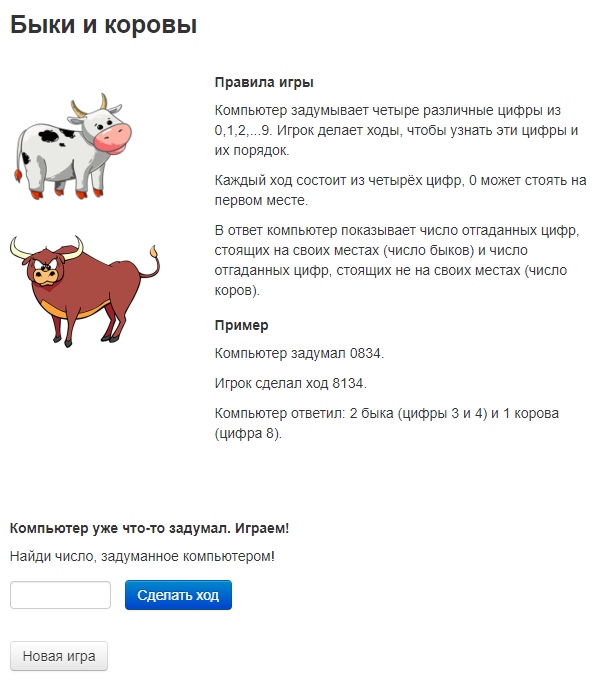


Рисунок 6 – Скриншот внешнего вида игры №3

Преимуществами данного аналога являются присутствие правил в интерфейсе игрового поля, наличием примера игры, а также красивым и привлекательным интерфейсом. К недостаткам можно отнести то, что в данную игру невозможно поиграть без доступа в интернет, а также отсутствие визуализации кол-ва «Быков» и «Коров» в веденом числе

Проведя анализ аналогов игры «Быки и коровы», был выявлен основной недостаток всех аналогов – невозможность игры без интернета. Поэтому необходимо разработать игру, которая будет запускаться через «exe» файл, будет иметь возможность игры без доступа в интернет. Так же в разработанной игре будет реализован более широкий функционал игры по сравнению с отдельными из представленных аналогов. Например: будут добавлены картинки, которых нет в некоторых аналогах, так как они очень хорошо заостряют внимание пользователя. Будет добавлена визуальная интеграция «Быков» и «Коров» (Будет выводиться столько соответствующих картинок, сколько «Быков» или «Коров» присутствуют в числе).

**4.2 Разработка алгоритмов функционирования и структуры программы**

Правила игры

Компьютер загадывает число. Цифры от 0 до 9 в числе не повторяются, при этом 0 может стоять на первом месте.

Игрок вводит свой вариант числа (содержащее только 4 цифры) , компьютер сравнивает его с "загаданным" числом и выдает результат в виде количества «Быков» и «Коров».

Количество «Быков» - количество цифр которые присутствуют в загаданном числе и стоят на своем месте

Количество «Коров» - количество цифр присутствующих в загаданном числе, но стоящих не на своем месте.

Игрок анализирует результат, и вводит следующий вариант. Соответственно после каждой такой попытки будут выводиться кол-во «Быков» и «Коров»

Игра будет окончена после того, как пользователь отгадает загаданное компьютером число или не закончит ее по собственному желанию.

Алгоритм игры

В начале компьютер задает любое рандомное число, числа которого находятся в диапазоне от 0 до 9, обязательно без повторений. После пользователь должен его отгадать. Если в введенном числе присутствует цифра, совпадающая и стоящая на том же месте что и в числе заданном компьютером, то она будет называться «Быком», а если она совпадает, но стоит не на своем месте, то она будет называться «Коровой». После каждого ввода пользователя выводиться кол-во «Быков» и «Коров».

Это является основной подсказкой для пользователя, чтобы победить в данной игре.

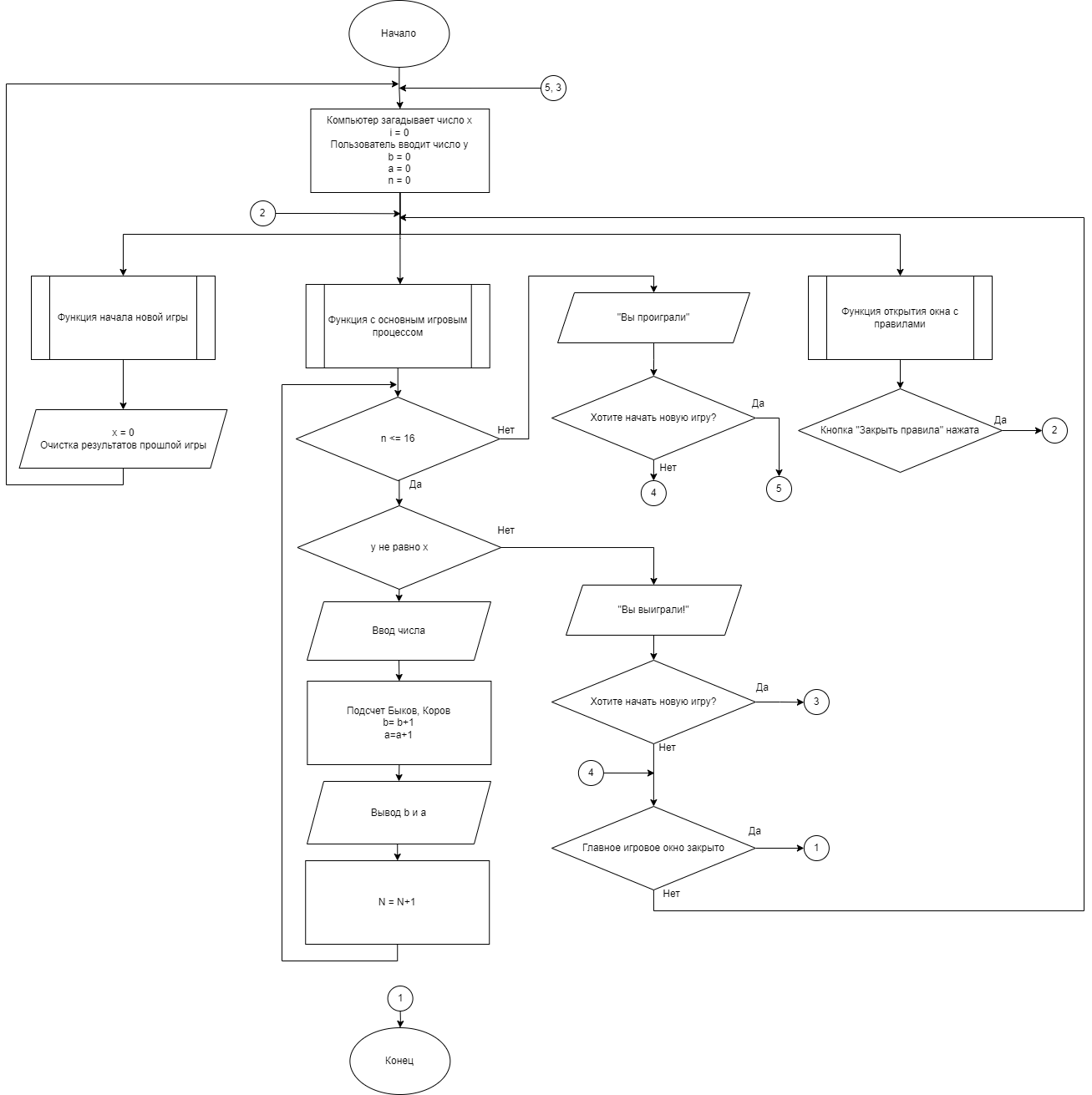


Рисунок 7 – Схема алгоритма

****

Рисунок 8 – Схема алгоритма

1. Пользователь может начать игру, прочитать правила, либо поменять загаданное число, начав новую игру с самого начала.
2. Открыв правила, он может их прочитать, а после закрыть кнопкой.
3. Нажав на кнопку новой игры, генерируется новое число, очищаются результаты прошлой игры (если таковы были).
4. Пользователь вводит число.
5. Компьютер проверяет его исходя из условий проверки и если оно их проходит он проверяет его на кол-во «Быков» и «Коров».
6. Игра продолжается на протяжении 15 ходов, если игрок отгадает число то выйдет окно с поздравлениями и вопросом, не желает ли он начать новую игру, если он выберет да (см. пункт 3), если нет, то (см. пункт 1)
7. Если игрок не сможет отгадать число, то ему выйдет сообщение, что к сожалению он не смог отгадать число и вопрос, не хочет ли он начать игру с самого начал, если он выберет да, то (см. пункт 3), если нет, то (см. пункт 1).

Прототипы экранной формы (см.рис 9 и 10)

