

**КУРСОВОЙ ПРОЕКТ**

**по дисциплине: МДК 01.01 Разработка программных модулей**

**Тема: Разработка программного модуля системы «Ипподром»**

**Специальность 09.02.07 «Информационные системы и программирование»**

|  |  |  |
| --- | --- | --- |
| **Выполнил студент(ка) группы 311ИС-22** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **М.Ш. Курбонмамадов** |
| **Руководитель** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Н.И. Кручинкина** |

**Москва 2024**



**УТВЕРЖДАЮ**

**Зам. директора КМПО**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.Ф. Гасанов**

**«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.**

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ**

**по дисциплине: МДК 01.01 Разработка программных модулей**

**Специальность 09.02.07 «Информационные системы и программирование»**

**Студент группы 311ИС-22 Курбонмамадов Мурид**

**ТЕМА: «Разработка программного модуля информационной**

**системы «Интернет магазин книг»**

Дата выдачи задания «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г.

Срок сдачи проекта «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г.

**Москва 2024**

**Перечень вопросов, подлежащих разработке:**

Введение

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Предпроектное обследование

1.2

2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА МОДУЛЯ

2.1 Эскизное проектирование

2.1.1

2.1.2

2.2 Проектирование базы данных

2.3 Техническое проектирование

2.3.1 Разработка архитектуры

2.3.2 Защита доступа к данным

3. ОТЛАДКА И ТЕСТИРОВАНИЕ МОДУЛЯ

3.1

3.2.

Заключение

**Исходные данные:**

Перечень разрабатываемых материалов для визуализации: электронная презентация, содержащая в том числе:

Блок-схемы алгоритмов программного модуля

Use-case диаграмма.

ER-диаграмма базы данных.

Архитектура программного модуля

Экранные формы интерфейса.

Задание выдал:

Руководитель курсового проекта Н.И. Кручинкина

Задание принял к исполнению В.Е. Третьякова

*Рассмотрено*

на заседании предметно-цикловой комиссии

информационных технологий и системного

администрирования

Протокол № \_\_\_*от «\_\_» \_\_\_\_\_\_\_\_\_* 2024 г.

Председатель ПЦК \_\_\_\_\_\_\_\_\_\_\_\_\_

**Оглавление**

[**ВВЕДЕНИЕ** 5](#_Toc185211047)

[**ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ** 7](#_Toc185211048)

[1.1. Информационное обеспечение задачи 7](#_Toc185211049)

[1.2. Обзор и анализ существующих программных решений 8](#_Toc185211050)

[1.3. Постановка задачи. Структура входной и выходной информации 8](#_Toc185211051)

[**ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА МОДУЛЯ** 9](#_Toc185211052)

[2.1 Построение модели системы 9](#_Toc185211053)

[2.2 Основные объекты и их описание 13](#_Toc185211054)

[2.3 План тестирования 15](#_Toc185211055)

[**3. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ МОДУЛЯ** 16](#_Toc185211056)

[3.1 Описание среды разработки 16](#_Toc185211057)

# **ВВЕДЕНИЕ**

Целью курсового проекта является разработка программного модуля системы «Ипподром», который обеспечит управление лошадьми, проведение гонок, учет сотрудников и управление финансами на ипподроме.

Задачи:

– разработать структуру базы данных для хранения информации о лошадях, сотрудниках, гонках и пользователях;

– реализовать функционал для добавления, редактирования и удаления лошадей, сотрудников и гонок;

– создать интерфейс для просмотра информации о лошадях, сотрудниках и истории гонок;

– реализовать функционал для проведения гонок и определения победителей;

– обеспечить безопасность данных и управление доступом для администраторов и пользователей.

Объектом исследования в данном проекте является деятельность ипподрома, включая управление лошадьми, проведение гонок и учет сотрудников. Предметом исследования является процесс автоматизации управления ипподромом с помощью программного модуля.

Структура курсовой работы обусловлена предметом, целью и задачами исследования. Курсовая работа состоит из введения, двух глав и заключения.

Введение раскрывает актуальность, определяет степень научной разработки темы, объект, предмет, цель, задачи и методы исследования, раскрывает теоретическую и практическую значимость работы.

В первой главе рассматривается теоретическая сторона проблемы.

Во второй главе исследования описывается процесс проектирования и разработки программного модуля.

# **ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

# Информационное обеспечение задачи

Ипподром — это специализированное спортивное сооружение, предназначенное для проведения конных соревнований, в первую очередь скачек, а также для тренировок лошадей и их владельцев. Предметная область включает в себя организацию, управление, проведение соревнований, взаимодействие с участниками (жокеями, владельцами лошадей, зрителями) и обеспечение безопасности.

