

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«Северный (Арктический) федеральный университет имени М.В. Ломоносова»

Высшая школа информационных технологий и автоматизированных систем
(наименование высшей школы / филиала / института / колледжа)

КУРСОВОЙ ПРОЕКТ

По дисциплине _____ Проекты _____

На тему _____ Разработка информационной системы «Зоопарк» _____

Выполнил обучающийся:

Попов Константин Алексеевич

(Ф.И.О.)

Направление подготовки / специальность:

09.03.02 «Информационные системы и

технологии»

(код и наименование)

Курс: 2

Группа: 351618

Руководитель: М.И. Антонов

(Ф.И.О. руководителя, должность / уч. степень / звание)

Признать, что проект выполнен и
защищен с отметкой

Руководитель

_____ (отметка прописью)

_____ (дата)

_____ (подпись руководителя)

М.И. Антонов

(инициалы, фамилия)

Архангельск 2018

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«Северный (Арктический) федеральный университет имени М.В. Ломоносова»

ЗАДАНИЕ

На курсовой проект по курсу

"Проекты"

ТЕМА: Разработка информационной системы «Зоопарк»

Студенту Попову Константину Алексеевичу 2 курса 351618 группы
Направление: «Информационные системы и технологии» 09.03.02

1 Общие требования

- 1.1 Необходимо разработать систему, тема которой согласована с преподавателем, ведущим дисциплину;
- 1.2 Система должна быть разработана в качестве настольного приложения (это не веб- или мобильное приложение). Для разработки системы вы можете выбрать между .NET или Java;
- 1.3 Вы можете использовать локальный сервер баз данных / экземпляр для разработки системы (обязательно к выполнению). Удаленный сервер баз данных может быть установлен после того, как система была разработана и готова к выпуску (по заданию не требуется);
- 1.4 Для выполнения задания должно быть разработаны требования по оформлению форм системы (цвета, шрифты, используемые логотипы).

2 Требуется к разработке

- 2.1 Разработать UML диаграмму для вашей системы. Сделать описание предполагаемых ролей (не менее 2 пользователей+ незарегистрированный пользователь) системы, описать набор доступных им функций;
- 2.2 Разработать ERD диаграмму и словарь данных;
- 2.3 Создайте базу данных (в базе не менее 5 таблиц), продемонстрируйте умение импортировать данные;
- 2.4 Создайте приложение согласно заданию. Приложение должно иметь:
 - 2.4.1 главную форму;
 - 2.4.2 форму регистрации;
 - 2.4.3 авторизации;
 - 2.4.4 формы для выполнения функционала каждого пользователя;
 - 2.4.5 интерактивная карта;
 - 2.4.6 предусмотреть возможность выгрузки информации;
 - 2.4.7 форма, позволяющая выполнять расчеты.
- 2.5 Описать тестирование системы;
- 2.6 Создать презентацию для демонстрации выполненной работы;
- 2.7 В приложении привести листинг.

3 Объем курсового проекта

Объем пояснительной записки - 30 листов формата А4 без учета листинга.

Дата выдачи работы: февраль 2018 г. Срок сдачи работы: июнь 2018 г.

Дата защиты: _____ Оценка: _____

Работу принял: _____

Руководитель работы:

Заведующий кафедрой:

ЛИСТ ДЛЯ ЗАМЕЧАНИЙ

СОДЕРЖАНИЕ

Определения, обозначения, сокращения.....	5
Введение	6
1 Анализ предметной области.....	7
2 Проектирование Приложения	9
3 Разработка приложения	13
4 Тестирование приложения.....	24
Заключение.....	29
Список использованных источников.....	30
Приложение А (обязательное) Листинг кода главной формы Form1.cs.....	31
Приложение Б (обязательное) Листинг кода формы регистрации пользователей RegistrationForm.cs.....	33
Приложение В (обязательное) Листинг кода формы меню Menu.cs	37
Приложение Г (обязательное) Листинг кода формы животных animal.cs	39
Приложение Д (обязательное) Листинг кода формы добавления нового животного animal_create.cs	41
Приложение Е (обязательное) Листинг кода формы экскурсий excursion.cs	43
Приложение Ж (обязательное) Листинг кода формы карты map.cs	46
Приложение З (обязательное) Листинг кода формы билетов Tickets.cs.....	49
Приложение И (обязательное) Листинг кода формы выгрузки данных в Excel ExportForm.cs	52

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

ER – entity-relationship model, модель «сущность — связь».

API – application programming interface, программный интерфейс приложения.

UML – Unified Modeling Language, унифицированный язык моделирования.

Use case – Сценарий использования, вариант использования, прецедент использования.

ВВЕДЕНИЕ

Все чаще и чаще различные компании обращаются за помощью к специалистам в области информационных технологий, с целью разработки прикладного программного обеспечения. Так как подобный продукт значительно повышает эффективность работы в целом.

Целью работы является разработка информационной системы «Зоопарк». Из цели вытекают следующие задачи:

- спроектировать архитектуру базы данных, создать ER-диаграмму и словарь данных;
- создать диаграмму вариантов использования;
- на основе созданной ER-диаграммы создать базу данных;
- создать приложение;
- протестировать созданное приложение.

Исходными данными для разрабатываемой темы являются требуемые отношения в базе данных, а также описание предметной области.

Поставленные задачи будем решать, разрабатывая систему с помощью интерфейса программирования приложений Windows Form. Windows Forms — интерфейс программирования приложений (API), отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Данный интерфейс упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде [1]. В качестве системы управления базы данных выберем MySQL. MySQL – это одна из самых популярных и самых распространенных СУБД (система управления базами данных) в интернете. Она не предназначена для работы с большими объемами информации, но ее применение идеально для интернет-сайтов, как небольших, так и достаточно крупных [2].

Ожидаемым результатом является работающее приложение, которое связано с созданной базой данных.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Выбранной предметной областью является Зоопарк, в котором мы можем выделить следующие отношения:

- животное;
- клиент;
- работник;
- экскурсия;
- корпус;
- билеты;
- пользователь;
- тип пользователя.

В рамках разрабатываемого приложения мы можем выделить три типа пользователей:

- работник;
- посетитель;
- незарегистрированный пользователь.

Посетитель может использовать все функции, которые не изменяют уже имеющиеся данные в базе данных, например, покупать билеты, просматривать карту зоопарка, просматривать животных и так далее.

Работник – это пользователь, которому разрешается корректировать данные в базе данных. Также имя этого пользователя будет известно посетителю, если данный работник является экскурсоводом. Помимо этого, ему доступны все функции, доступные посетителю, кроме записи на экскурсию.

Система не подразумевает доступ к полному функционалу для незарегистрированного пользователя, поэтому ему доступна лишь одна функция – регистрация.

Основными процессами, которые происходят в данной предметной области, являются покупка билета и запись на экскурсию. Так же система должна поддерживать функционал изменения данных о животных, который доступен только работнику. Функция запись на экскурсию доступна только посетителю, а регистрация – незарегистрированному пользователю.

Посетитель и работник являются зарегистрированными пользователями и им доступны следующие функции:

- авторизация;
- просмотр животных;
- просмотр карты зоопарка;
- покупка билетов.

2 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

На основе собранной информации о предметной области мы можем составить ER-диаграмму, которая представлена на рисунке 1.