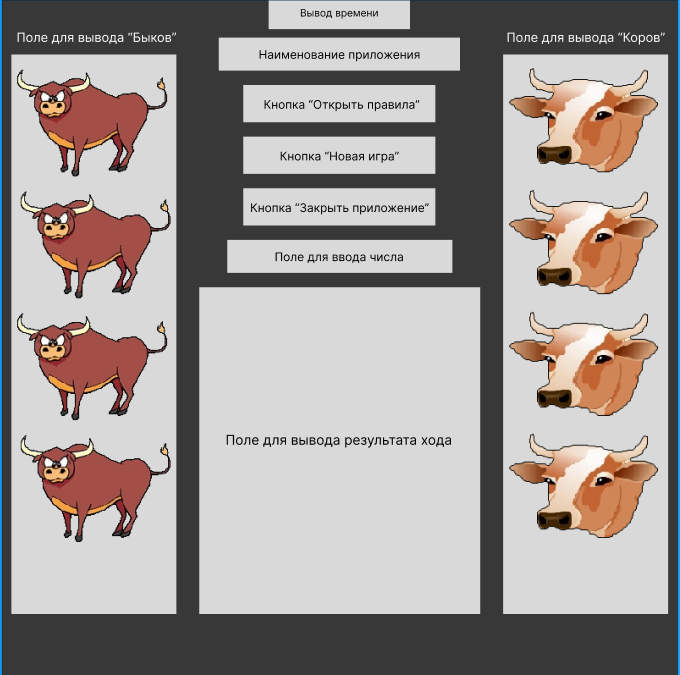


Рисунок 9 – Прототип экранной формы

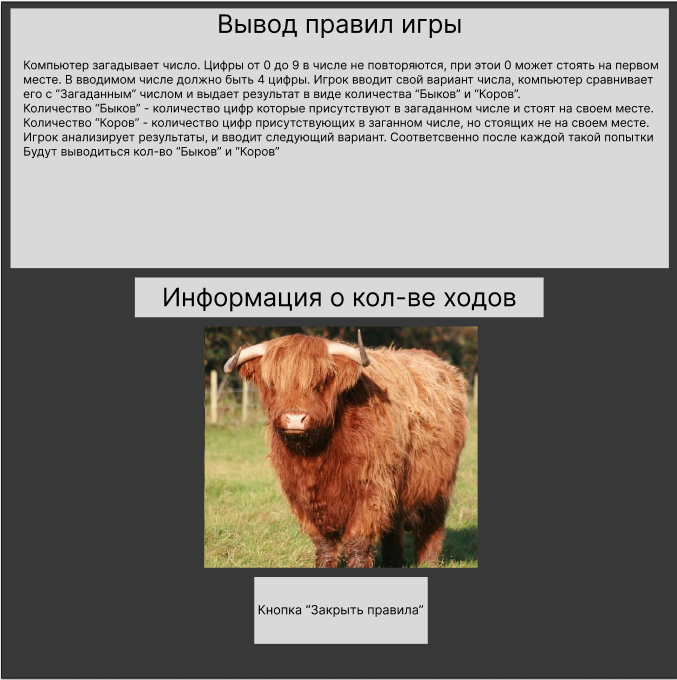


Рисунок 10 – Прототип экранной формы №2

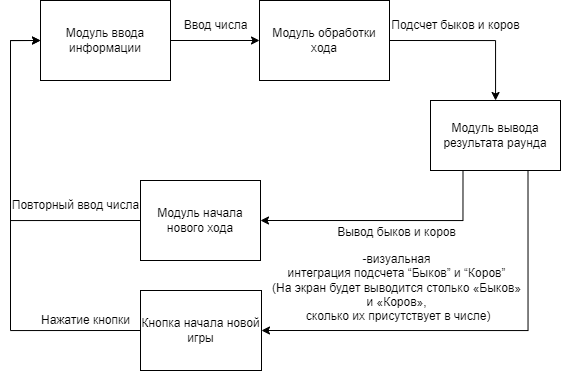
****

Рисунок 11 – Модульная структура

Модуль — функционально законченный фрагмент программы, оформленный в виде отдельного файла с исходным кодом.

Модуль ввода информации – это модуль в котором пользователь вводит число на устройстве считывания информации, тем самым оно заносится в программу.

Модуль обработки хода – это модуль в котором компьютер сравнивает загаданное число с введенным пользователем, исходя из условий проверки. И подсчитывает число «Быков» и «Коров».

Модуль вывода результата раунда – это модуль благодаря которому пользователь может увидеть результат раунда, программа выведет ему количество «Быков» и «Коров», а так же их картинки в количестве того, сколько их в веденом числе.

Модуль начала нового хода – это модуль в котором после вывода «Быков» и «Коров» игрок должен повторно ввести число.

Модуль начала новой игры – это модуль в котором после вывода результата раунда пользователь наживает на кнопку «Новая игра» и после этого загаданное число генерируется заново, а результаты прошлой игры аннулируются

**4.3 Реализация программы. Разработка пользовательского интерфейса**

На игровом поле использовались такие элементы как Label (Например: наименование программы, вывод реального времени (по МСК), информация о том как открыть игру на полный экран и как вернуть ее в прежнее положение.)

Так же использовался элемент Button (Например: кнопка «Новая игра», «Закрыть приложение», «Открыть правила».)

А также Entry (Например: поле для ввода числа.) Все это можно посмотреть на рисунке 12.

Используются такие элементы как Canvas (холст), они используются для визуализации количества «Быков» и «Коров», исходя из того сколько их присутствует в числе

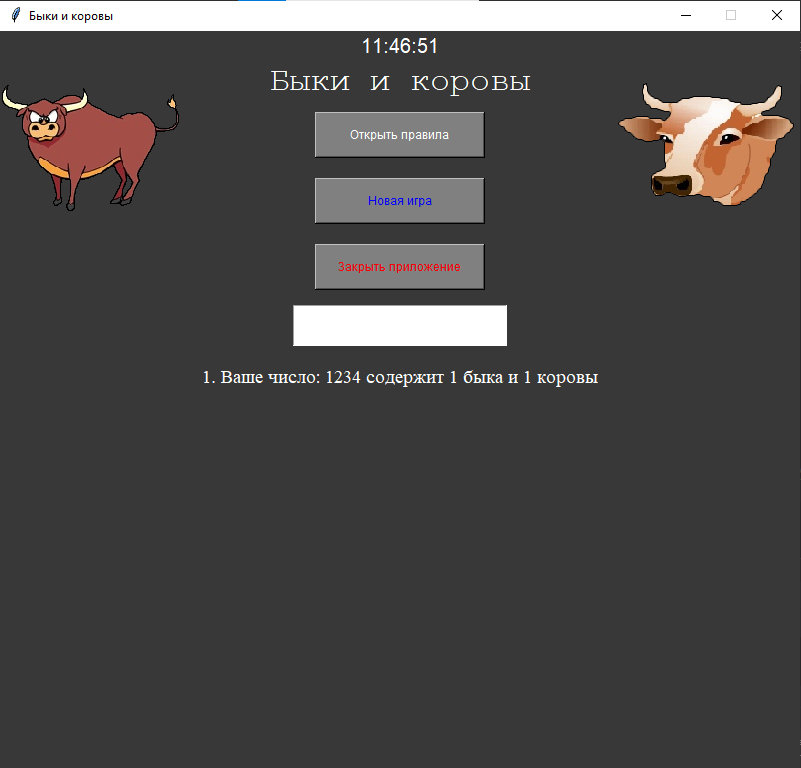
****

Рисунок 12 – Игровое поле игры «Быки и коровы»

Ниже будут приведены примеры 4 «Быков и 4 «Коров» (см.рис 13 и см.рис 14)

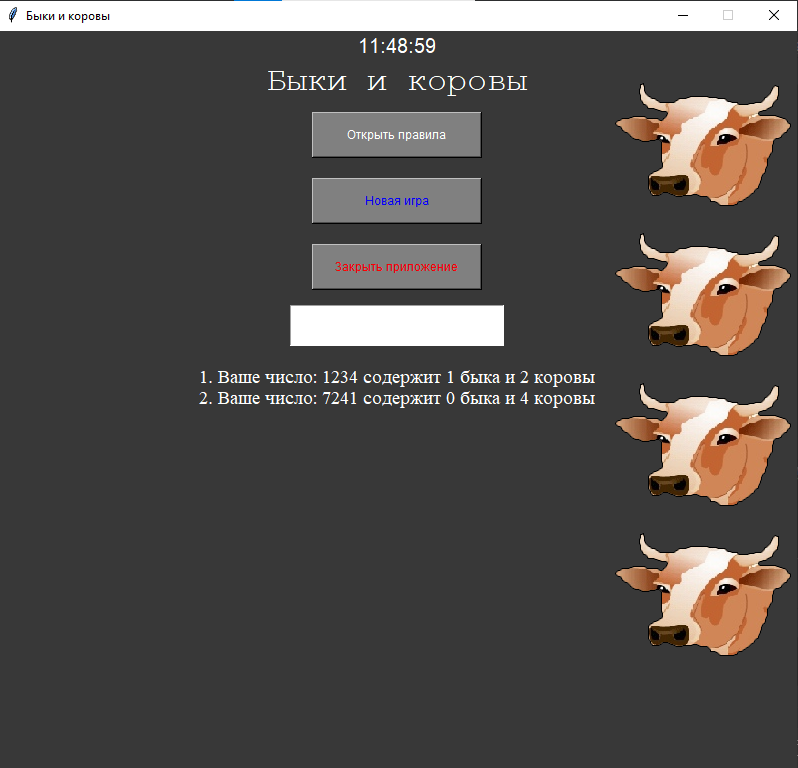


Рисунок 13 – Пример наличия 4 «Коров» в веденом числе

На данном рисунке, после ввода числа и проверки всех условий, сделался вывод что в веденом числе 4 «Коровы», в следствии чего на экране появилось 4 соответствующих изображения

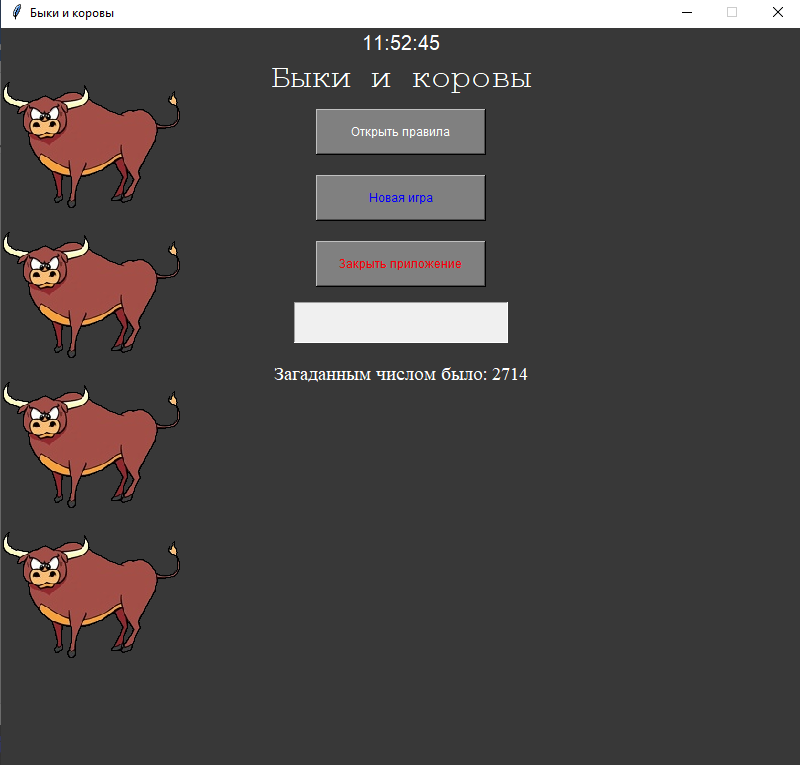


Рисунок 14 – Пример наличия 4 «Быков» в веденом числе

На данном рисунке, после ввода числа и проверки всех условий, сделался вывод что в веденом числе 4 «Быка», в следствии чего на экране появилось 4 соответствующих изображения и окно о победе.