Организация работы ипподрома требует координации множества процессов. Это включает управление заездами, учет лошадей, жокеев и их владельцев, финансовые операции, такие как продажа билетов, прием ставок и учет выигрышей. Все эти процессы взаимосвязаны и требуют четкого управления, чтобы обеспечить бесперебойную работу ипподрома.

# Обзор и анализ существующих программных решений

На рынке существует множество решений для управления ипподромами. Проведенный анализ выявляет следующие показывает следующее ключевые платформы:

1. Equibase – известная платформа, специализирующаяся на отслеживании результатов заездов и предлагающая обширные аналитические функции.
2. TrackMaster – сервис, предназначенный для анализа заездов и прогнозирования их результатов.
3. Horse Racing Maneger – комплексный инструмент для управления ипподромами, включая мобильные приложения для удобства пользователей.

Анализ демонстрирует, что большинство решений обладают схожими функциями, однако различаются в отношении уровня сервиса и ценовой стратегии. Разработка специализированного продукта способна повысить эффективность управления ипподромом.

# Постановка задачи. Структура входной и выходной информации

В результате анализа функционирования ипподрома были определены ключевые задачи:

– учет заездов: создание и поддержание базы данных о проведенных заездах;

– управление участниками: мониторинг участников и их лошадей;

– анализ ставок: составление отчетов по ставкам;

– введение базы лошадей: сбор и учет информации о лошадях;

– учет сотрудников: поддержание данных о работниках;

– данные о клиентах (имя, фамилия, контактная информация);

– данные о сотрудниках (имя, должность, зарплата);

– отчет о продаже билетов.

# **ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА МОДУЛЯ**

# Построение модели системы

Модуль для системы «Ипподром» должен обеспечивать функциональность управления заездами, учетом лошадей, их тренировками, а также взаимодействием с участниками (жокеями, тренерами, владельцами). Основные задачи системы включают управление данными о лошадях, заездах, результатами, а также предоставление отчетов для анализа эффективности.

Функциональные требования:

Управление лошадьми:

– хранение данных о каждой лошади, включая имя, возраст, тренера, владельца, описание и количество побед;

– возможность добавления, редактирования и удаления информации о лошадях;

– отображение списка лошадей с возможностью просмотра детальной информации о каждой лошади.

Управление заездами:

– создание, редактирование и удаление информации о заездах;

– хранение данных о времени начала и окончания заезда, участвующих лошадях, жокеях и их результатах;

– возможность просмотра истории заездов с деталями о победителях и их времени.

Управление участниками:

– хранение данных о жокеях, тренерах и владельцах лошадей;

– возможность добавления, редактирования и удаления информации о участниках;

– отслеживание статистики участников (например, количество побед жокея, успешность тренировок).

Анализ и отчеты:

– генерация отчетов по результатам заездов, включая победителей, их время и другие статистические данные;

– возможность анализа эффективности лошадей и жокеев на основе исторических данных;

– создание графиков и диаграмм для визуализации результатов.

Пользовательский интерфейс:

– простой и интуитивно понятный интерфейс для управления данными;

– возможность поиска и фильтрации данных по различным критериям (например, по имени лошади, дате заезда).

Нефункциональные требования:

Производительность:

– система должна обеспечивать быстрый доступ к данным, особенно при работе с большим объемом информации;

– время отклика системы на запросы пользователя не должно превышать 2 секунд.

Масштабируемость:

– архитектура системы должна позволять легко добавлять новые функции и расширять ее возможности в будущем;

– система должна поддерживать работу с базой данных, содержащей большое количество записей.

Документация:

– весь код должен быть хорошо задокументирован, включая комментарии к ключевым участкам и инструкции по использованию;

– документация должна быть понятной и доступной для разработчиков и пользователей.

Безопасность:

– система должна обеспечивать безопасность данных, включая защиту от несанкционированного доступа;

– должна быть реализована система аутентификации и авторизации для пользователей с различными уровнями доступа.

# Основные объекты и их описание

Таблица 1 – основные объекты и их описание

|  |  |  |
| --- | --- | --- |
| № | Наименование объекта | Краткое описание |
| 1 | Пользователи | Пользователи системы, которые могут быть администраторами или обычными пользователями. |
| 2 | Лошади | Объекты, представляющие лошадей с атрибутами, такими как имя, возраст, тренер, владелец, описание и количество побед. |
| 3 | Гонки | Записи о заездах, включая участвующие лошади, победителя и результаты. |
| 4 | Сотрудники | Записи о сотрудниках, включая их имя, роль, зарплату, контактную информацию. |
| 5 | Интерфейс пользователя | Графический интерфейс, который обеспечивает взаимодействие с системой. |