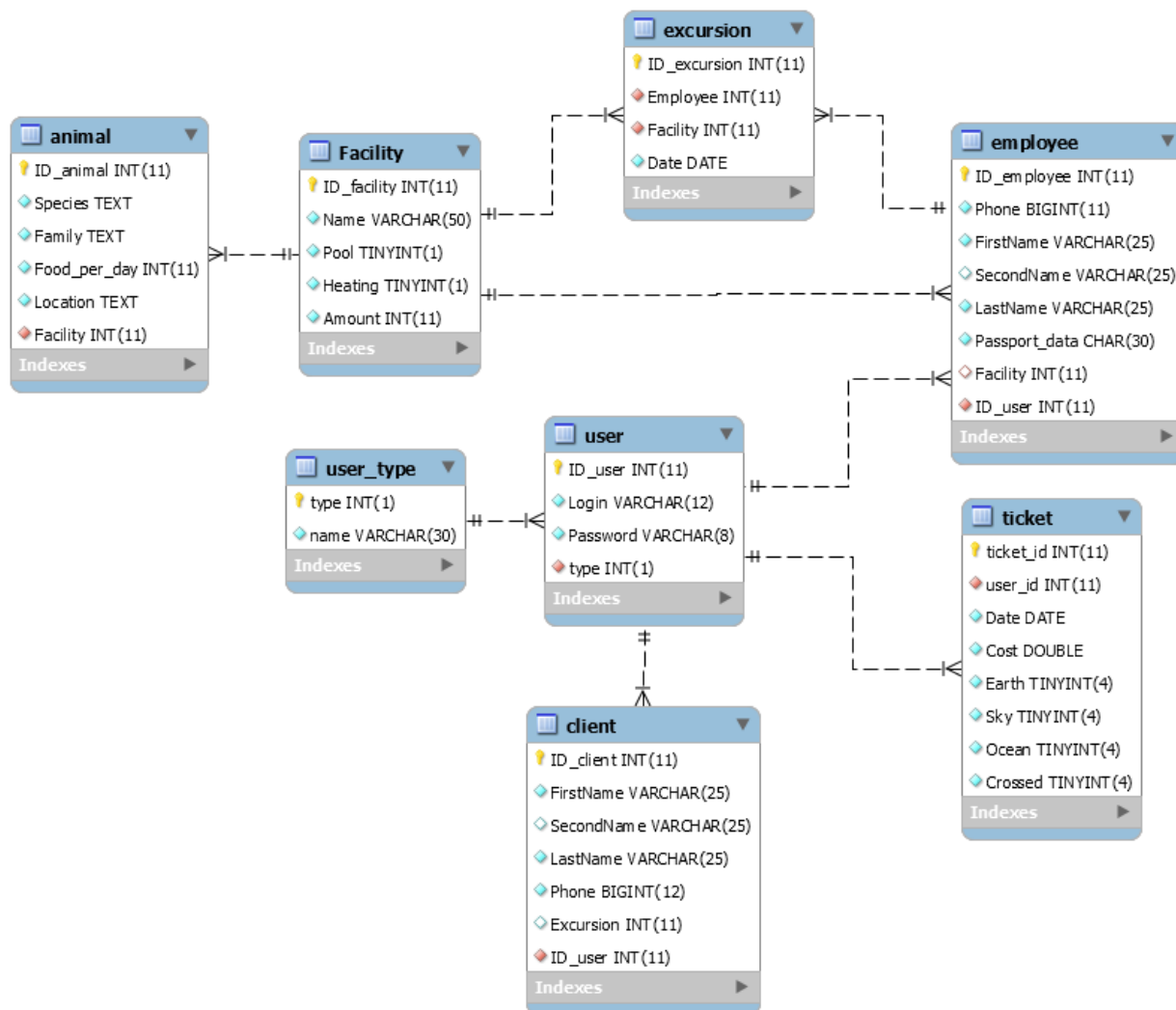


Рисунок 1 – ER-диаграмма

Также был составлен словарь данных, представленный в таблицах 1 – 8.

Таблица 1 – Отношение animal

Атрибут, наименование	Тип данных	Комментарии
ID_animal	INT(11)	Первичный ключ, Not Null
Species	TEXT	Not Null
Family	TEXT	Not Null
Food_per_day	INT(11)	Not Null
Location	TEXT	Not Null
Facility	INT(11)	Внешний ключ, Not Null

Таблица 2 – Отношение client

Атрибут, наименование	Тип данных	Комментарии
ID_client	INT(11)	Первичный ключ, Not Null
FirstName	VARCHAR(25)	Not Null
SecondName	VARCHAR(25)	
LastName	VARCHAR(25)	Not Null
Phone	BIGINT(12)	Not Null
Excursion	INT(11)	Внешний ключ
ID_user	INT(11)	Внешний ключ, Not Null

Таблица 3 – Отношение employee

Атрибут, наименование	Тип данных	Комментарии
ID_employee	INT(11)	Первичный ключ, Not Null
FirstName	VARCHAR(25)	Not Null
SecondName	VARCHAR(25)	
LastName	VARCHAR(25)	Not Null
Phone	BIGINT(12)	Not Null
Passport_data	CHAR(30)	Not null
Facility	INT(11)	Внешний ключ
ID_user	INT(11)	Внешний ключ, Not Null

Таблица 4 – Отношение excursion

Атрибут, наименование	Тип данных	Комментарии
ID_excursion	INT(11)	Первичный ключ, Not Null
Employee	INT(11)	Внешний ключ, Not Null
Date	DATE	Not Null
Facility	INT(11)	Внешний ключ

Таблица 5 – Отношение Facility

Атрибут, наименование	Тип данных	Комментарии
ID_facility	INT(11)	Первичный ключ, Not Null
Name	VARCHAR(50)	Not Null
Pool	TINYINT(1)	Not Null
Heating	TINYINT(1)	Not Null
Amount	INT(11)	Not Null, Default = 0

Таблица 6 – Отношение ticket

Атрибут, наименование	Тип данных	Комментарии
ticket_id	INT(11)	Первичный ключ, Not Null
user_id	INT(11)	Внешний ключ, Not Null
Date	DATE	Not Null
Cost	DOUBLE	Not Null
Earth	TINYINT(1)	Not Null
Sky	TINYINT(1)	Not Null
Ocean	TINYINT(1)	Not Null
Crosse	TINYINT(1)	Not Null

Таблица 7 – Отношение user

Атрибут, наименование	Тип данных	Комментарии
ID_user	INT(11)	Первичный ключ, Not Null
Login	VARCHAR(12)	Not Null, Unique
Password	VARCHAR(8)	Not Null
type	INT(1)	Внешний ключ, Not Null

Таблица 8 – Отношение user_type

Атрибут, наименование	Тип данных	Комментарии
type	INT(1)	Первичный ключ, Not Null
name	VARCHAR(30)	Not Null

Также, благодаря анализу предметной области, мы можем составить диаграмму варианта использования. Use-case диаграмма представлена на рисунке 2.

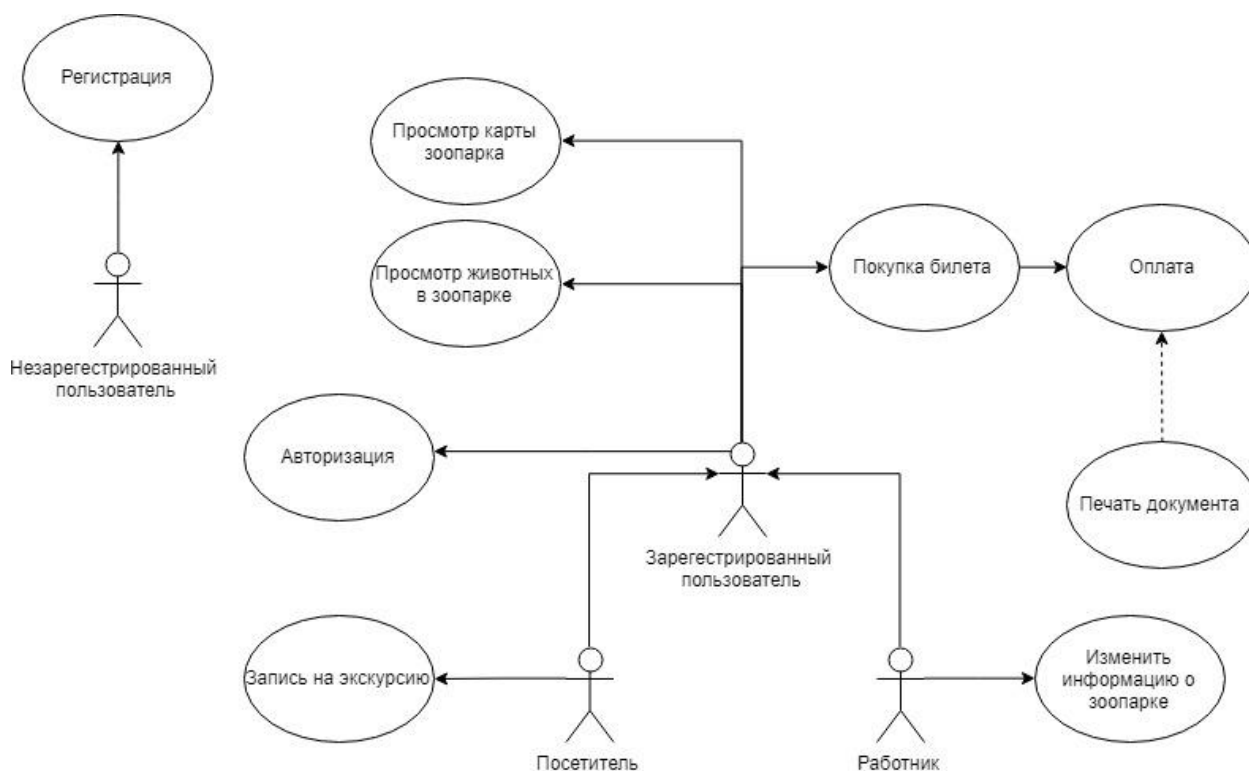


Рисунок 2 – Use-case диаграмма

Каждая форма приложения должна соответствовать основному дизайну. А именно:

- заголовок формы оформляется шрифтом Century Gothic; 21,75пт;
- остальной текст на форме шрифтомTahoma; 11,25пт;
- заголовок, основное тело и элементы нижней части формы отделяются цветными полосами;
- меню может быть произвольного оформления, но должно содержать интерактивные элементы;
- на каждой форме должны быть кнопка, позволяющая вернуться на предыдущий экран;
- все места, где требуется ввести числовое значение, представлять в виде ползунков или счетчиков, чтобы избежать ввода некорректного значения;

- кнопки имеют стандартное для Windows Form оформление с учетом выбранного шрифта.