Так как у нас есть кнопка «Открыть правила», соответственно исходя из ее названия, у нас открывается новое окно. (см рис.15)

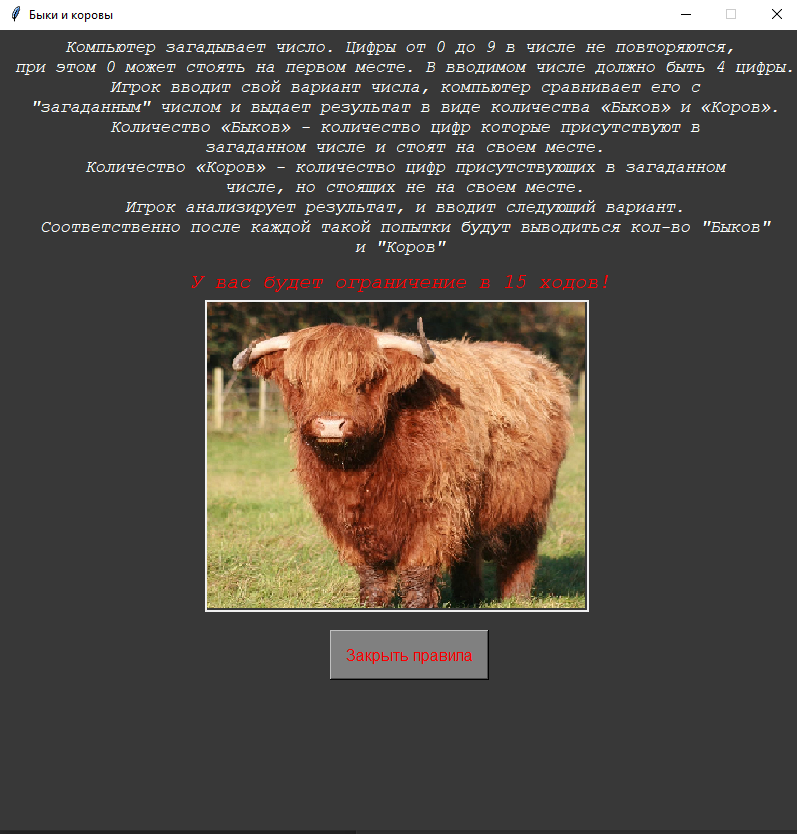
****

Рисунок 15 – Окно с правила в игры «Быки и коровы»

В нем у нас так же имеются такие элементы как два Lable, в первом у нас выводятся сами правила игры, а во втором информация о том сколько у игрока осталось ходов. Еще присутствует элемент Button, он отвечает за закрытие окна с правила, и непосредственным возвратом к основному окну с игрой, а также использовался Canvas (холст) для вывода картинки

Фрагмент кода для проверки числа на правильность:



Рисунок 16 – Фрагмент кода

Функция создания нового окна:



Рисунок 17 – Фрагмент кода №2

Функция вывода картинок «Быков» и «Коров»:



Рисунок 18 – Фрагмент кода №3

**4.4 Тестирование программы**

Таблица 1 – Протокол испытаний

| № | Шаг испытаний (проверок) | № пункта  Методики | Отметка о прохождении (да/нет) | Примечания |
| --- | --- | --- | --- | --- |
| 1 | Предоставление «Технического задания» отвечающего на все требования | 1 | Да | - |
| 2 | Предоставление «Руководства пользователя» отвечающего на все требования | 1 | Да | - |
| 3 | Предоставление настоящей программы и «Методики тестирования» отвечающего на все требования | 1 | Да | - |
| 4 | Создание нового окна после нажатия на кнопку «Открыть правила» | 2 | Да | - |
| 5 | Создание прописанных в коде форм (таких как текст, картинка, кнопка «Закрыть правила») | 2 | Да | - |
| 6 | Блокировка кнопки «Открыть правила», после нажатия на нее и создания нового окна | 2 | Да | - |
| 7 | Полное уничтожение окна с правила, разблокировка кнопки «Открыть правила», после нажатия на кнопку «Закрыть правила» | 3 | Да | - |
| 8 | Вывод диалогового окна с вопросом о том, хотел бы игрок начать новую игру | 4 | Да | - |
| 9 | Очистка поля с результатами игры после положительного ответа в диалоговом окне | 4 | Да | - |
| 10 | Генерация нового числа компьютером исходя из всех условий после положительного ответа в диалоговом окне | 4 | Да | - |
| 11 | Возвращение в той игровой стадий на которой находился игрок, при отрицательном ответе в диалоговом окне | 4 | Да | - |
| 12 | Вывод диалогового окна с вопросом о том, хотел бы игрок закрыть приложение, после нажатия на кнопку «Закрыть правила» | 5 | Да | - |
| 13 | Полное закрытие приложения при положительном ответе в диалоговом окне | 5 | Да | - |
| 14 | Возвращение в той игровой стадий на которой находился игрок, при отрицательном ответе в диалоговом окне | 5 | Да | - |
| 15 | Загаданное число полностью удовлетворяет условиям, прописанными правилами игры | 6 | Да | - |
| 16 | При вводе 0567 и при загаданном числе в виде 0123, на экран будет выведена 1 картинка «Быка» | 7 | Да | - |
| 17 | При вводе 5670 и загаданном числе в виде 0123, на экран будет выведена 1 картинка «Коровы» | 8 | Да | - |
| 18 | Вывод ошибки при наличии «Пробела» в веденном числе | 9 | Да | - |
| 19 | Вывод ошибки при наличии «Спецсимволов» в веденном числе | 9 | Да | - |
| 20 | Вывод ошибки при наличии «Букв» в веденном числе | 9 | Да | - |
| 21 | Вывод ошибки при наличии менее 4-ех символов в веденном числе | 9 | Да | - |
| 22 | Вывод ошибки при наличии более 4-ех цифр в веденном числе | 9 | Да | - |
| 23 | При вводе 1234 и загаданном числе в виде 1537, в поле для вывода результатов, должно быть оглашено, что в веденном числе 2 «Быка» и «0» коров | 10 | Да | - |

Экранные формы с проведением проверок:

1. Проверка работоспособности кнопки «Открыть правила»

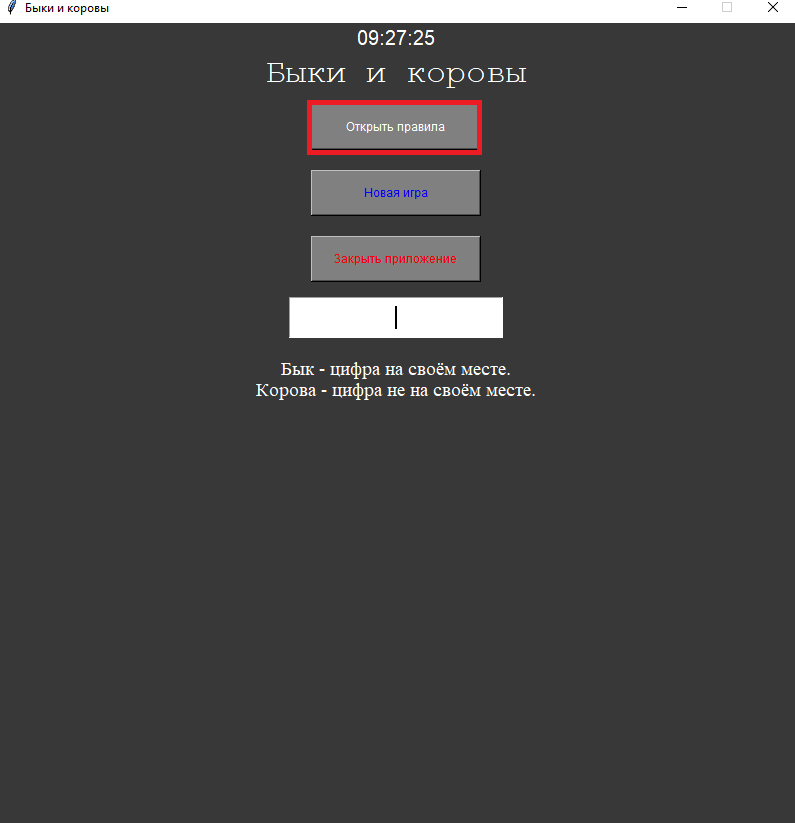


Рисунок 19 – Нажатие на кнопку «Открыть правила»

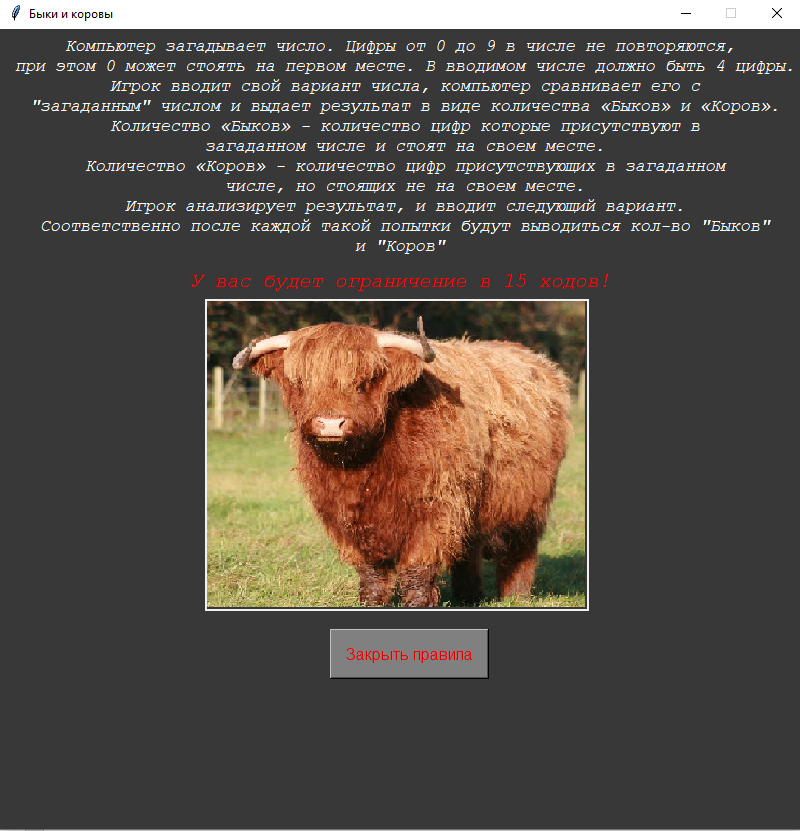


Рисунок 20 – Открытое в результате нажатия на кнопку окно с правилами

1. Проверка работоспособности кнопки закрыть правила:

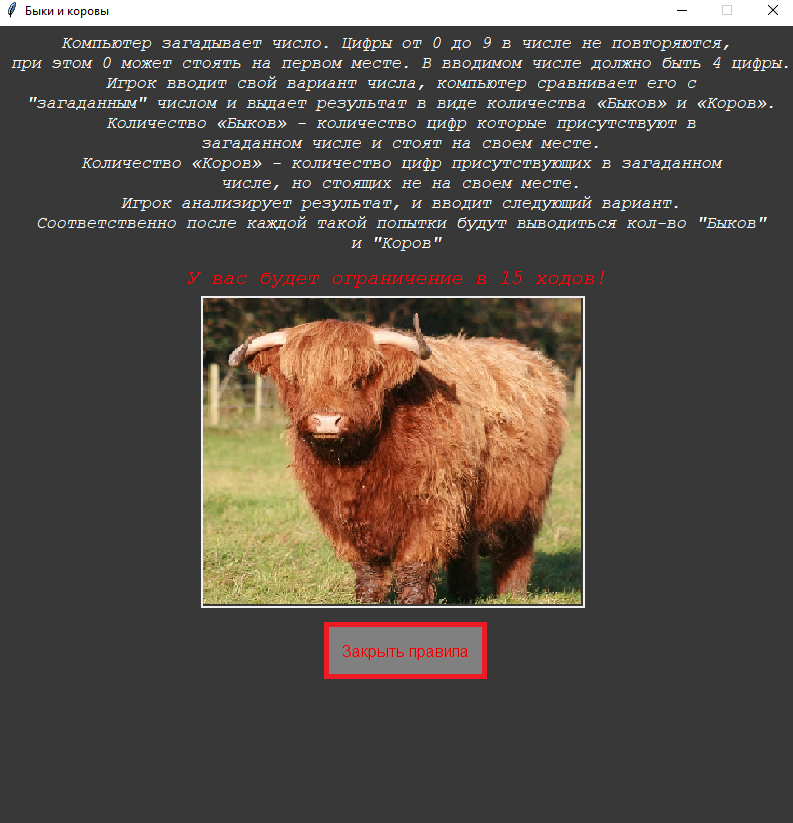


Рисунок 21 – Нажатие на кнопку «Закрыть правила»

В следствии нажатия на данную кнопку, окно с правилами было полостью закрыто.

1. Проверка работоспособности кнопки «Новая игра»

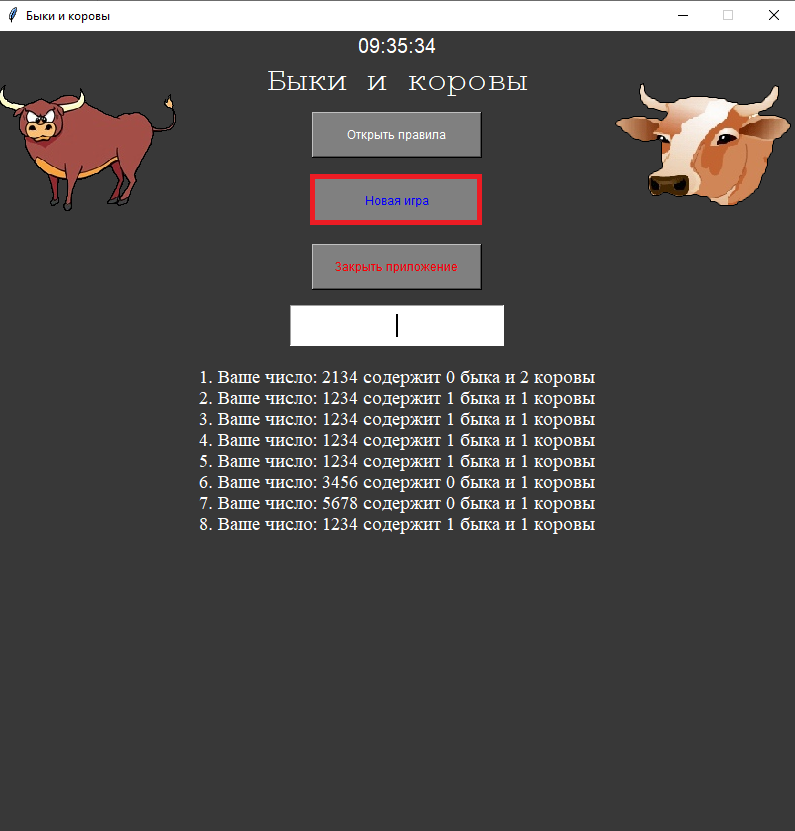


Рисунок 22 – Нажатие на кнопку «Новая игра» во время игрового процесса

В следствии нажатия на данную кнопку, результат прошлой игры полностью обнулился, а игровое поле очистилось (см.рисунок 20)

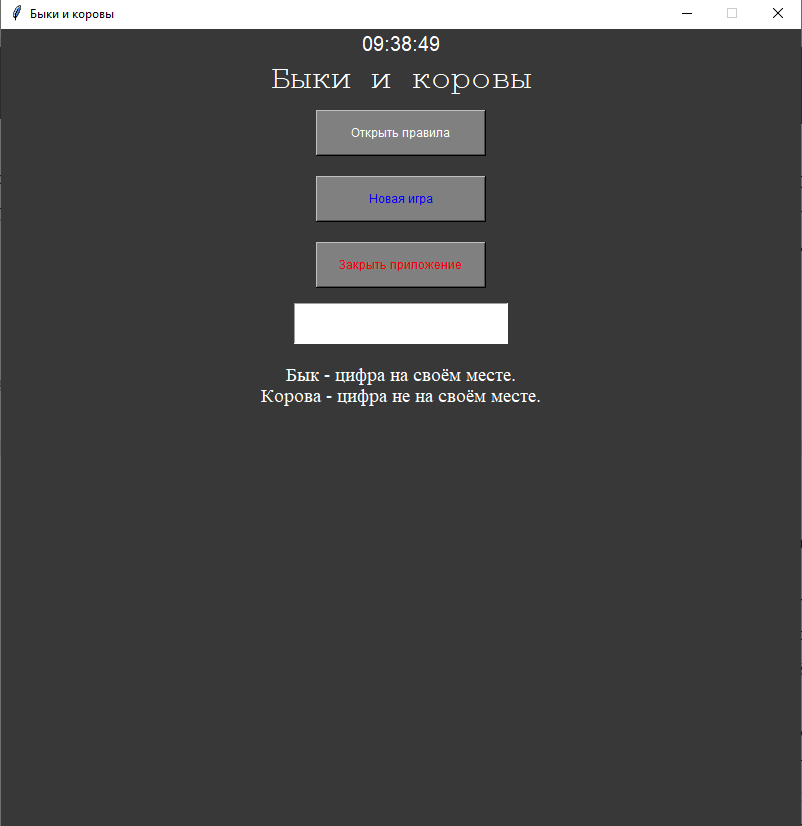


Рисунок 23 – Результат работы кнопки «Новая игра»

Проверка работоспособности кнопки «Закрыть приложение»

1. Проверка работоспособности кнопки «Закрыть приложение»

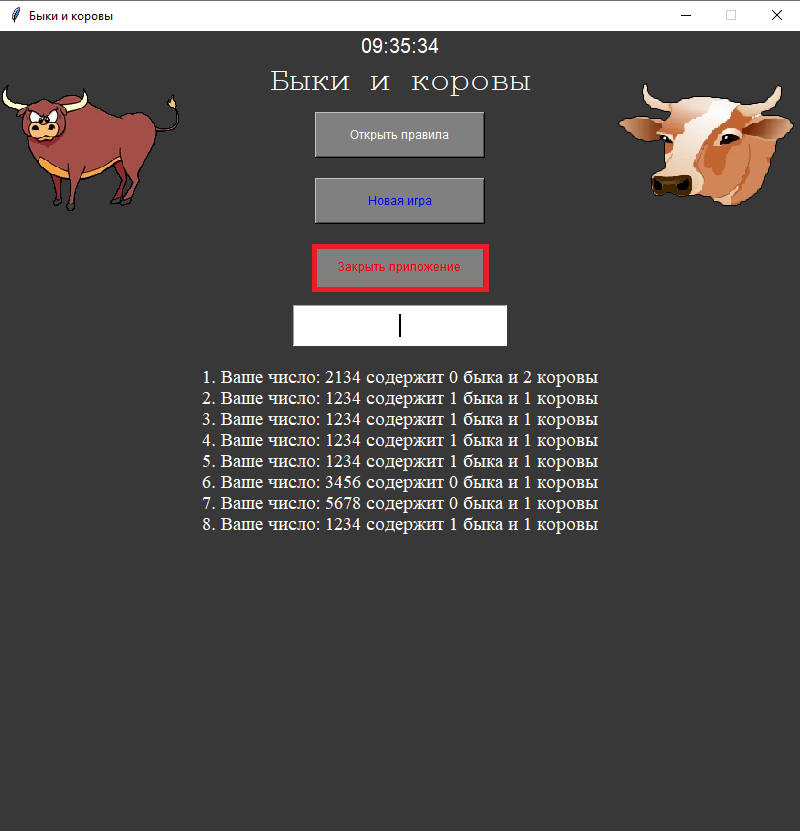


Рисунок 24 – Нажатие на кнопку «Закрыть приложение»

В следствии нажатия на данную кнопку, приложение было полностью закрыто.