Таблица 2 – сущность “Пользователи”  Пользователи системы, которые могут быть  
администраторами или обычными пользователями.

|  |  |  |
| --- | --- | --- |
| Атрибуты | Тип | Описание |
| id | INTEGER | Уникальный идентификатор лошади. |
| username | TEXT | Имя пользователя. |
| password | TEXT | Пароль пользователя. |
| role | TEXT | Роль пользователя (администратор или пользователь). |
| balance | INTEGER | Баланс пользователя (используется для учета расходов). |

Таблица 3 – Сущность “Лошади” Объекты, представляющие лошадей с атрибутами,  
такими как имя, возраст, тренер, владелец,  
описание и количество побед.

|  |  |  |
| --- | --- | --- |
| Атрибуты | Тип | Описание |
| Id | INTEGER | Уникальный идентификатор лошади. |
| Name | TEXT | Имя лошади |
| Age | INTEGER | Возраст лошади |
| Trainer | TEXT | Имя тренера |
| Owner | TEXT | Имя владельца |
| Description | TEXT | Описание лошади |
| wins | INTEGER | Количество побед лошади в гонках |

Таблица 4 – Сущность “Гонки” Записи о заездах, включая участвующие лошади,  
победителя и результаты.

|  |  |  |
| --- | --- | --- |
| Атрибуты | Тип | Описание |
| Id | INTEGER | Уникальный идентификатор гонки. |
| Winner\_id | INTEGER | Идентификатор победителя гонки (ссылка на лошадь). |
| participants | TEXT | Список участвующих лошадей в гонке. |

Таблица 5 – Сущность “Сотрудник”

|  |  |  |
| --- | --- | --- |
| Атрибуты | Тип | Описание |
| id | INTEGER | Уникальный идентификатор сотрудника. |
| name | TEXT | Имя сотрудника. |
| role | TEXT | Роль сотрудника (например, тренер, жокей, администратор). |
| salary | INTEGER | Зарплата сотрудника. |
| phone | TEXT | Номер телефона сотрудника. |
| email | TEXT | Email сотрудника. |
| address | TEXT | Адрес сотрудника. |

Таблица 6 – Сущность “Интерфейс пользователя” графический интерфейс, который обеспечивает пользователя взаимодействие с системой.

|  |  |  |
| --- | --- | --- |
| Атрибуты | Тип | Описание |
| window | Tkinter.Tk | Основное окно приложения. |
| buttons | Tkinter.Button | Кнопки для взаимодействия с системой (например, добавить лошадь, начать гонку). |
| labels | Tkinter.Label | Метки для отображения информации (например, имя лошади, возраст, тренер). |
| listbox | Tkinter.Listbox | Список для отображения данных (например, список лошадей или сотрудников). |

Связи в программе:

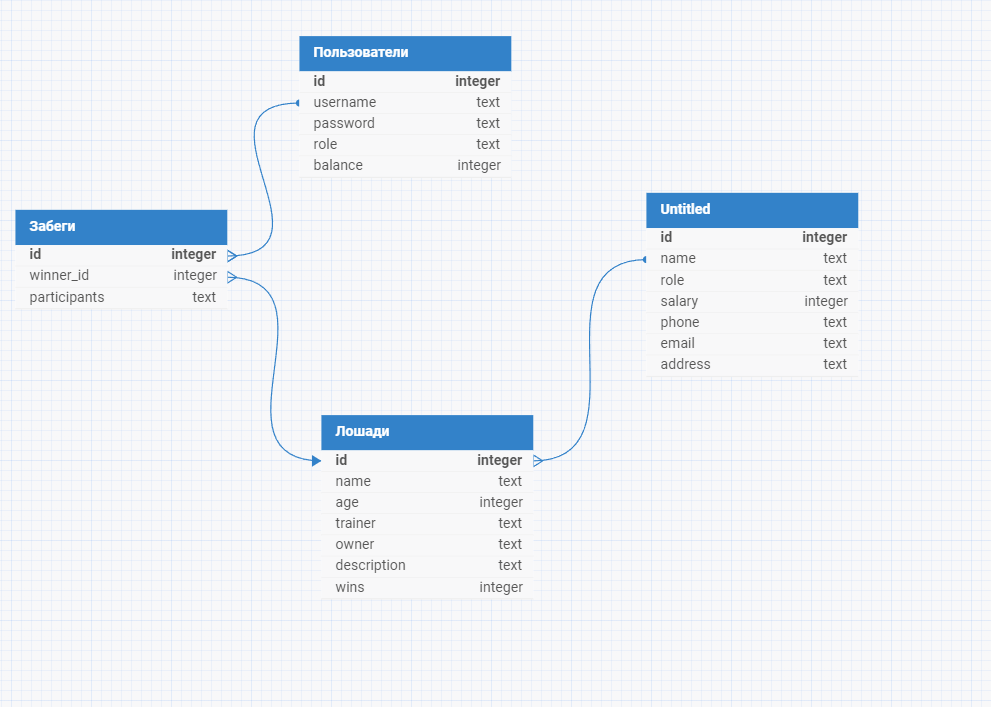


Рисунок 1 –Диаграмма бд ипподрома

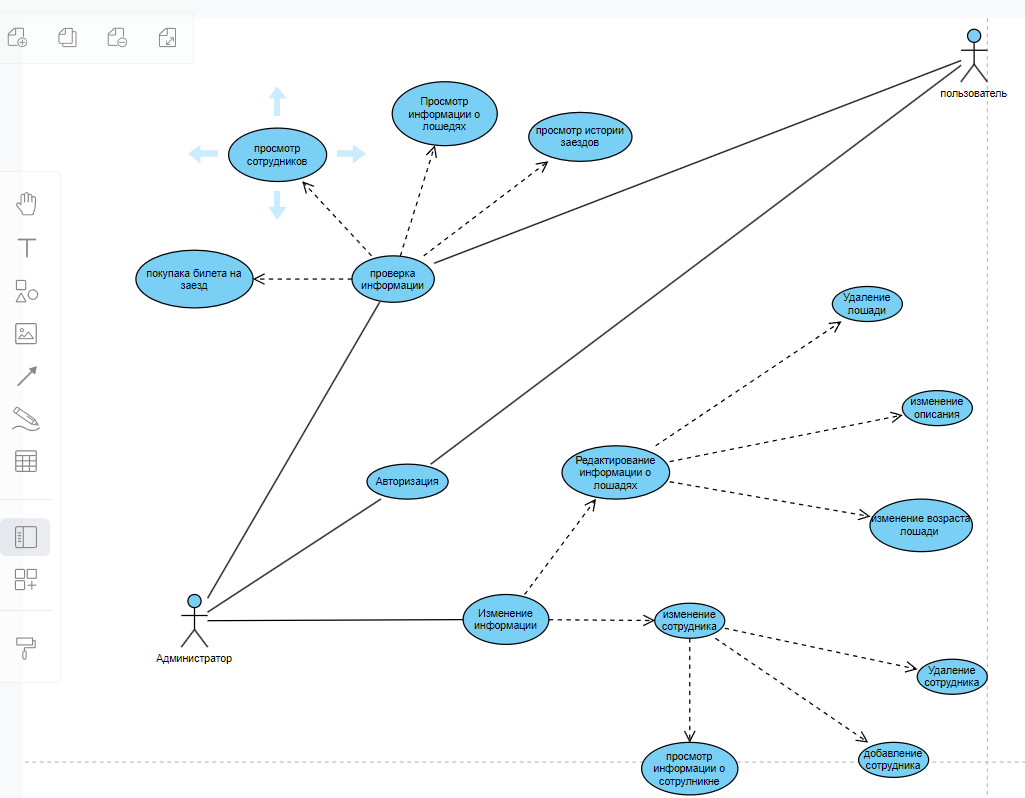


Рисунок 2 – ER-диаграмма бд ипподрома

# План тестирования

Регистрация и управление пользователями:

– проверить можно ли регистрироваться в данном приложении;

– проверить можно ли регистрировать администраторов.

Управление добавления и удаления лошадей:

– проверить, что система может добавлять и удалять лошадей из базы данных;

– проверить, что система позволяет редактировать данные о лошадях.

Проверка работоспособности:

– проверить, что система работает при большом количестве данных и пользователей.

Проверка обновления данных в реальном времени:

– проверить, что обновляет данные моментально.

Проверка интерфейса:

– проверить, что интерфейс нормально работают и нет никаких сбоев.

Тестирование проверка разделения между ролями:

– проверить, что пользователь не может использовать права администратора и редактировать что-то в базе данных.

Тестирование процессов добавления сотрудников:

– провести тестирование процесса добавления сотрудников.