3 РАЗРАБОТКА ПРИЛОЖЕНИЯ

Разработка приложения началась с создания базы данных с помощью MySQL Workbench – инструмента для визуального проектирования баз данных, интегрирующего проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL [3]. Работа с данной программой представлена на рисунке 3.

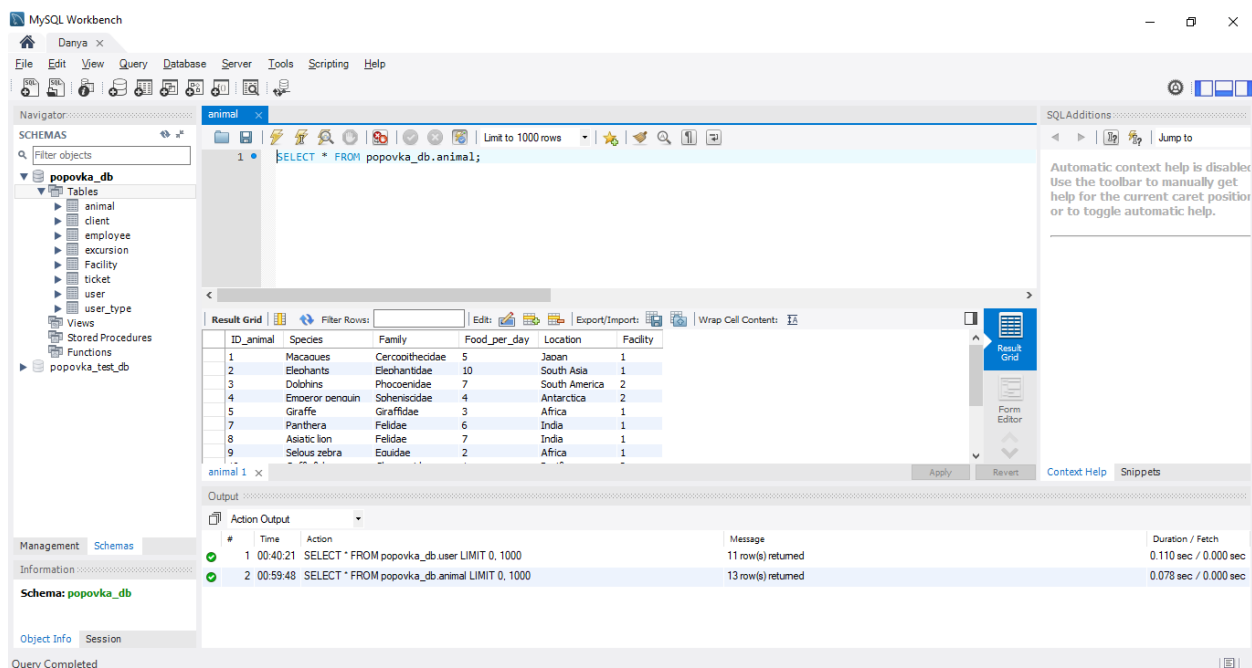


Рисунок 3 – MySQL Workbench

После создания базы данных, началось создание самого приложения с помощью Visual Studio Community - полнофункциональной, расширяемой и бесплатной интегрированной среды разработки для создания современных приложений Android, iOS и Windows, а также веб-приложений и облачных служб [4].

Для написания был выбран язык программирования C# - объектно-ориентированный язык программирования, разработанный в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework [5].

Первой была разработана форма авторизации, которая встречает пользователя при запуске приложения, на которой можно пройти авторизацию или перейти на форму с регистрацией. Данная форма представлена на рисунке 4.

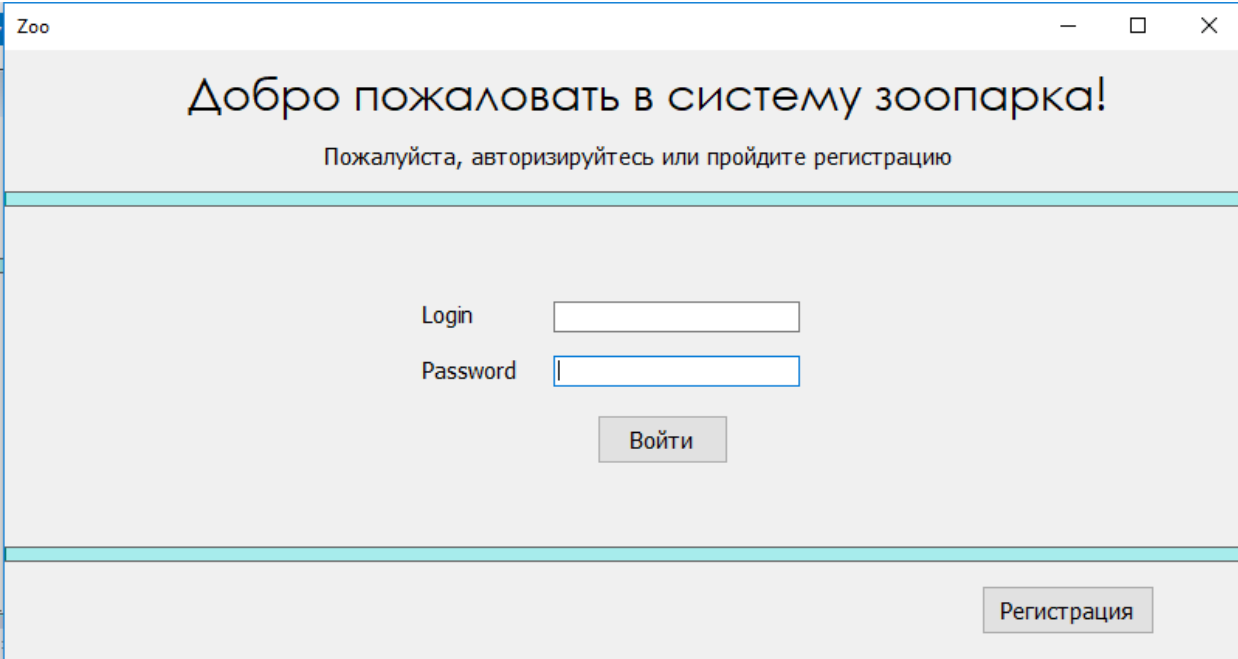
The image shows a window titled "Zoo" with standard Windows window controls (minimize, maximize, close). The main content area has a light gray background. At the top, there is a header section with the text "Добро пожаловать в систему зоопарка!" in a large, bold, black font, and below it, in a smaller font, "Пожалуйста, авторизируйтесь или пройдите регистрацию". Below this header is a large white rectangular area containing the login fields. There are two labels, "Login" and "Password", each followed by a white text input field. Below the "Password" field is a gray button with the text "Войти". At the bottom of the window, there is a gray footer area containing a button labeled "Регистрация".

Рисунок 4 – Форма авторизации

На форме с регистрацией пользователь может выбрать под каким типом он хочет пройти регистрацию и ввести данные, которые требуются системе. Регистрация для работника представлена на рисунке 5, а для посетителя на рисунке 6.

При нажатии на кнопку «Зарегистрироваться», данные пользователя заносятся в базу данных.

Registration

Регистрация нового пользователя

Пожалуйста, заполните поля ниже

Логин

Пароль

Тип пользователя

Телефон

Фамилия Имя Отчество

Паспортные данные

Здание работы

Рисунок 5 – Регистрация работника

Registration

Регистрация нового пользователя

Пожалуйста, заполните поля ниже

Логин

Пароль

Тип пользователя

Телефон

Фамилия Имя Отчество

Рисунок 6 – Регистрация посетителя

После авторизации, появляется меню, из которого можно перейти во все разделы системы, а также выйти из системы. Форма меню представлена на рисунке 7.