1. Проверка правильности генерации числа

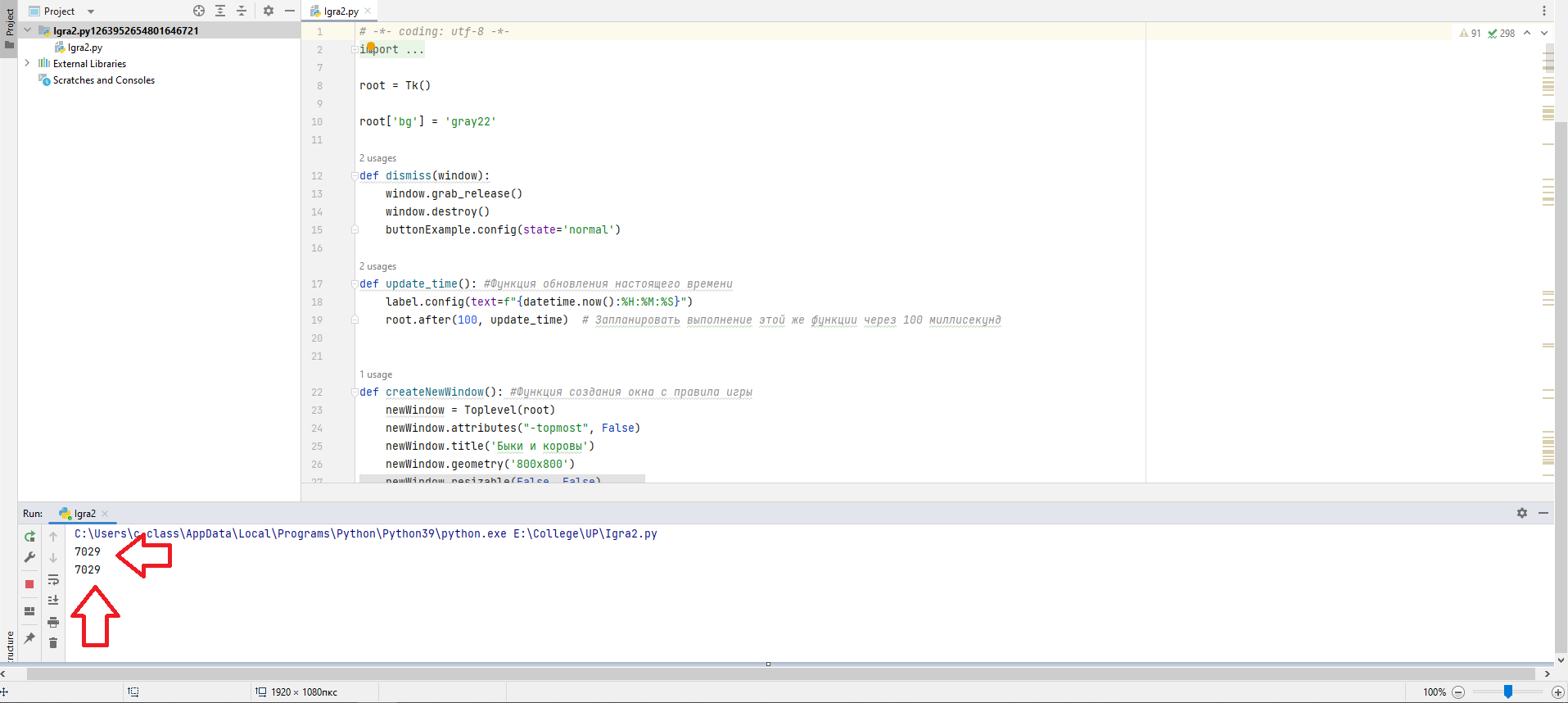


Рисунок 25 – Вывод в консоль загаданного числа

Данная проверка проводилась множество раз, и был сделан вывод, что сгенерированное число соответствует всем условиям.

1. Проверка вывода картинок «Быков»

На игровое поле должно выводится столько «Быков», сколько их в веденном числе

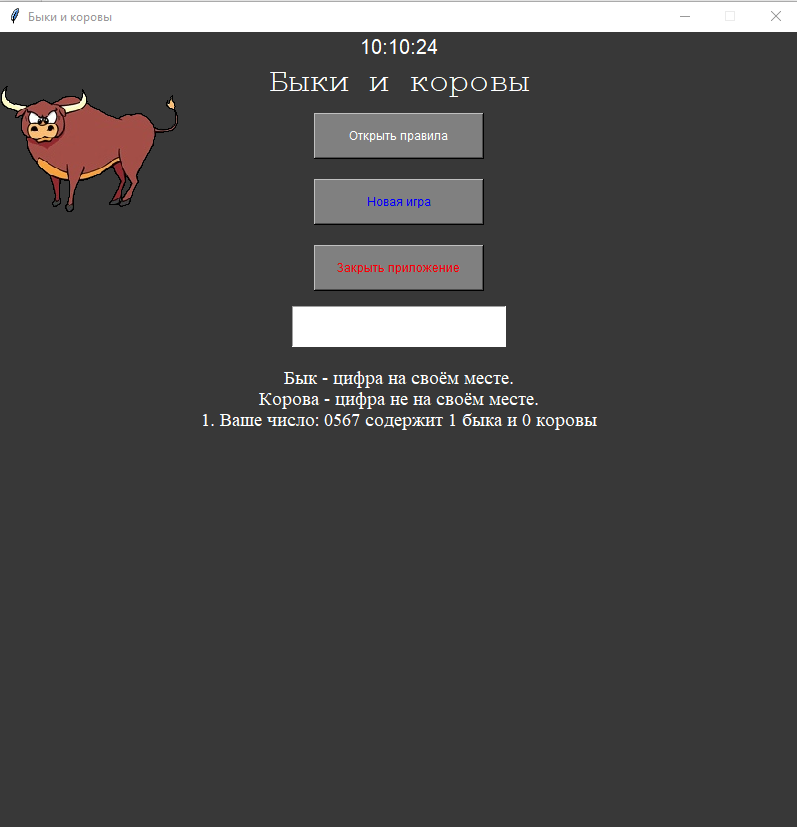


Рисунок 26 – Пример работы вывода картинок «Быков» на игровое поле

1. Проверка вывода картинок «Коров»