**3. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ МОДУЛЯ**

3.1 Описание среды разработки

Для разработки программного модуля системы «Ипподром» были выбраны следующие инструменты и технологии:

1. **Среда разработки: PyCharm**  
   PyCharm — это интегрированная среда разработки (IDE), разработанная компанией JetBrains. Она предоставляет широкий набор инструментов для разработки на языке Python, включая интеллектуальное автодополнение кода, встроенный отладчик и поддержку систем тестирования. PyCharm также обладает удобным интерфейсом и множеством плагинов, что делает её популярным выбором среди разработчиков Python.
2. **Язык программирования: Python**  
   Python — это универсальный и мощный язык программирования, который сочетает в себе простоту синтаксиса и большую экосистему библиотек. Его использование позволяет быстро разрабатывать приложения, а его кроссплатформенность обеспечивает работу программ на различных операционных системах, таких как Windows, macOS и Linux.
3. **Библиотеки и модули:**
   * **Tkinter** — стандартная библиотека Python для создания графического интерфейса пользователя (GUI). Она предоставляет простой и интуитивно понятный API для создания окон, кнопок, полей ввода и других элементов интерфейса.
   * **Tkinter.ttk** — подмодуль Tkinter, который предлагает современные виджеты с улучшенным внешним видом. Он используется для создания более современных элементов интерфейса, таких как Combobox.
   * **Matplotlib** — библиотека для создания графиков и диаграмм. Она используется для визуализации данных, например, для построения круговых диаграмм, которые отображают распределение товаров на складе.
   * **Datetime** — встроенная библиотека Python для работы с датой и временем. Она используется для получения текущей даты и времени при создании транзакций.
   * **OS** — встроенная библиотека Python для работы с операционной системой. Она используется для управления файловой системой и создания директорий.
   * **Pillow** — библиотека для работы с изображениями. Она используется для загрузки и обработки изображений, которые отображаются в интерфейсе приложения.
4. **База данных: SQLite3**  
   SQLite3 — это встраиваемая реляционная система управления базами данных (СУБД), которая не требует отдельного сервера. Она поддерживает большинство стандартных SQL-команд и обеспечивает высокую производительность при работе с небольшими и средними объемами данных. SQLite3 идеально подходит для приложений, которые требуют простого и легкого в использовании хранилища данных.

### Преимущества выбранного инструментария:

* **PyCharm** обеспечивает высокую производительность разработки благодаря интеллектуальной поддержке кода, встроенному отладчику и поддержке тестирования.
* **Python** — простой и доступный язык программирования с богатой стандартной библиотекой и огромным количеством внешних библиотек, что позволяет решать широкий спектр задач.
* **Tkinter** — легко используемая библиотека для создания GUI, которая встроена в стандартную библиотеку Python, что делает её доступной без необходимости установки дополнительных пакетов.
* **SQLite3** — встраиваемая СУБД с низким потреблением ресурсов, что делает её идеальным выбором для приложений, требующих простого и эффективного хранения данных.

Таким образом, выбранный инструментарий обеспечивает необходимый уровень функциональности для разработки программного модуля, сохраняя простоту использования и обеспечивая эффективность в процессе работы.

3.2 Реализация пользовательского интерфейса приложения

Для упрощения работы пользователей с программным модулем системы «Ипподром» создан интерфейс, обеспечивающий интуитивно понятный доступ ко всем функциям системы. Он включает в себя различные элементы управления, такие как кнопки, текстовые поля, выпадающие списки и графические диаграммы, которые помогают пользователям эффективно выполнять необходимые операции.

Основные функции программного модуля были реализованы с использованием языка программирования Python и библиотеки Tkinter для создания графического интерфейса. Вот основные функции, реализованные в модуле:

– **Управление лошадьми**: добавление, редактирование и удаление информации о лошадях;

– **Управление заездами**: создание, редактирование и удаление информации о заездах;

– **Управление сотрудниками**: добавление, редактирование и удаление информации о сотрудниках;

– **Проведение гонок**: автоматическое определение победителя и сохранение результатов в базе данных;

– **Пользовательский интерфейс**: создание интуитивно понятного интерфейса для взаимодействия с системой.

Пользовательский интерфейс был разработан с использованием библиотеки Tkinter. Он включает в себя несколько экранов, которые позволяют пользователям выполнять различные действия, такие как просмотр информации о лошадях, управление сотрудниками и проведение гонок. Основные экраны интерфейса представлены на рисунках 1 и 2.

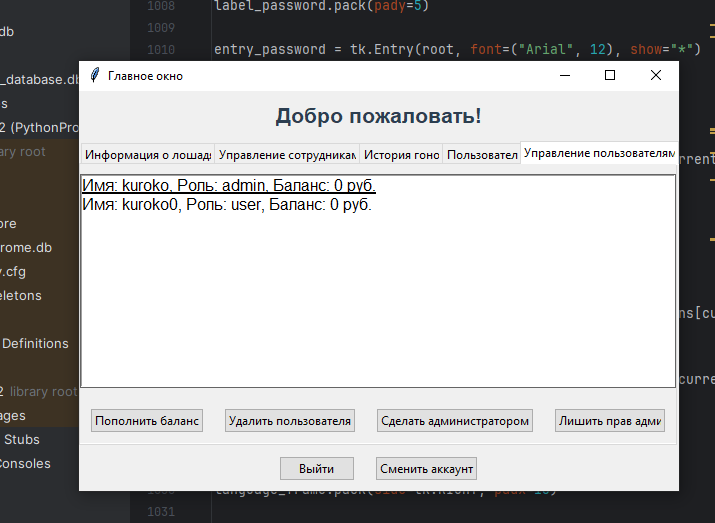


Рисунок 1 – главное окно приложения

Когда пользователь нажмет на просмотр лошадей его перекинет на новое окно и там буду указаны лошади, которые были добавлены в базу данных.