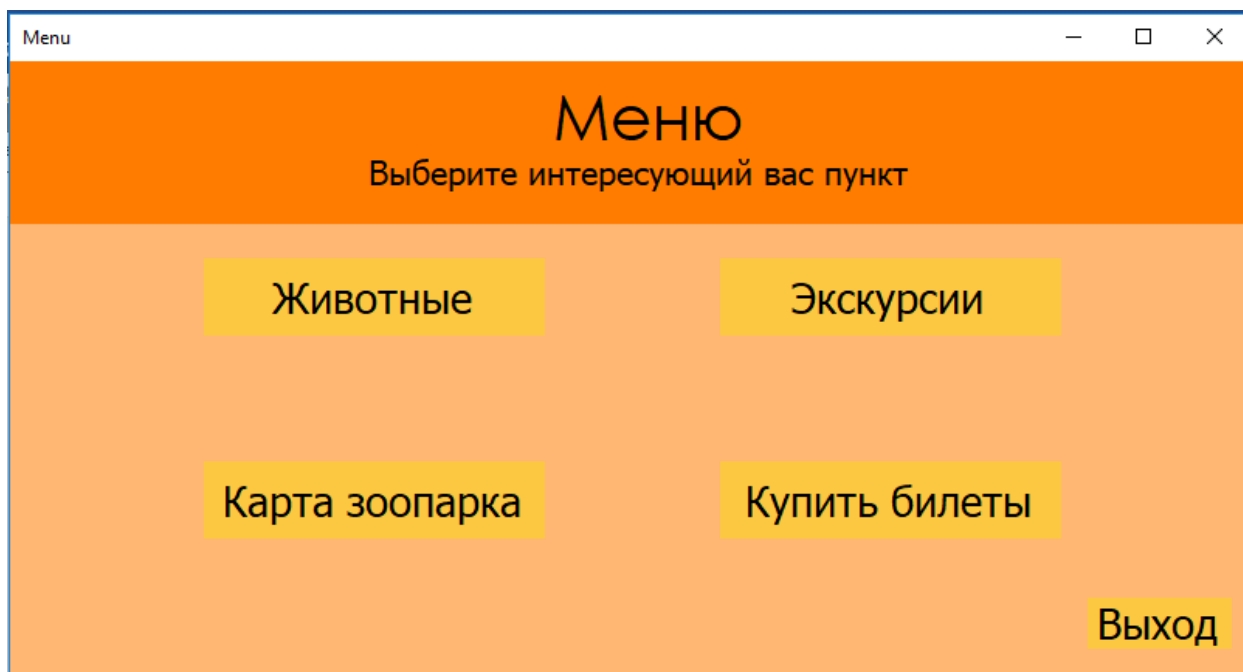


Рисунок 7 – Форма меню

При нажатии на кнопку с надписью «Животные» открывается окно, посвященное разделу работы с животным. На данной форме можно просмотреть всех животных. С помощью выпадающего списка пользователь может отсортировать животных по нахождению в различных корпусах, а также осуществить поиск по виду. Форма представляемая посетителю, представлена на рисунке 8. Для работника доступна функция добавления животного, его форма показана на рисунке 9.

animal

Животные

Ниже представлен список животных, находящихся в зоопарке

Корпус

Поиск по имени

Species	Family	Location	Facility
Macaques	Cercopithecidae	Japan	Earth World
Elephants	Elephantidae	South Asia	Earth World
Giraffe	Giraffidae	Africa	Earth World
Panthera	Felidae	India	Earth World
Asiatic lion	Felidae	India	Earth World
Selous zebra	Equidae	Africa	Earth World
Zmey	Piton	Jungle	Earth World
Dolphins	Phocoenidae	South America	Ocean World
Emperor penguin	Spheniscidae	Antarctica	Ocean World
Coffinfish	Chaunacidae	Pacific	Ocean World
Cod	Gadidae	North Atlantic	Ocean World
KRAB	Kraby	MORE	Ocean World

Рисунок 8 – Форма посетителя, посвященная животным

animal

Животные

Ниже представлен список животных, находящихся в зоопарке

Корпус

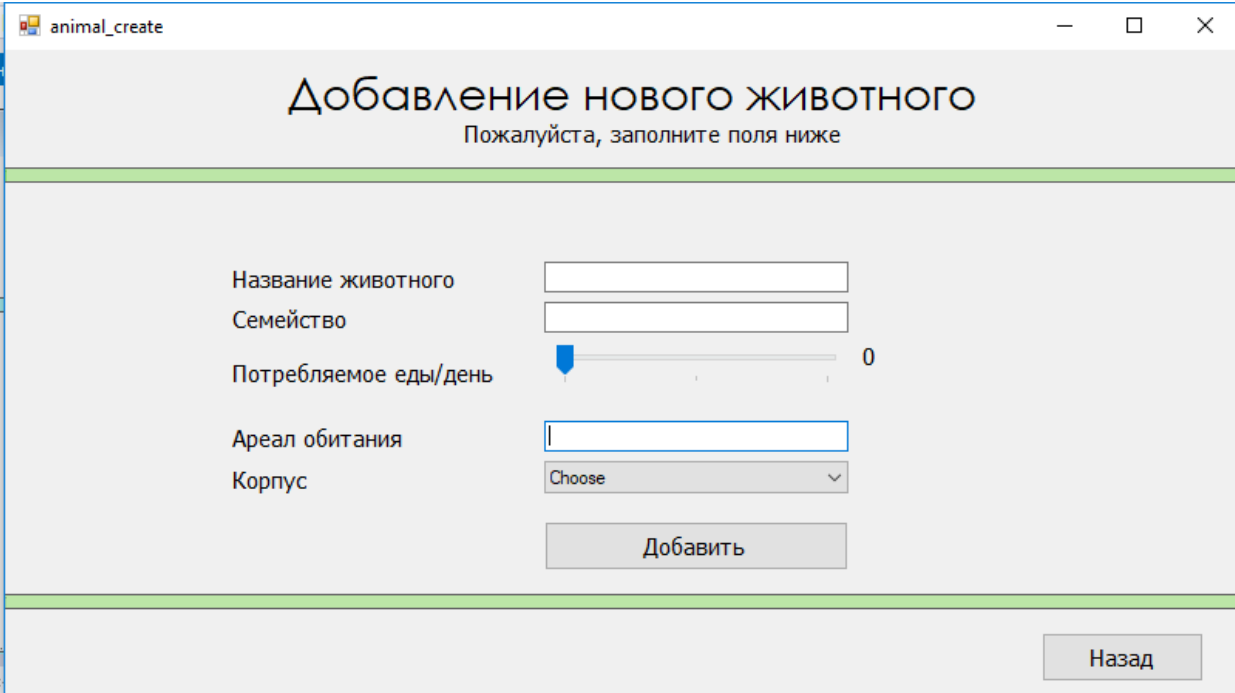
Поиск по имени

Species	Family	Location	Facility
Macaques	Cercopithecidae	Japan	Earth World
Elephants	Elephantidae	South Asia	Earth World
Giraffe	Giraffidae	Africa	Earth World
Panthera	Felidae	India	Earth World
Asiatic lion	Felidae	India	Earth World
Selous zebra	Equidae	Africa	Earth World
Zmey	Piton	Jungle	Earth World
Dolphins	Phocoenidae	South America	Ocean World
Emperor penguin	Spheniscidae	Antarctica	Ocean World
Coffinfish	Chaunacidae	Pacific	Ocean World
Cod	Gadidae	North Atlantic	Ocean World
KRAB	Kraby	MORE	Ocean World

Рисунок 9 – Форма работника, посвященная животным

При нажатии на кнопку «Добавить новое животное» появляется форма добавления животного. На форме, представленной на рисунке 10, работник вводит

данные добавляемого животного. При нажатии на кнопку «Добавить», данные животного заносятся в базу данных.



The screenshot shows a web browser window with the title 'animal_create'. The main heading is 'Добавление нового животного' (Adding a new animal), followed by the instruction 'Пожалуйста, заполните поля ниже' (Please fill in the fields below). The form contains the following elements:

- 'Название животного' (Animal name) with a text input field.
- 'Семейство' (Family) with a text input field.
- 'Потребляемое еды/день' (Food consumed per day) with a horizontal slider control, currently set to 0.
- 'Ареал обитания' (Habitat) with a text input field.
- 'Корпус' (Body) with a dropdown menu showing 'Choose'.
- A 'Добавить' (Add) button.
- A 'Назад' (Back) button in the bottom right corner.