На игровое поле должно выводится столько «Коров», сколько их в веденном числе

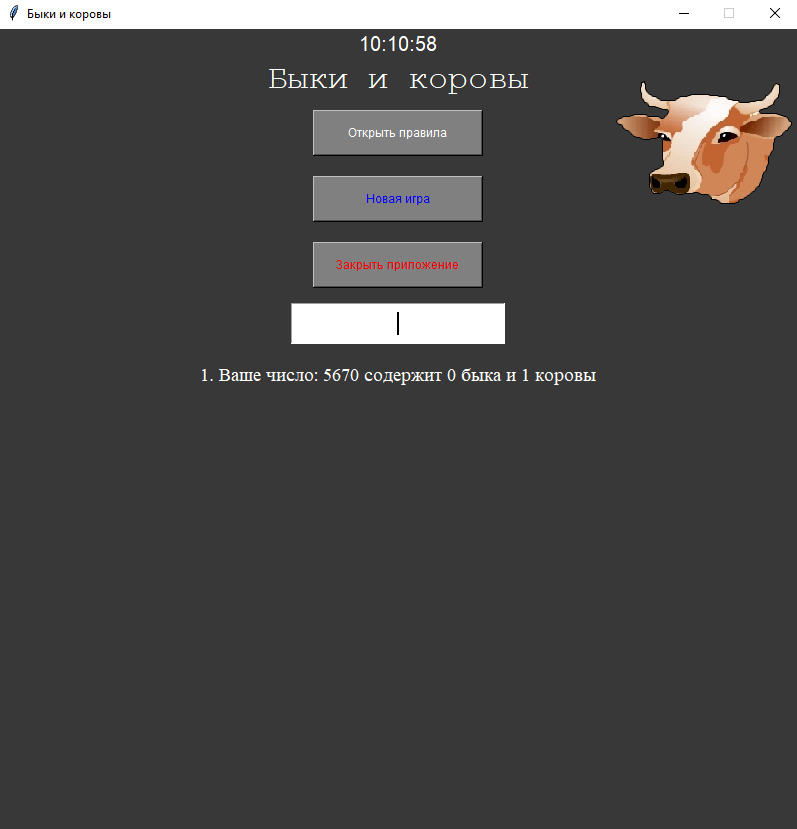


Рисунок 27 – Пример работы вывода картинок «Быков» на игровое поле

1. Проверка работоспособности проверок, на выполнения условий ввода числа

Выполнение проверки на наличие одинаковых символов

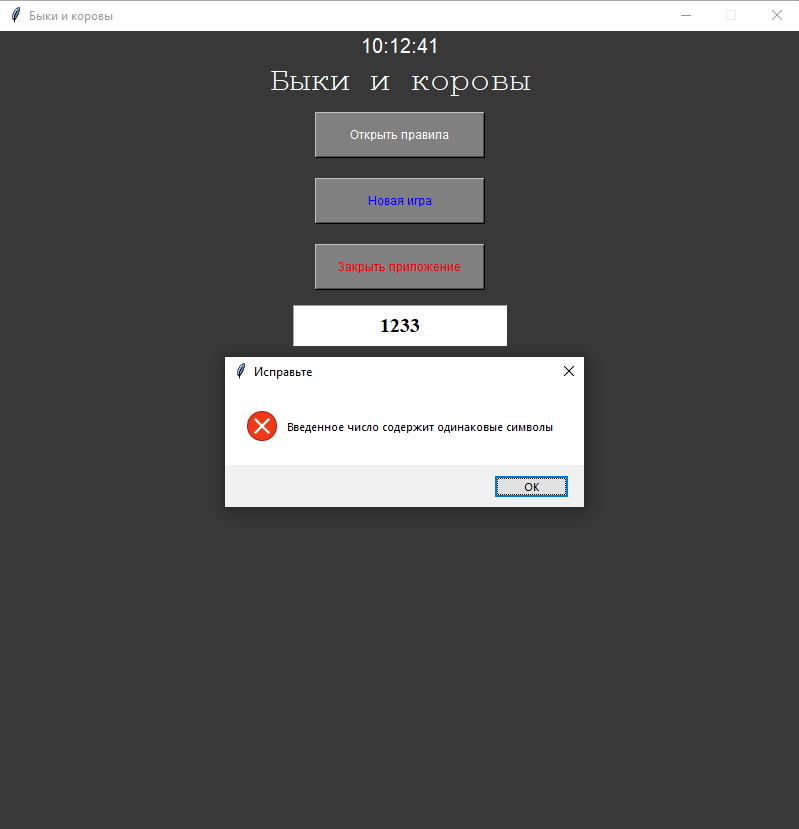


Рисунок 28 –Результат проверки на наличие одинаковых символов

Выполнение проверки на наличие пробелов

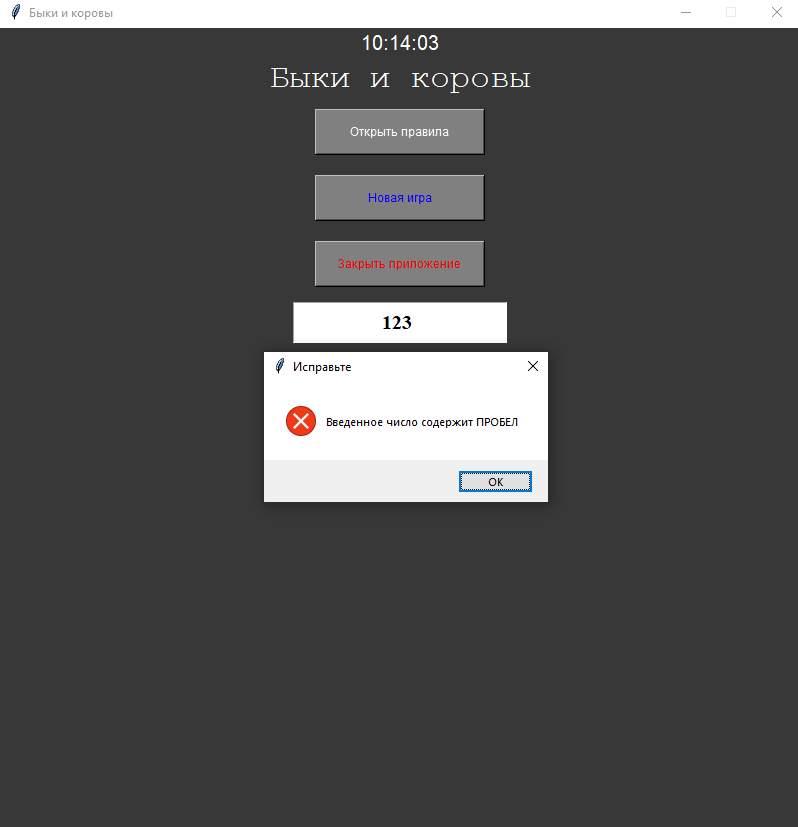


Рисунок 29 – Результат проверки на наличие пробелов

Выполнение проверки на наличие спецсимволов

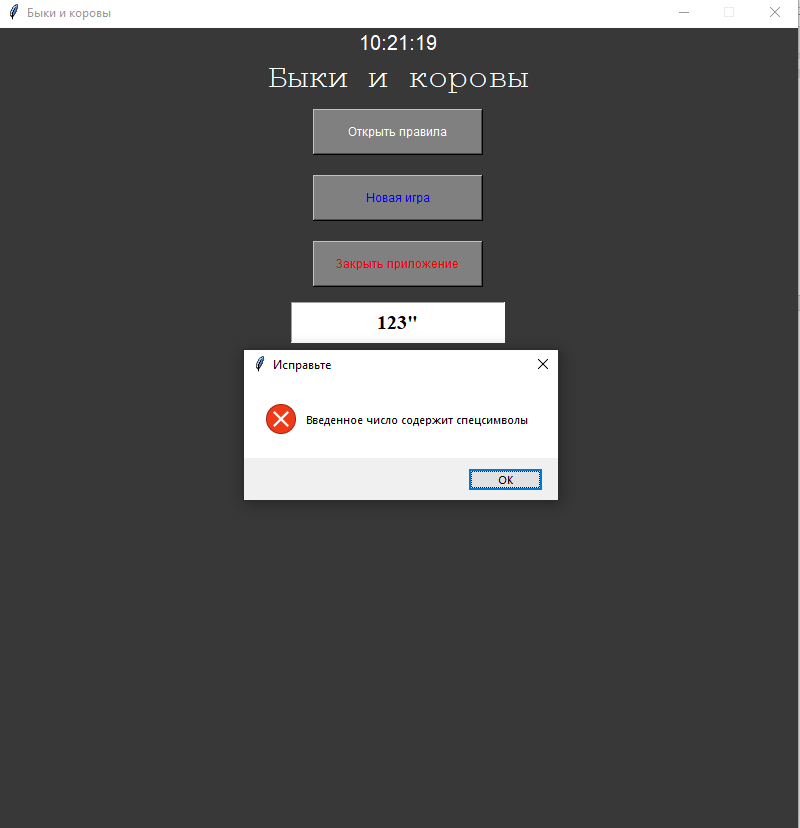


Рисунок 30 – Результат проверки на наличие спецсимволов

Выполнение проверки на то, что в числе больше 4 символов

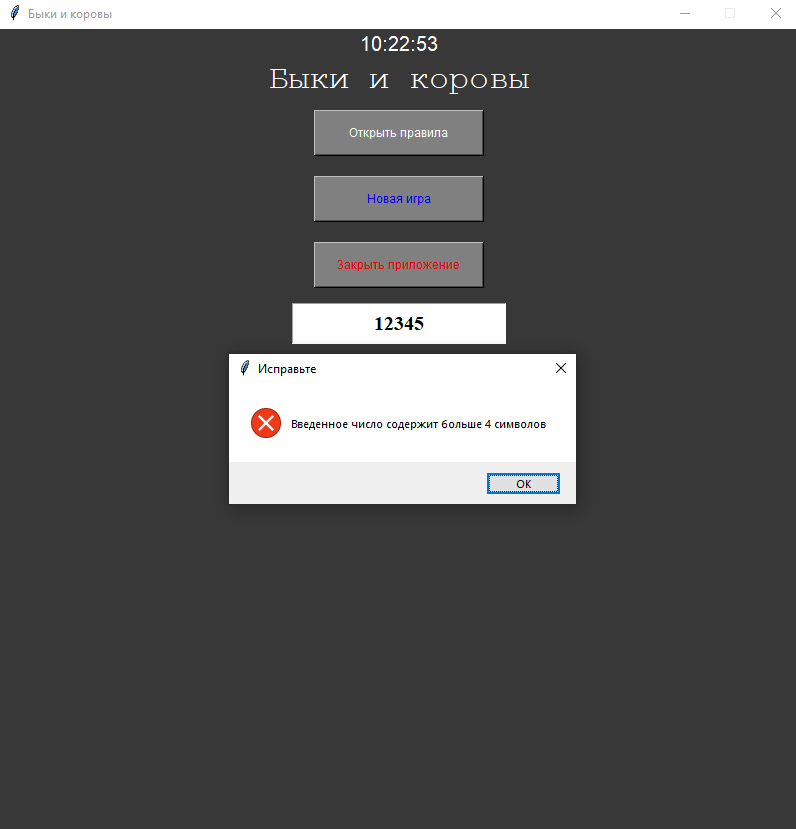


Рисунок 31 – Результат проверки на наличие более 4 символов в веденном числе

Выполнение проверки на то, что в числе меньше 4 символов

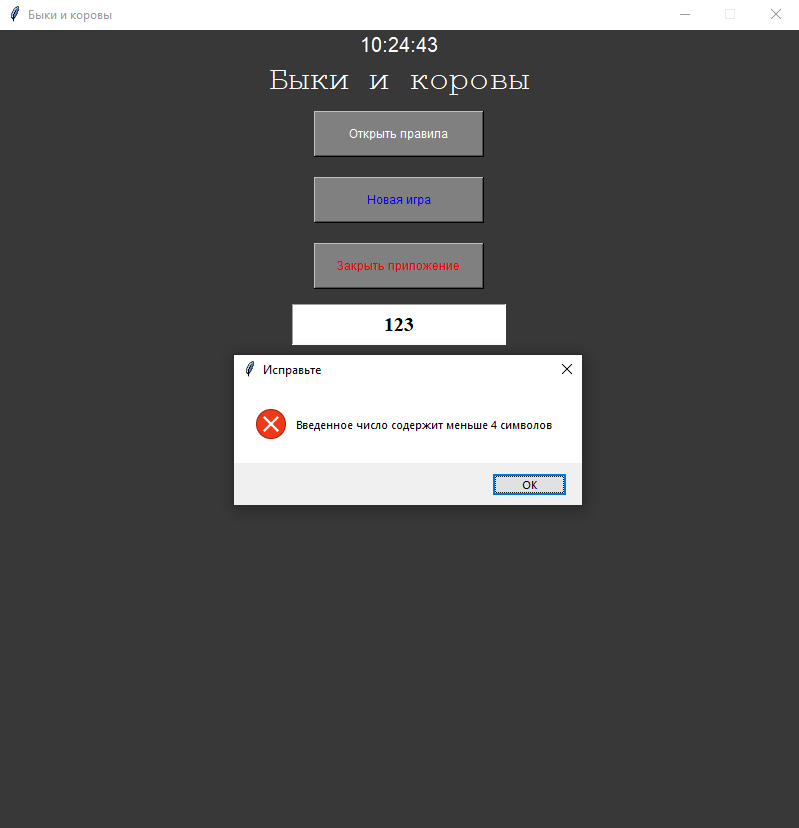


Рисунок 32 – Результат проверки на наличие менее 4 символов в веденном числе

Выполнение проверки на наличие букв

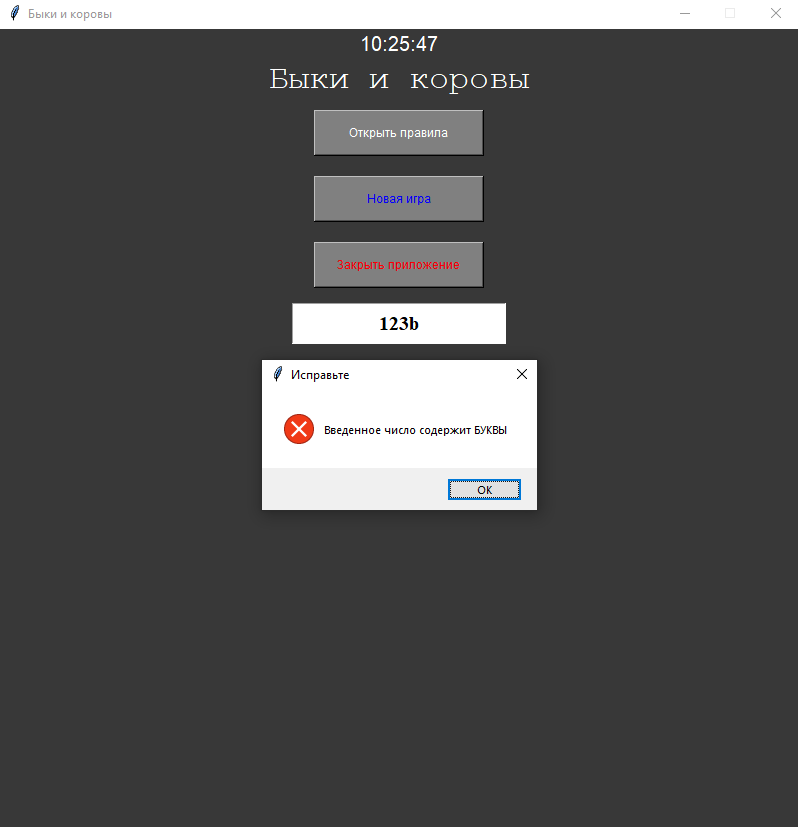


Рисунок 33 – Результат выполнения проверки на наличие букв

1. Проверка на правильность вывода результатов хода

При вводе числа, на определенной области в середине игрового поля, должны выводиться результаты игры, где оговаривается, сколько в веденном числе “Быков” и “Коров”