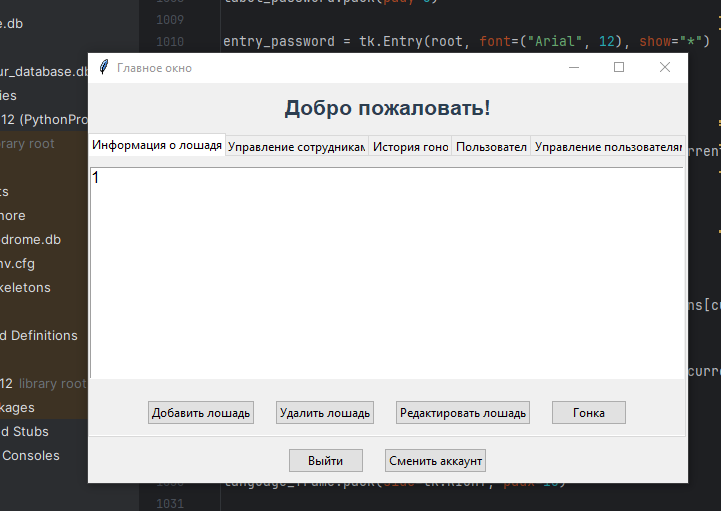


Рисунок 2 – окно просмотра лошадей

Для управления лошадьми были реализованы следующие функции:

– **Добавление лошади**: пользователь может добавить новую лошадь, указав её имя, возраст, тренера, владельца и описание. Информация сохраняется в базе данных.

– **Редактирование лошади**: пользователь может изменить информацию о существующей лошади, включая её имя, возраст, тренера, владельца и описание.

– **Удаление лошади**: пользователь может удалить лошадь из базы данных.

В просмотре лошадей можно будет смотреть данные о лошадях их имя, возраст, тренера, и т.д.

Для управления сотрудниками были реализованы следующие функции:

– **Добавление сотрудника**: пользователь может добавить нового сотрудника, указав его имя, роль, зарплату, контактную информацию и адрес. Информация сохраняется в базе данных.

– **Редактирование сотрудника**: пользователь может изменить информацию о существующем сотруднике, включая его имя, роль, зарплату, контактную информацию и адрес.

– **Удаление сотрудника**: пользователь может удалить сотрудника из базы данных.

Для проведения гонок были реализованы следующие функции:

– **Начало гонки**: система случайным образом выбирает участников гонки из списка лошадей и определяет победителя. Результаты гонки сохраняются в базе данных.

– **Визуализация гонки**: для наглядности процесса гонки была реализована анимация, которая показывает движение лошадей по полосе.



Пользовательский интерфейс был разработан с использованием библиотеки Tkinter. Он включает в себя следующие элементы:

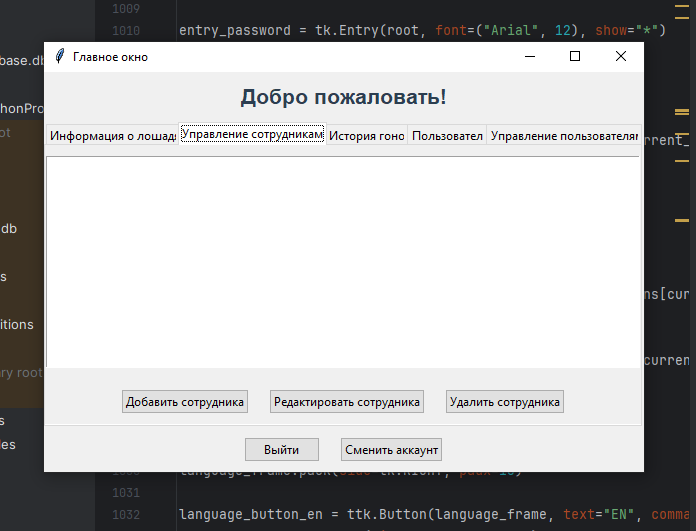
– **Главное окно**: отображает основные функции системы, такие как просмотр лошадей, управление сотрудниками и проведение гонок.