Рисунок 10 -Форма добавления животного

При переходе из меню в раздел «Экскурсии» появляется форма, предоставляющая информацию о доступных экскурсиях. Пользователю необходимо выбрать корпус, доступного экскурсовода, а затем дату. Посетителю доступна функция записи на экскурсию, работнику она не доступна. При нажатии посетителем на кнопку «Записаться» данные базы данных обновляются. Форма для работника представлена на рисунке 11, а для посетителя на рисунке 12.

excursion

Экскурсии

Все доступные экскурсии представлены ниже

Корпус

Экскурсовод

Дата

Назад

Рисунок 11 – Форма экскурсии для работника

excursion

Экскурсии

Все доступные экскурсии представлены ниже

Корпус

Экскурсовод

Дата

Записаться

Назад

Рисунок 12 – Форма экскурсии для посетителя

При переходе из меню в раздел «Купить билеты» открывается соответствующее окно. Пользователь может выбрать, в какие корпуса он хочет пойти, а также сколько билетов и какого типа он хочет купить. В строке «Общая

сумма» динамически обновляется значение, в зависимости от выбранных параметров билета. При нажатии на кнопку «Купить» добавляется новый кортеж в отношение «ticket» в базе данных.

Работнику доступна функция выгрузки данных по билетам в Excel. При нажатии на кнопку «Выгрузка данных», открывается соответствующая форма, на которой работник может отредактировать данные и выгрузить их в Excel.

Пример формы работника представлен на рисунке 13, 14 и 15, а посетителя на рисунке 16.

The screenshot shows a web application window titled "Tickets". The main heading is "Билеты" (Tickets) with a subtitle "Заполните данные, необходимые для покупки билета, ниже" (Fill in the data necessary for ticket purchase below). The form contains several input fields and checkboxes. Under "Корпус:" (Cabin/Room), there are checkboxes for "Earth World", "Ocean World" (checked), "Sky World", and "Cross World" (checked). The "Дата:" (Date) field shows "24 июня 2018 г.". Below this, "Количество билетов для взрослых:" (Number of tickets for adults) is set to 2, and "для детей:" (for children) is set to 1. A map of a park area is displayed with two grey markers labeled "Choosed". At the bottom left, "Общая цена: 875" (Total price: 875) is shown next to a "Купить" (Buy) button. At the very bottom, there are two buttons: "Выгрузка данных" (Export data) and "Назад" (Back).

Рисунок 13 – Форма покупки билетов для работника

Билеты

Заполните данные, необходимые для покупки билета, ниже

Корпус: ☐ Earth World ☐ Ocean World ☐ Sky World ☐ Cross World Дата: 21 июня 2018 г.

Количество билетов
для взрослых: 0
для детей: 0

Общая цена: 0

Рисунок 16 – Форма покупки билетов для посетителя

При переходе в раздел «Карта» из меню открывается окно, на котором расположена карта зоопарка. При наведении курсора на здание выплывает информация о нём. Пример представлен на рисунке 17.



Рисунок 17 – Форма карты

4 ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ

Тестирование системы происходило по стратегии черного ящика. Тестирование чёрного ящика или поведенческое тестирование — стратегия (метод) тестирования функционального поведения объекта (программы, системы) с точки зрения внешнего мира, при котором не используется знание о внутреннем устройстве тестируемого объекта. Под стратегией понимаются систематические методы отбора и создания тестов для тестового набора. Стратегия поведенческого теста исходит из технических требований и их спецификаций [6].

Тестирование авторизации.

При попытке авторизоваться без введенных данных, система выведет предупреждение, представленное на рисунке 18.

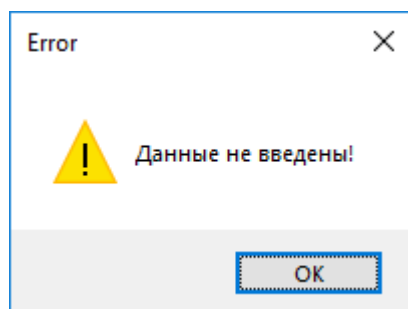


Рисунок 18 – Предупреждение системы

При вводе несуществующих данных, система выводит ошибку, представленную на рисунке 19.

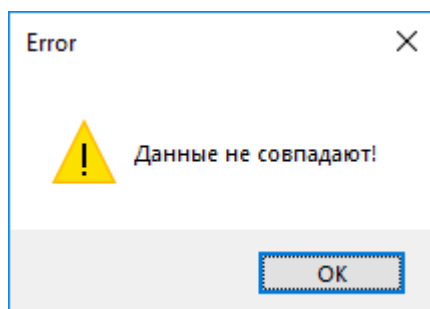


Рисунок 19 – Введены несуществующие данные

Тестирование регистрации.

Если оставить все поля пустыми или ввести некорректные данные система сообщит об ошибке (рисунок 20).

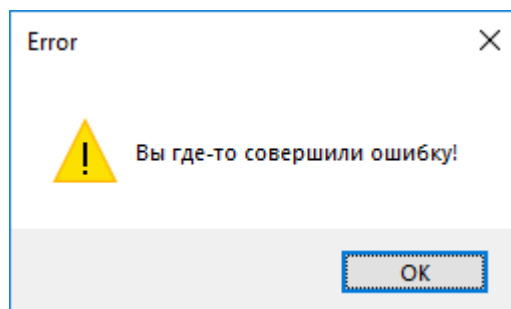


Рисунок 20 – Ошибка при пустых полях

Если ввести некорректные логин и пароль, система об этом сообщит (рисунок 21).

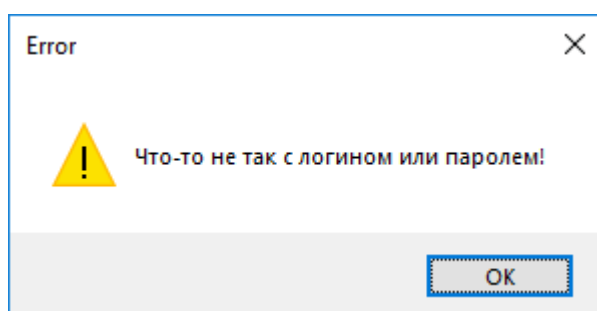


Рисунок 21 – Неверные логин или пароль

При вводе существующих логин и пароля выводится сообщение, представленное на рисунке 22.

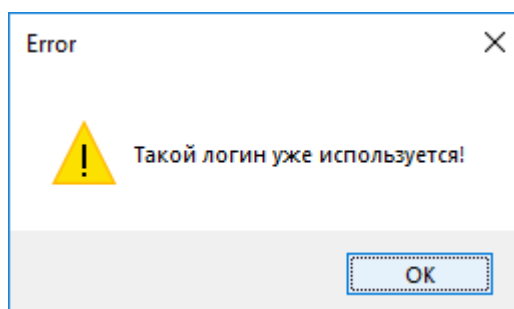


Рисунок 22 – Ввод существующего логина

Тестирование покупки билетов

Если оставить все поля пустыми и попробовать совершить покупку система не даст это сделать (Рисунок 23).

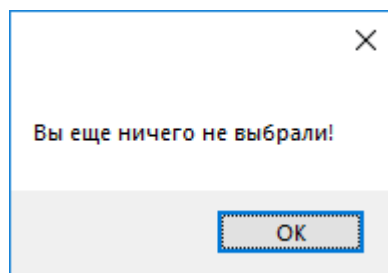


Рисунок 23 – Ошибка при покупке билетов

Если установить неверную дату, например, недельной давности, то система об этом скажет (рисунок 24).

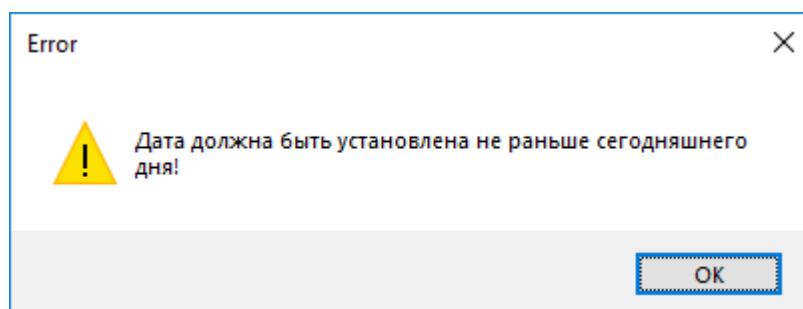


Рисунок 24 – Неверная дата

Тестирование записи на экскурсию.

Так как элементов управления на этой форме достаточно мало, то единственную ошибку, которую тут можно совершить: не до конца выбрать все параметры экскурсии, в этом случае система оповестит пользователя. Пример на рисунке 25.