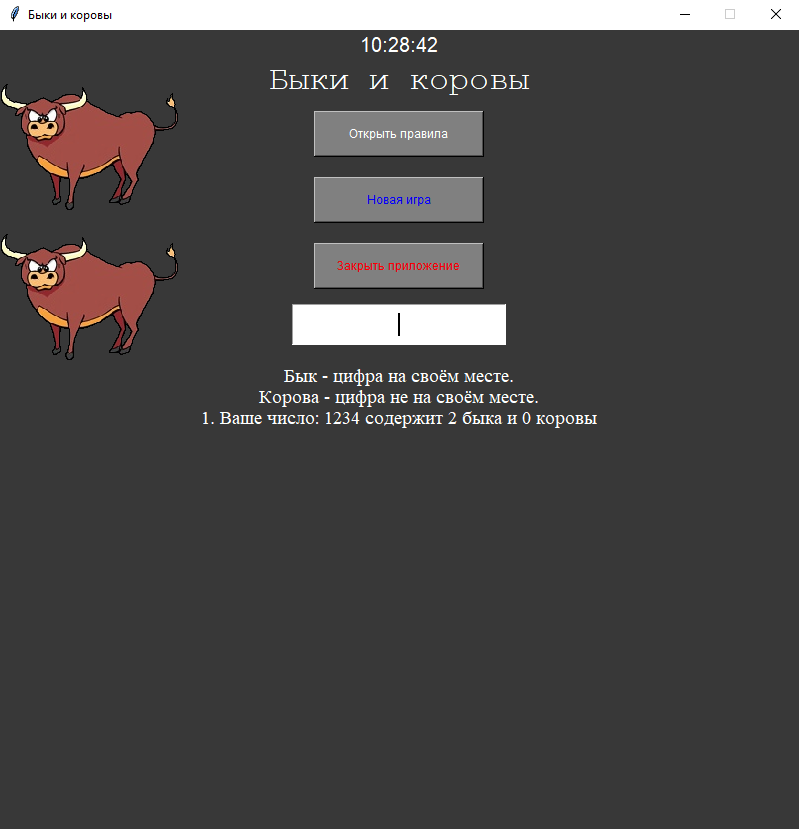


Рисунок 34 – Результат выполнение проверки на правильность вывода результата хода

# **ЗАКЛЮЧЕНИЕ**

Таким образом, программа учебной практики ПМ.05 в период с 11.01.2021 по 16.06.2021 была выполнена мной в полном объеме.

При выполнении работ, предусмотренных заданием на практику, в организации ФГБОУ ВО «Вятский государственный университет» мной, Ананьиным Кириллом Сергеевичем, соблюдались правила внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологические правила и гигиенические нормативы. Пройден вводный инструктаж и инструктаж на рабочем месте. Замечаний не имел.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. tkinter — Python interface to Tcl/Tk [Электронный ресурс] – URL: <https://docs.python.org/3/library/tkinter.html> (20.03.2023)
2. Позиционирование. Pack [Электронный ресурс] – URL: <https://metanit.com/python/tkinter/2.4.php> (22.03.2023)
3. Окна. Создания окон [Электронный ресурс] – URL: <https://metanit.com/python/tkinter/5.1.php> (25.03.2023)
4. Canvas. Добавление элементов на Canvas [Электронный ресурс] – URL:  [https://metanit.com/python/tkinter/7.1.php](%20https:/metanit.com/python/tkinter/7.1.php) (26.03.2023)
5. Поле ввода Entry [Электронный ресурс] – URL: <https://metanit.com/python/tkinter/2.8.php> (26.03.2023)

# **ПРИЛОЖЕНИЕ А**

**ИСХОДНЫЙ КОД**

*# -\*- coding: utf-8 -\*-*import tkinter  
from tkinter import \*  
from random import choice  
from tkinter import messagebox  
from datetime import datetime  
  
root = Tk()  
  
root['bg'] = 'gray22'  
  
def dismiss(window):  
 window.grab\_release()  
 window.destroy()  
 buttonExample.config(state='normal')  
  
def update\_time(): *#Функция обновления настоящего времени* label.config(text=f"{datetime.now():%H:%M:%S}")  
 root.after(100, update\_time) *# Запланировать выполнение этой же функции через 100 миллисекунд*def createNewWindow(): *#Функция создания окна с правила игры* print(x)  
 newWindow = Toplevel(root)  
 newWindow.attributes("-topmost", False)  
 newWindow.title('Быки и коровы')  
 newWindow.geometry('800x800')  
 newWindow.resizable(False, False)  
 newWindow['bg'] = 'gray22'  
 newWindow.protocol("WM\_DELETE\_WINDOW", lambda: dismiss(newWindow))  
 buttonExample.config(state='disabled')  
 labelExample = tkinter.Label(newWindow, width=400, padx=10, pady=5, justify="center", text = 'Компьютер загадывает число. Цифры от 0 до 9 в числе не повторяются,\n при этом 0 может стоять на первом месте. В вводимом числе должно быть 4 цифры.\n Игрок вводит свой вариант числа, компьютер сравнивает его с\n "загаданным" числом и выдает результат в виде количества «Быков» и «Коров».\n Количество «Быков» - количество цифр которые присутствуют в\n загаданном числе и стоят на своем месте.\n Количество «Коров» - количество цифр присутствующих в загаданном\n числе, но стоящих не на своем месте.\n Игрок анализирует результат, и вводит следующий вариант.\n Соответственно после каждой такой попытки будут выводиться кол-во "Быков"\n и "Коров" ')  
 labelExample.pack(anchor=N)  
 labelExample.config(fg='white',  
 font=('Courier', 13, 'italic'), bg='gray22')  
 label2 = tkinter.Label(newWindow, width=400, padx=10, pady=5, justify="center", text='У вас будет ограничение в 15 ходов!')  
 label2.pack(anchor=N)  
 label2.config(fg='red',  
 font=('Courier', 15, 'italic'), bg='gray22')  
 close\_button = tkinter.Button(newWindow, text="Закрыть правила",bg='grey', fg='red', font='Helvetica 12', command=lambda: dismiss(newWindow), pady=10, padx=10)  
 close\_button.place(x=330, y=600)  
 can = Canvas(newWindow,bg='gray22', width=380, height=308, highlightthickness=2)  
 can.place(x=205, y=270)  
 can.create\_image(0, 0, image=png4, anchor=NW)  
  
window\_width = 800  
window\_height = 800  
screen\_width = root.winfo\_screenwidth()  
screen\_height = root.winfo\_screenheight()  
o = int((screen\_width / 2) - (window\_width / 2))  
q = int((screen\_height / 2) - (window\_height / 2))  
root.geometry("{}x{}+{}+{}".format(window\_width, window\_height, o, q))  
label = tkinter.Label(root, font=("helvetica", 15), bg='gray22', fg='white')  
label.pack()  
update\_time()  
  
x = ''  
n = 0  
k = 0  
j = 0  
p = ('~', '`', '!', '@', '#', '№', '$', '%', '^', '&', '?', '\*', '(', ')', '\_', '-', '+', '/', '>', '<', ':', ';', '"', ".", "|", "[","]","{","}","'","^",",", '\\')  
  
def on\_closing(): *#Функция вывода окна с вопросом, и выход из приложения при положительном ответе* if messagebox.askokcancel('Выход из приложения', 'Хотите выйти из приложения?'):  
 root.destroy()  
  
def play(event): *#Основная игровая функция, привязанная к Entry* global x, n, k, p, m  
 m = 16  
 if len(x) == 0: *#Генерация числа* z = '0123456789'  
 x = choice(z[1:10])  
 for i in range(3):  
 z = ''.join(z.split(x[i]))  
 x += choice(z)  
 sms2.config(text='')  
 y = Entry.get()  
 print(x)  
 can = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can.place(x=0, y=50)  
 can.create\_image(0, 0, image=png3, anchor=NW)  
 can2 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can2.place(x=0, y=200)  
 can2.create\_image(0, 0, image=png3, anchor=NW)  
 can3 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can3.place(x=0, y=350)  
 can3.create\_image(0, 0, image=png3, anchor=NW)  
 can4 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can4.place(x=0, y=500)  
 can4.create\_image(0, 0, image=png3, anchor=NW)  
 can5 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0) *#Все холсты после выполнения хода закрашиваются фоном приложения* can5.place(x=615, y=50)  
 can5.create\_image(0, 0, image=png3, anchor=NW)  
 can6 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can6.place(x=615, y=200)  
 can6.create\_image(0, 0, image=png3, anchor=NW)  
 can7 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can7.place(x=615, y=350)  
 can7.create\_image(0, 0, image=png3, anchor=NW)  
 can8 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can8.place(x=615, y=500)  
 can8.create\_image(0, 0, image=png3, anchor=NW)  
  