– **Окно просмотра лошадей**: отображает список лошадей и позволяет пользователю просматривать детальную информацию о каждой лошади.

– **Окно управления сотрудниками**: отображает список сотрудников и позволяет пользователю добавлять, редактировать и удалять сотрудников.

– **Окно проведения гонок**: отображает процесс гонки и результаты.

Так же можно смотреть и управлять сотрудниками при переходе на главный экран навестись и нажать на управление сотрудниками (это можно сделать только с помощью аккаунта администратора).В этом окне можно добавлять и удалять сотрудников распределять их по ролям и редактировать их данные.



**Заключение.**

В ходе выполнения курсового проекта была разработана информационная система для управления ипподромом, которая позволяет автоматизировать процессы управления лошадьми, проведением гонок, учетом сотрудников и финансовыми операциями. Основной целью проекта было создание программного модуля, который обеспечивает централизованное управление всеми аспектами работы ипподрома, что позволяет повысить эффективность работы, упростить взаимодействие между администраторами, пользователями и сотрудниками, а также обеспечить хранение и обработку данных в реальном времени.

В рамках проекта были решены следующие задачи:

1. **Анализ предметной области и существующих решений.**  
   В ходе анализа были изучены основные аспекты работы ипподрома, включая управление лошадьми, проведение гонок, учет сотрудников и финансовые операции. Также был проведен обзор существующих программных решений для управления ипподромами, что позволило выявить их преимущества и недостатки. В результате было принято решение о разработке специализированного программного модуля, который будет адаптирован под специфику работы ипподрома.
2. **Проектирование и разработка программного модуля.**  
   Была разработана архитектура программного модуля, включая структуру базы данных, функциональные требования и интерфейс пользователя. Основные объекты системы, такие как лошади, заезды, сотрудники и пользователи, были описаны с учетом их атрибутов и методов взаимодействия. Была создана база данных, которая позволяет хранить информацию о лошадях, заездах, сотрудниках и пользователях.
3. **Реализация функциональности модуля.**  
   В рамках проекта были реализованы основные функции программного модуля, включая управление лошадьми, проведение гонок, учет сотрудников и финансовые операции. Пользовательский интерфейс был разработан с учетом удобства и простоты использования, что позволяет администраторам и пользователям эффективно взаимодействовать с системой.
4. **Тестирование и отладка модуля.**  
   Для обеспечения корректной работы программного модуля был разработан план тестирования, который включал проверку основных функций системы, таких как регистрация пользователей, управление лошадьми, проведение гонок и анализ данных. В ходе тестирования были выявлены и устранены ошибки, что позволило обеспечить стабильную работу системы.
5. **Разработка пользовательского интерфейса.**  
   Для удобства пользователей был разработан интуитивно понятный интерфейс, который включает в себя различные элементы управления, такие как кнопки, текстовые поля, выпадающие списки и графические диаграммы. Интерфейс был разработан с учетом требований пользователей, что позволяет им эффективно выполнять необходимые операции.
6. **Обеспечение безопасности данных.**  
   В рамках проекта была реализована система аутентификации и авторизации, которая обеспечивает безопасность данных и управление доступом для администраторов и пользователей. Это позволяет предотвратить несанкционированный доступ к данным и обеспечить их защиту.

Разработанный программный модуль системы «Ипподром» позволяет автоматизировать процессы управления лошадьми, проведением гонок, учетом сотрудников и финансовыми операциями. Основные результаты выполнения проекта включают:

1. **Создание структуры базы данных.**  
   Была разработана база данных, которая позволяет хранить информацию о лошадях, заездах, сотрудниках и пользователях. Структура базы данных была спроектирована с учетом требований к хранению и обработке данных.
2. **Реализация функциональности модуля.**  
   Были реализованы основные функции программного модуля, включая управление лошадьми, проведение гонок, учет сотрудников и финансовые операции. Пользовательский интерфейс был разработан с учетом удобства и простоты использования.
3. **Тестирование и отладка модуля.**  
   В ходе тестирования были выявлены и устранены ошибки, что позволило обеспечить стабильную работу системы. Тестирование показало, что программный модуль работает корректно и соответствует требованиям пользователей.
4. **Разработка пользовательского интерфейса.**  
   Для удобства пользователей был разработан интуитивно понятный интерфейс, который включает в себя различные элементы управления, такие как кнопки, текстовые поля, выпадающие списки и графические диаграммы. Интерфейс был разработан с учетом требований пользователей, что позволяет им эффективно выполнять необходимые операции.
5. **Обеспечение безопасности данных.**  
   В рамках проекта была реализована система аутентификации и авторизации, которая обеспечивает безопасность данных и управление доступом для администраторов и пользователей. Это позволяет предотвратить несанкционированный доступ к данным и обеспечить их защиту.