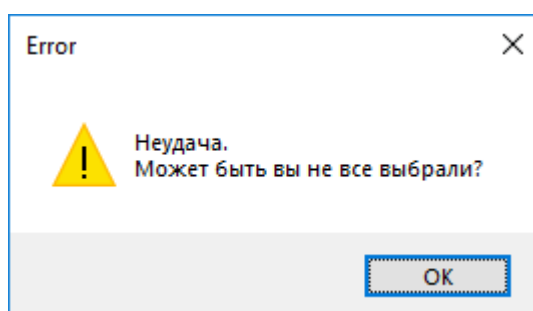


Рисунок 25 – Ошибка на форме экскурсий

Тестирования раздела «животные».

При добавление нового животного, если пользователь ввел не корректные данные или оставил поля пустыми, система выведет ошибку, представленную на рисунке 26.

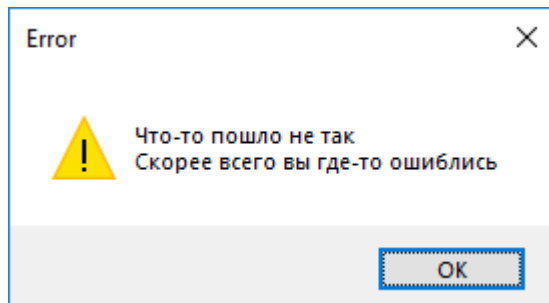


Рисунок 26 – Ошибка при добавлении нового животного

Если при в строке поиска животных ввести вид, которого в зоопарке нет, то система об этом сообщит (рисунок 27).

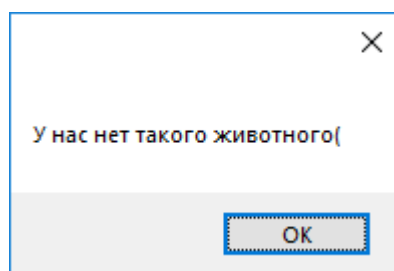


Рисунок 27 – Неудачный поиск

Тестирование карты.

Тестирование карты показало, что при определенном положении мыши, всплывающая информация может уйти за границу приложения. Пример такой ошибки представлен на рисунке 28.



Рисунок 28 – Ошибка при обработке всплывающей информации

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта были составлены UML и ER диаграммы, созданы база данных и приложение.

В результате анализа предметной области были выделены пользователи и функции, которые они могут выполнять, а также сущности, которые присутствуют в выбранной области.

По окончании проектирования приложения, была создана архитектура будущего приложения, а также его дизайн. Был создан словарь данных.

В процессе разработки приложения, было созданы все запланированные формы и реализованы все необходимые функции. Также само приложение было связано с базой данных.

На этапе тестирования, была проверена отказоустойчивость данной системы посредством стратегии черного ящика. Выявлены некоторые ошибки.

Для улучшения системы необходимо:

- расширить функционал редактирования животных;
- улучшить дизайн приложения;
- исправить ошибки, выявленные на этапе тестирования;
- более детальное уведомлять пользователей о ошибках, которые они совершают.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Windows Forms – Википедия [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Windows_Forms (дата обращения 20.06.2018).

2 Что такое MySQL. Взаимодействие с PHP – SoftTime [Электронный ресурс] – Режим доступа: http://www.softtime.ru/bookphp/gl12_1.php (дата обращения 20.06.2018).

3 MySQL Workbench – Википедия [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/MySQL_Workbench (дата обращения 20.06.2018).

4 Visual Studio Community – Microsoft [Электронный ресурс] – Режим доступа: <https://visualstudio.microsoft.com/ru/vs/community/?rr=https%3A%2F%2Fru.wikipedia.org%2F> (дата обращения 21.06.2018).

5 C Sharp – Википедия [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/C_Sharp (дата обращения 21.06.2018).

6 Поведенческое тестирование – Википедия [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Тестирование_по_стратегии_чёрного_ящика (дата обращения 21.06.2018).

ПРИЛОЖЕНИЕ А

(обязательное)

Листинг кода главной формы Form1.cs

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();

        static MySqlConnection connection = new
        MySqlConnection("Server=207.154.220.247;UserId=popov
        ka;Password=tqh3t4wxmsgqjq3k;Database=popovka_db");

        private DataSet get_data()
        {
            connection.Open();
            MySqlDataAdapter adapter = new
            MySqlDataAdapter("select * from popovka_db.user",
            connection);
            DataSet data = new DataSet();
            adapter.Fill(data);
            connection.Close();
            return data;
        }

        private void Form1_Load(object sender, EventArgs
        e)
        {

        }

        private void button2_Click(object sender,
        EventArgs e)
```

```

    {
        if (LoginBox.Text != "" && PassBox.Text != "")
        {
            DataSet data = get_data();
            DataTable table = data.Tables[0];
            bool logged = false;
            for (int i = 0; i < table.Rows.Count; i++)
            {
                if (LoginBox.Text ==
                table.Rows[i][1].ToString() && PassBox.Text ==
                table.Rows[i][2].ToString())
                {
                    Menu win = new
                    Menu(int.Parse(table.Rows[i][3].ToString()),
                    int.Parse(table.Rows[i][0].ToString()), this);
                    win.Show();
                    this.Visible = false;
                    logged = true;
                }
            }
            if(logged is false) MessageBox.Show("Данные
            не совпадают!", "Error", MessageBoxButtons.OK,
            MessageBoxIcon.Exclamation);
        }
        else MessageBox.Show("Данные не
        введены!", "Error", MessageBoxButtons.OK, MessageBoxIcon.
        Exclamation);
    }

    private void button1_Click(object sender,
    EventArgs e)
    {
        RegistrationForm win = new
        RegistrationForm(this);
        win.Show();
        this.Visible = false;
    }
}
```

```
    }  
  
    private void button1_Click_1(object sender,  
EventArgs e)  
    {  
        Menu win = new Menu(2, 1, this);  
        win.Show();  
        this.Visible = false;  
    }  
}
```


ПРИЛОЖЕНИЕ Б

(обязательное)

Листинг кода формы регистрации пользователей

RegistrationForm.cs

```
public partial class RegistrationForm : Form
{
    public RegistrationForm(Form1 par)
    {
        InitializeComponent();
        parent = par;
        fill();
    }
    Form1 parent;
    static MySqlConnection connection = new
    MySqlConnection("Server=207.154.220.247;UserId=popov
    ka;Password=tqh3t4wxmsgqjq3k;Database=popovka_db");

    private void fill()
    {
        connection.Open();
        MySqlDataAdapter types = new
        MySqlDataAdapter("select * from
        popovka_db.user_type", connection);
        MySqlDataAdapter facilit = new
        MySqlDataAdapter("select * from
        popovka_db.Facility", connection);

        DataSet data_types = new DataSet();
        DataSet data_facili = new DataSet();

        types.Fill(data_types);
        facilit.Fill(data_facili);
```

```
        UserType.DataSource = data_types.Tables[0];
        UserType.DisplayMember = "name";
        data_facili.Tables[0].Rows.Add();
        Facility.DataSource = data_facili.Tables[0];
        Facility.DisplayMember = "Name";
        connection.Close();
    }

    private void invisibility()
    {
        PassportData.Visible = false;
        label9.Visible = false;
        label10.Visible = false;
        Facility.Visible = false;
    }

    private void visibility()
    {
        PassportData.Visible = true;
        label9.Visible = true;
        label10.Visible = true;
        Facility.Visible = true;
    }

    private void CancelButton_Click(object sender,
    EventArgs e)
    {
        parent.Visible = true;
        this.Close();
    }

    private void
    UserType_SelectedIndexChanged(object sender,
    EventArgs e)
    {
        ComboBox current = sender as ComboBox;
```

```

        if (current.Text == "user") invisibility();
        else visibility();
    }

    private string whichtype()
    {
        MySqlCommand command = new
        MySqlCommand("select type from popovka_db.user_type
        where name='"+UserType.Text+"'", connection);
        connection.Open();
        MySqlDataReader reader =
        command.ExecuteReader();
        reader.Read();
        string type = reader[0].ToString();
        connection.Close();
        return type;
    }

    private bool logincheck()
    {
        connection.Open();
        MySqlDataAdapter adapter = new
        MySqlDataAdapter("select Login from
        popovka_db.user", connection);
        DataSet set = new DataSet();
        adapter.Fill(set);
        connection.Close();
        BindingSource bs = new BindingSource(set,
        set.Tables[0].TableName);
        if (bs.Find("Login", Login.Text) == -1) return
        true;
        else return false;
    }

    private string whichfacility()
    {

```

```

        MySqlCommand command = new
        MySqlCommand("select ID_facility from
        popovka_db.Facility where Name='" + Facility.Text +
        "'", connection);
        connection.Open();
        MySqlDataReader reader =
        command.ExecuteReader();
        reader.Read();
        string facil = reader[0].ToString();
        connection.Close();
        return facil;
    }

    private string user_id()
    {
        MySqlCommand command = new
        MySqlCommand("select ID_user from popovka_db.user
        where Login='" + Login.Text + "'", connection);
        connection.Open();
        MySqlDataReader reader =
        command.ExecuteReader();
        reader.Read();
        string id = reader.GetString(0);
        connection.Close();
        return id;
    }

    private void RegistrationButton_Click(object
    sender, EventArgs e)
    {
        bool correct = false;
        if (logincheck())
        {
            try
            {

```