 b = 0  
 c = 0  
 k = 0  
 if y == '': *# Проверка на то пустое ли поле ввода* messagebox.showerror("Исправьте", "Поле для ввода пусто")  
 Entry.delete(0, END)  
 return  
 if len(y) > 4: *# Проверка на кол-во символом* messagebox.showerror("Исправьте", "Введенное число содержит больше 4 символов")  
 k = k + 1  
 Entry.delete(0, END)  
 return  
 elif len(y) < 4: *# Проверка на кол-во символом* messagebox.showerror('Исправьте', 'Введенное число содержит меньше 4 символов')  
 k = k + 1  
 Entry.delete(0, END)  
 else:  
 for i in range(len(y)): *# Проверка на наличие пробелов* if y[i-1] == ' ':  
 messagebox.showerror("Исправьте", "Введенное число содержит ПРОБЕЛ")  
 k = k+1  
 Entry.delete(0, END)  
 return  
 elif (y[i] == y[(len(y) - i)-1] or y[2] == y[3] or y[1] == y[2] or y[0] == y[1] or y[1] == y[3] or y[0] == y[2]): *# Проверка на уникальность* messagebox.showerror("Исправьте", "Введенное число содержит одинаковые символы")  
 k = k+1  
 Entry.delete(0, END)  
 return  
 elif y[i] in p: *#Проверка на спецсимволы* messagebox.showerror('Исправьте', 'Введенное число содержит спецсимволы')  
 k = k + 1  
 Entry.delete(0, END)  
 return  
 if y.isalnum() == True: *# Проверка на наличие букв* if y.isnumeric() == False:  
 messagebox.showerror("Исправьте", "Введенное число содержит БУКВЫ")  
 k = k + 1  
 Entry.delete(0, END)  
 if k==0: *#Проверка на кол-во "Быков" и "Коров"* for i in range(4):  
 if x[i] == y[i]:  
 b += 1  
 elif y[i] in x:  
 c += 1  
 if b == 1:  
 can = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can.place(x=0, y=50)  
 can.create\_image(0, 0, image=png, anchor=NW)  
 if b == 2:  
 can = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can.place(x=0, y=50)  
 can.create\_image(0, 0, image=png, anchor=NW)  
 can2 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can2.place(x=0, y=200)  
 can2.create\_image(0, 0, image=png, anchor=NW)  
 if b == 3:  
 can = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can.place(x=0, y=50)  
 can.create\_image(0, 0, image=png, anchor=NW)  
 can2 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can2.place(x=0, y=200)  
 can2.create\_image(0, 0, image=png, anchor=NW)  
 can3 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can3.place(x=0, y=350)  
 can3.create\_image(0, 0, image=png, anchor=NW)  
 if b == 4:  
 can = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can.place(x=0, y=50)  
 can.create\_image(0, 0, image=png, anchor=NW)  
 can2 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can2.place(x=0, y=200)  
 can2.create\_image(0, 0, image=png, anchor=NW)  
 can3 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can3.place(x=0, y=350)  
 can3.create\_image(0, 0, image=png, anchor=NW)  
 can4 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can4.place(x=0, y=500)  
 can4.create\_image(0, 0, image=png, anchor=NW)  
 if c == 1:  
 can5 = Canvas(bg='gray22', width=185, height=183, highlightthickness=0)  
 can5.place(x=615, y=50)  
 can5.create\_image(0, 0, image=png2, anchor=NW)  
 if c == 2:  
 can5 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can5.place(x=615, y=50)  
 can5.create\_image(0, 0, image=png2, anchor=NW)  
 can6 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can6.place(x=615, y=200)  
 can6.create\_image(0, 0, image=png2, anchor=NW)  
 if c == 3:  
 can5 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can5.place(x=615, y=50)  
 can5.create\_image(0, 0, image=png2, anchor=NW)  
 can6 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can6.place(x=615, y=200)  
 can6.create\_image(0, 0, image=png2, anchor=NW)  
 can7 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can7.place(x=615, y=350)  
 can7.create\_image(0, 0, image=png2, anchor=NW)  
 if c == 4:  
 can5 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can5.place(x=615, y=50)  
 can5.create\_image(0, 0, image=png2, anchor=NW)  
 can6 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can6.place(x=615, y=200)  
 can6.create\_image(0, 0, image=png2, anchor=NW)  
 can7 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can7.place(x=615, y=350)  
 can7.create\_image(0, 0, image=png2, anchor=NW)  
 can8 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can8.place(x=615, y=500)  
 can8.create\_image(0, 0, image=png2, anchor=NW)  
 if k == 0: *#Вывод информации о результате хода* n += 1  
 sms2.config(text=sms2.cget('text') + str(n) + '. Ваше число: ' + y + ' содержит ' + str(b) + ' быка и ' + str(c) + ' коровы\n')  
 Entry.delete(0, END)  
 else:  
 k = 0  
 Entry.delete(0, END)  
 if n > 15: *#То, что будет происходить при поражении* messagebox.showinfo('Поражение', 'Вы не смогли отгадать число, им было: ' + x)  
 answer2 = messagebox.askyesno(title='Новая игры', message='Хотите начать новую игру?')  
 sms2.config(text='')  
 sms2.config(text=sms2.cget('text') + 'Загаданным числом было : ' + x + '\n Было затраченно: ' + str(n) + ' ходов \n ')  
 if answer2:  
 x = ''  
 n = 0  
 sms2.config(text='Бык - цифра на своём месте.\n'  
 'Корова - цифра не на своём месте.')  
 can = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can.place(x=0, y=50)  
 can.create\_image(0, 0, image=png3, anchor=NW)  
 can2 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can2.place(x=0, y=200)  
 can2.create\_image(0, 0, image=png3, anchor=NW)  
 can3 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can3.place(x=0, y=350)  
 can3.create\_image(0, 0, image=png3, anchor=NW)  
 can4 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can4.place(x=0, y=500)  
 can4.create\_image(0, 0, image=png3, anchor=NW)  
 can5 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can5.place(x=615, y=50)  
 can5.create\_image(0, 0, image=png3, anchor=NW)  
 can6 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can6.place(x=615, y=200)  
 can6.create\_image(0, 0, image=png3, anchor=NW)  
 can7 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can7.place(x=615, y=350)  
 can7.create\_image(0, 0, image=png3, anchor=NW)  
 can8 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can8.place(x=615, y=500)  
 can8.create\_image(0, 0, image=png3, anchor=NW)  
 else:  
 Entry.unbind('<Return>')  
 Entry.config(state='disabled')  
 if x == y: *#То, что будет происходить при победе* sms2.config(text='')  
 sms2.config(text=sms2.cget('text') + 'Загаданным числом было: ' + x + '\nБыло затраченно: ' + str(n) + ' ходов ')  
 messagebox.showinfo('Победа!', 'Поздравляем, вы отгадали число.')  
 answer = messagebox.askyesno(title='Новая игры', message='Хотите начать новую игру?')  
 if answer:  
 x = ''  
 n = 0  
 sms2.config(text='Бык - цифра на своём месте.\n'  
 'Корова - цифра не на своём месте.')  
 can = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can.place(x=0, y=50)  
 can.create\_image(0, 0, image=png3, anchor=NW)  
 can2 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can2.place(x=0, y=200)  
 can2.create\_image(0, 0, image=png3, anchor=NW)  
 can3 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can3.place(x=0, y=350)  
 can3.create\_image(0, 0, image=png3, anchor=NW)  
 can4 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can4.place(x=0, y=500)  
 can4.create\_image(0, 0, image=png3, anchor=NW)  
 can5 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can5.place(x=615, y=50)  
 can5.create\_image(0, 0, image=png3, anchor=NW)  
 can6 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can6.place(x=615, y=200)  
 can6.create\_image(0, 0, image=png3, anchor=NW)  
 can7 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can7.place(x=615, y=350)  
 can7.create\_image(0, 0, image=png3, anchor=NW)  
 can8 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can8.place(x=615, y=500)  
 can8.create\_image(0, 0, image=png3, anchor=NW)  
 else:  
 Entry.unbind('<Return>')  
 Entry.config(state='disabled')  
  
def new\_game(): *#Функция начала новой игры* global n, x  
 answer3 = messagebox.askyesno(title='Новая игра', message='Вы уверены что хотите начать новую игру?')  
 if answer3:  
 n = 0  
 x = ''  
 sms2.config(text='Бык - цифра на своём месте.\n'  
 'Корова - цифра не на своём месте.')  
 Entry.config(state='normal')  
 Entry.bind('<Return>', play)  
 can = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can.place(x=0, y=50)  
 can.create\_image(0, 0, image=png3, anchor=NW)  
 can2 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can2.place(x=0, y=200)  
 can2.create\_image(0, 0, image=png3, anchor=NW)  
 can3 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can3.place(x=0, y=350)  
 can3.create\_image(0, 0, image=png3, anchor=NW)  
 can4 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can4.place(x=0, y=500)  
 can4.create\_image(0, 0, image=png3, anchor=NW)  
 can5 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can5.place(x=615, y=50)  
 can5.create\_image(0, 0, image=png3, anchor=NW)  
 can6 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can6.place(x=615, y=200)  
 can6.create\_image(0, 0, image=png3, anchor=NW)  
 can7 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can7.place(x=615, y=350)  
 can7.create\_image(0, 0, image=png3, anchor=NW)  
 can8 = Canvas(bg='gray22', width=200, height=200, highlightthickness=0)  
 can8.place(x=615, y=500)  
 can8.create\_image(0, 0, image=png3, anchor=NW)  
  
png = PhotoImage(file='bik.png')  
png2 = PhotoImage(file='korova.png')  
png3 = PhotoImage(file='gray22.png')  
png4 = PhotoImage(file='21.png')  
  
label1 = tkinter.Label(text='Быки и коровы', font='Courier 25', bg='gray22', fg='white') *#Вывод названия игры*label1.pack(anchor=N)  
  
buttonExample = tkinter.Button(root, *#Кнопка открытия правил* text="Открыть правила",  
 command=createNewWindow, bg='grey', fg='white', width=20, font='Helvetica 9')  
buttonExample.pack(anchor=S, pady=10, ipadx=10, ipady=10)  
  
buttonnewgame = tkinter.Button(root, text='Новая игра', command=new\_game, bg='grey', fg='blue', width=20, font='Helvetica 9') *#Кнопка новой игры*buttonnewgame.pack(anchor=S, pady=10, ipadx=10, ipady=10)  
  
buttonexit = tkinter.Button(root, *#Кнопка закрытия приложения* text='Закрыть приложение',  
 command=on\_closing, bg='grey', fg='red', width=20, font='Helvetica 9')  
buttonexit.pack(anchor=S, pady=10, ipadx=10, ipady=10)  
  
Entry = Entry(width=20, font=("Times", 15, "bold"), justify='center') *#Поле для ввода числа*Entry.pack(pady=5, ipadx=5, ipady=7)  
Entry.focus()  
Entry.bind( '<Return>', play) *#Привязка кнопки Enter, а так же основной игровой функции к полю Entry*sms2 = Message(width=600, padx=10, pady=10, justify="center", *#Поле для вывода результатов хода* text='Бык - цифра на своём месте.\n'  
 'Корова - цифра не на своём месте.\n', bg='gray22', fg='white')  
sms2.pack(anchor=N, pady=5)  
sms2.config(font=('times', 14))  
  
root.protocol("WM\_DELETE\_WINDOW", on\_closing)  
root.title('Быки и коровы')  
root.resizable(False, False)  
root.wm\_attributes('-topmost', 1)  
  
root.mainloop()