Разработка программного модуля системы «Ипподром» является важным шагом в автоматизации процессов управления ипподромом. Созданная система позволяет централизованно управлять всеми аспектами работы ипподрома, что повышает эффективность работы, упрощает взаимодействие между администраторами, пользователями и сотрудниками, а также обеспечивает хранение и обработку данных в реальном времени.

В ходе выполнения проекта были решены все поставленные задачи, включая разработку структуры базы данных, реализацию функциональности модуля, тестирование и отладку системы, разработку пользовательского интерфейса и обеспечение безопасности данных. Результаты выполнения проекта показали, что разработанный программный модуль соответствует требованиям пользователей и может быть использован для управления ипподромом.

Таким образом, разработка программного модуля системы «Ипподром» имеет практическую значимость и поможет в совершенствовании управления ипподромом, что в свою очередь повысит его эффективность и конкурентоспособность.

**Список литературы:**

Ссылка на GitHub: [kurokoten/kursachrsabota](https://github.com/kurokoten/kursachrsabota)

1. **ГОСТ 7.32-2017**  
   "Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления".  
   Этот ГОСТ определяет структуру и правила оформления научно-исследовательских работ, включая курсовые проекты.
2. **ГОСТ 19.701-90 (ISO 5807:1985)**  
   "Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения".  
   Этот ГОСТ регламентирует правила создания блок-схем и других схем, используемых в программировании.
3. **ГОСТ 34.601-90**  
   "Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания".  
   Этот ГОСТ описывает стадии создания автоматизированных систем, что может быть полезно при проектировании программного модуля.
4. **ГОСТ 34.602-89**  
   "Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы".  
   Этот ГОСТ определяет структуру и содержание технического задания на разработку программного обеспечения.
5. **ГОСТ 24.211-82**  
   "Единая система конструкторской документации. Требования к текстовым документам".  
   Этот ГОСТ устанавливает требования к оформлению текстовых документов, включая пояснительные записки к курсовым проектам.

### **Литература по программированию и разработке:**

1. **Лутц, Марк. "Изучаем Python"**  
   Книга, которая поможет разобраться в основах языка Python, используемого в проекте.
2. **Бизли, Дэвид. "Python. К вершинам мастерства"**  
   Подробное руководство по продвинутым аспектам программирования на Python.
3. **Документация по Tkinter**  
   Официальная документация по библиотеке Tkinter, используемой для создания графического интерфейса:  
   <https://docs.python.org/3/library/tkinter.html>
4. **Документация по SQLite**  
   Официальная документация по работе с базой данных SQLite:  
   <https://www.sqlite.org/docs.html>
5. **Matplotlib Documentation**  
   Официальная документация по библиотеке Matplotlib для создания графиков и диаграмм:  
   <https://matplotlib.org/stable/contents.html>

### **Литература по проектированию информационных систем:**

1. **Гейн, А. Г. "Основы информатики и вычислительной техники"**  
   Классический учебник по основам информатики и программирования.
2. **Себеста, Р. В. "Основные концепции компьютерных наук"**  
   Книга, которая поможет разобраться в основных концепциях разработки программного обеспечения.
3. **Фаулер, Мартин. "UML. Основы"**  
   Книга, посвященная разработке программного обеспечения с использованием UML (Unified Modeling Language).
4. **Сэмюэл Грингард. "Python. Разработка на основе тестирования"**  
   Книга, которая поможет разобраться в процессе тестирования программного обеспечения.

### **Литература по управлению проектами:**

1. **PMBOK Guide (Project Management Body of Knowledge)**  
   Руководство по управлению проектами, которое может быть полезно для структурирования курсового проекта.
2. **ГОСТ Р ИСО 10006-2005**  
   "Системы менеджмента качества. Руководство по менеджменту качества проектов".  
   Этот ГОСТ поможет в управлении качеством проекта.

### **Дополнительные источники:**

1. **Документация по PyCharm**  
   Официальная документация по IDE PyCharm:  
   <https://www.jetbrains.com/help/pycharm/>
2. **Документация по Python**  
   Официальная документация по языку программирования Python:  
   <https://docs.python.org/3/>
3. **Статьи и материалы по разработке информационных систем**
   * Журнал "Программирование"
   * Журнал "Системный администратор"
   * Статьи на сайте <https://habr.com/>
4. **Книги по базам данных:**
   * Кристофер Дж. Date. "Введение в системы баз данных"
   * Томас Карел, Хью Дарвен. "SQL для простых смертных"