```

        string com = "insert into
popovka_db.user(login, password, type) values('" +
Login.Text + "',''"
        + Password.Text + "',''" + whichtype() +
"'')";
        MySqlCommand command = new
MySqlCommand(com, connection);
        connection.Open();
        command.ExecuteNonQuery();
        connection.Close();
        correct = true;
    }
    catch
    {
        MessageBox.Show("Что-то не так с логином
или паролем!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
        connection.Close();
        correct = false;
    }
}
else MessageBox.Show("Такой логин уже
используется!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);

if (correct)
{
    try
    {
        if (whichtype() == "2")
        {
            string usercom = "insert into
popovka_db.client(FirstName, SecondName, LastName,
Phone, ID_user) values('" + FirstName.Text + "',''" +
                SecondName.Text + "',''" + LastName.Text
+ "',''" + Telephon.Text + "',''" + user_id() + "'')";

```

```

        MySqlCommand userCommand = new
MySqlCommand(usercom, connection);
        connection.Open();
        userCommand.ExecuteNonQuery();
        connection.Close();
    }
    else
    {
        string usercom;

        if (Facility.Text == "") usercom =
"insert into popovka_db.employee(FirstName,
SecondName, LastName, Phone, Passport_data, ID_user)
values('" + FirstName.Text + "',''" +
                SecondName.Text + "',''" + LastName.Text
+ "',''" + Telephon.Text + "',''" + PassportData.Text
+ "',''" + user_id() + "'')";
        else usercom = "insert into
popovka_db.employee(FirstName, SecondName, LastName,
Phone, Passport_data, Facility, ID_user) values('" +
FirstName.Text + "',''" +
                SecondName.Text + "',''" + LastName.Text
+ "',''" + Telephon.Text + "',''" + PassportData.Text
+ "',''" + whichtfacility() + "',''" + user_id() +
"'')";

        MySqlCommand userCommand = new
MySqlCommand(usercom, connection);
        connection.Open();
        userCommand.ExecuteNonQuery();
        connection.Close();
    }
    MessageBox.Show("Поздравляю, вы
зарегистрированы!");
    parent.Visible = true;
    this.Close();
}

```

```

    }

    catch
    {
        MySqlCommand del_com = new
        MySqlCommand("delete from popovka_db.user where
        login='" + Login.Text + "'", connection);
        del_com.ExecuteNonQuery();
        connection.Close();
        MessageBox.Show("Вы где-то совершили
        ошибку!", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Exclamation);
    }
}

private void RegistrationForm_Load(object
sender, EventArgs e)
{

}

private void RegistrationForm_FormClosed(object
sender, FormClosedEventArgs e)
{
    parent.Show();
}
}

```

ПРИЛОЖЕНИЕ В

(обязательное)

Листинг кода формы меню Menu.cs

```
public partial class Menu : Form
{
    public Menu(int type, int id_rec, Form1 par)
    {
        InitializeComponent();
        parent = par;
        id = id_rec;
        user_type = type;
    }
    Form1 parent;
    int id;
    int user_type;

    private void label4_Click(object sender,
EventArgs e)
    {
        parent.Visible = true;
        this.Close();
    }

    private void animal_click(object sender,
EventArgs e)
    {
        animal win = new animal(this, user_type);
        win.Show();
        this.Visible = false;
    }

    private void map_Click(object sender, EventArgs
e)
```

```
    {
        map win = new map(this);
        win.Show();
        this.Hide();
    }

    private void excursion_Click(object sender,
EventArgs e)
    {
        excursion win = new excursion(this, id,
user_type);
        win.Show();
        this.Visible = false;
    }

    private void tickets_Click(object sender,
EventArgs e)
    {
        Tickets win = new Tickets(this, id,
user_type);
        win.Show();
        this.Hide();
    }

    private void left_MouseLeave(object sender,
EventArgs e)
    {
        Label text = sender as Label;
        text.BackColor = Color.FromArgb(252, 200, 66);
    }

    private void left_MouseMove(object sender,
MouseEventArgs e)
    {
        Label text = sender as Label;
        text.BackColor = Color.FromArgb(255, 124, 0);
    }
}
```

```

    }

    private void label3_MouseUp(object sender,
MouseEventArgs e)
    {
        Label text = sender as Label;
        text.BorderStyle = BorderStyle.None;
        text.BackColor = Color.FromArgb(252, 200, 66);
    }

    private void label3_MouseDown(object sender,
MouseEventArgs e)
    {
        Label text = sender as Label;
        text.BorderStyle = BorderStyle.Fixed3D;
    }

    private void BG2_MouseLeave(object sender,
EventArgs e)
    {
        Label square = sender as Label;
        square.BackColor = Color.FromArgb(255, 183,
115);
    }

    private void BG2_MouseMove(object sender,
MouseEventArgs e)
    {
        Label square = sender as Label;
        square.BackColor = Color.FromArgb(255, 157,
64);
    }

    private void Menu_FormClosed(object sender,
FormClosedEventArgs e)
    {
        parent.Show();
    }
}

```

ПРИЛОЖЕНИЕ Г

(обязательное)

Листинг кода формы животных animal.cs

```
public partial class animal : Form
{
    public animal(Menu par, int type)
    {
        InitializeComponent();
        parent = par;
        fill_grid();
        if (type == 2) Add_animal.Hide();
    }
    Menu parent;
    static MySqlConnection connection = new
    MySqlConnection("Server=207.154.220.247;UserId=popov
    ka;Password=tqh3t4wxmsgjq3k;Database=popovka_db");

    public void fill_grid()
    {
        connection.Open();
        MySqlDataAdapter adapter = new
        MySqlDataAdapter("select Species, Family, Location,
        Facility.Name as 'Facility' " +
            "from popovka_db.animal left join Facility
        on (animal.Facility = Facility.ID_facility)",
        connection);
        MySqlDataAdapter facilit = new
        MySqlDataAdapter("select * from
        popovka_db.Facility", connection);

        DataSet data_facili = new DataSet();
        DataSet data = new DataSet();
```

```
        adapter.Fill(data);
        facilit.Fill(data_facili);

        DataRow all = data_facili.Tables[0].NewRow();
        all["Name"] = "All";
        data_facili.Tables[0].Rows.Add(all);

        BindingSource bs = new BindingSource(data,
        data.Tables[0].TableName);
        Facility.DataSource = data_facili.Tables[0];
        Facility.DisplayMember = "Name";
        Facility.SelectedIndex =
        data_facili.Tables[0].Rows.Count - 1;

        dataGridView1.DataSource = bs;
        connection.Close();
    }

    private void Search_Click(object sender,
    EventArgs e)
    {
        BindingSource bs = dataGridView1.DataSource as
        BindingSource;
        int position = bs.Find("Species",
        textBox1.Text);
        if (position > 0) bs.Position = position;
        else MessageBox.Show("У нас нет такого
        животного(");
    }

    private void
    Facility_SelectionChangeCommitted(object sender,
    EventArgs e)
    {
        MySqlDataAdapter adapter;
```

```

        if (Facility.Text != "All")
        {
            connection.Open();
            adapter = new MySqlDataAdapter("select
Species, Family, Location, Facility.Name as
'Facility' " +
            "from popovka_db.animal right join
Facility on (animal.Facility =
Facility.ID_facility)" +
            "where Facility.Name ='" + Facility.Text +
            "'", connection);
        }
        else
        {
            adapter = new MySqlDataAdapter("select
Species, Family, Location, Facility.Name as
'Facility' " +
            "from popovka_db.animal right join Facility
on (animal.Facility = Facility.ID_facility)",
connection);
        }
        DataSet data = new DataSet();

        adapter.Fill(data);
        BindingSource bs = new BindingSource(data,
data.Tables[0].TableName);
        dataGridView1.DataSource = bs;
        connection.Close();
    }

    private void CancelButton_Click(object sender,
EventArgs e)
    {
        parent.Visible = true;
        this.Close();
    }

```

```

    private void add_click(object sender, EventArgs
e)
    {
        animal_create win = new animal_create(this);
        this.Hide();
        win.Show();
    }

    private void textBox1_TextChanged(object sender,
EventArgs e)
    {
        BindingSource bs = dataGridView1.DataSource as
BindingSource;
        bs.Position = bs.Find("Species",
textBox1.Text);
    }

    private void animal_FormClosed(object sender,
FormClosedEventArgs e)
    {
        parent.Show();
    }
}

```


ПРИЛОЖЕНИЕ Д

(обязательное)

Листинг кода формы добавления нового животного

animal_create.cs

```
public partial class animal_create : Form
{
    public animal_create(animal par)
    {
        InitializeComponent();
        parent = par;
        Food_label.Text = Food.Value.ToString();
        fill_facility();
    }
    animal parent;
    static MySqlConnection connection = new
    MySqlConnection("Server=207.154.220.247;UserId=popov
    ka;Password=tqh3t4wxmsgqjq3k;Database=popovka_db");

    private void fill_facility()
    {
        connection.Open();
        MySqlDataAdapter facilit = new
        MySqlDataAdapter("select * from
        popovka_db.Facility", connection);

        DataSet data_facili = new DataSet();

        facilit.Fill(data_facili);

        Facility.DataSource = data_facili.Tables[0];
        Facility.DisplayMember = "Name";
        connection.Close();
```

```
        DataRow all = data_facili.Tables[0].NewRow();
        all["Name"] = "Choose";
        data_facili.Tables[0].Rows.Add(all);
        Facility.SelectedIndex =
        data_facili.Tables[0].Rows.Count - 1;
    }
    private void CancelButton_Click(object sender,
    EventArgs e)
    {
        parent.Visible = true;
        this.Close();
    }

    private void Food_ValueChanged(object sender,
    EventArgs e)
    {
        Food_label.Text = Food.Value.ToString();
    }

    private int whichfacility()
    {
        MySqlCommand command = new
        MySqlCommand("select ID_facility from
        popovka_db.Facility where Name='" + Facility.Text +
        "'", connection);
        connection.Open();
        MySqlDataReader reader =
        command.ExecuteReader();
        reader.Read();
        int facil = Int32.Parse(reader[0].ToString());
        connection.Close();
        return facil;
    }

    private void update_facili()
```

```

    {
        connection.Open();
        MySqlDataAdapter adapter = new
        MySqlDataAdapter("select * from popovka_db.animal
        left join Facility on(animal.Facility =
        Facility.ID_facility) where Facility.Name =
        '"+Facility.Text+"'", connection);
        DataSet set = new DataSet();
        adapter.Fill(set);

        MySqlCommand command = new
        MySqlCommand("update popovka_db.Facility set
        Amount=" + set.Tables[0].Rows.Count+" where
        Name='"+Facility.Text+"'", connection);
        command.ExecuteNonQuery();
        connection.Close();
    }
    private void Add_button_Click(object sender,
    EventArgs e)
    {
        try
        {
            string insert_str = "insert into
            animal(Species, Family, Food_per_day, Location,
            Facility) " +
            "values('" + Species.Text + "', '" +
            Family.Text + "'," + Food_label.Text + "', '" +
            Location.Text + "'," + whichfacility() + ")";
            MySqlCommand insert = new
            MySqlCommand(insert_str, connection);
            connection.Open();
            insert.ExecuteNonQuery();
            connection.Close();
            MessageBox.Show("Животное добавлено!");
            parent.fill_grid();
            update_facili();
        }
    }

```

```

    }
    catch
    {
        connection.Close();
        MessageBox.Show("Что-то пошло не так" +
        Environment.NewLine+ "Скорее всего вы где-то
        ошиблись", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Exclamation);
    }

    }

    private void animal_create_FormClosed(object
    sender, FormClosedEventArgs e)
    {
        parent.Show();
    }
}

```

ПРИЛОЖЕНИЕ Е

(обязательное)

Листинг кода формы экскурсий excursion.cs

```
public partial class excursion : Form
{
    public excursion(Menu par, int id, int type)
    {
        InitializeComponent();
        parent = par;
        user_id = id;
        fill_facil(type);
    }
    Menu parent;
    static MySqlConnection connection = new
    MySqlConnection("Server=207.154.220.247;UserId=popov
    ka;Password=tqh3t4wxmsgqjq3k;Database=popovka_db");
    int user_id;

    private void fill_facil(int type)
    {
        connection.Open();
        MySqlDataAdapter facilit = new
        MySqlDataAdapter("select Name from
        popovka_db.Facility", connection);

        DataSet data_facili = new DataSet();

        facilit.Fill(data_facili);
        connection.Close();

        DataRow all = data_facili.Tables[0].NewRow();
        all["Name"] = "Choose";
```

```
        data_facili.Tables[0].Rows.Add(all);

        Facility.DataSource = data_facili.Tables[0];
        Facility.DisplayMember = "Name";
        Facility.SelectedIndex =
        data_facili.Tables[0].Rows.Count - 1;

        if(type == 1) Registration.Hide();
    }

    private void fill_empl()
    {
        MySqlDataAdapter empl = new
        MySqlDataAdapter("select employee.LastName as
        'Name', Facility.Name as 'Facility' from employee "
        +
        "right join excursion on
        (employee.ID_employee = excursion.Employee) " +
        "left join Facility on (employee.Facility =
        Facility.ID_facility) where Facility.Name='" +
        Facility.Text + "'", connection);
        connection.Open();

        DataSet data_empl = new DataSet();

        empl.Fill(data_empl);
        connection.Close();

        Employee.DataSource = data_empl.Tables[0];
        Employee.DisplayMember = "Name";
    }

    private void fill_date()
    {
```

```

        MySqlDataAdapter date = new
        MySqlDataAdapter("select excursion.Date,
        employee.LastName as 'Employee', Facility.Name as
        'Facility' from excursion " +
            "left join employee on (excursion.Employee =
        employee.ID_employee) " +
            "left join Facility on (excursion.Facility =
        Facility.ID_facility) " +
            "where Facility.Name='" + Facility.Text +
        "'" +
            "and (employee.LastName +
        employee.SecondName + employee.FirstName)='" +
        Employee.Text + "'", connection);
        connection.Open();

        DataSet data_date = new DataSet();

        date.Fill(data_date);
        connection.Close();

        exc_date.DataSource = data_date.Tables[0];
        exc_date.DisplayMember = "Date";
    }

    private void
    Facility_SelectedValueChanged(object sender,
    EventArgs e)
    {
        fill_empl();
        fill_date();
    }

    private void
    Employee_SelectedValueChanged(object sender,
    EventArgs e)
    {

```

```

        fill_date();
    }

    private void CancelButton_Click(object sender,
    EventArgs e)
    {
        parent.Visible = true;
        this.Close();
    }

    private void Registration_Click(object sender,
    EventArgs e)
    {
        try
        {
            DateTime date =
            DateTime.Parse(exc_date.Text);
            string id_com = "select ID_excursion from
            popovka_db.excursion " +
                "left join popovka_db.Facility on
            (excursion.Facility = Facility.ID_facility) " +
                "left join popovka_db.employee
            on(excursion.Employee = employee.ID_employee)" +
                "where Facility.Name = '"+Facility.Text+
            "' and employee.LastName = '" + Employee.Text+"'" and
            excursion.Date = '"+ date.ToString("yyyy-MM-dd") +
            "'";

            connection.Open();
            MySqlCommand id = new MySqlCommand(id_com,
            connection);
            MySqlDataReader reader = id.ExecuteReader();

            string com = "";

            while (reader.Read())
            {

```

```

        com = "update popovka_db.client set
Excursion =' " + reader.GetValue(0) + "' +
        "where ID_user =" + user_id;
    }

    MySqlCommand command = new MySqlCommand(com,
connection);
    reader.Close();

    DialogResult result = MessageBox.Show("Вы
уверены что хотите записаться на экскурсию в " +
Facility.Text + "?", "Подтверждение",
MessageBoxButtons.YesNo);
    if(result == DialogResult.Yes)
    {
        string check_text = "select Excursion from
popovka_db.client where ID_user=" + user_id;
        MySqlCommand command_check = new
MySqlCommand(check_text, connection);
        MySqlDataReader check_read =
command_check.ExecuteReader();
        DialogResult result_check =
DialogResult.Yes;
        while (check_read.Read()) result_check =
MessageBox.Show("Вы уже записаны на экскурсию,
перезаписать?", "Подтверждение",
MessageBoxButtons.YesNo);
        check_read.Close();
        if (result_check == DialogResult.Yes)
command.ExecuteNonQuery();
    }
    connection.Close();
    MessageBox.Show("Вы успешно записались на
экскурсию!");
}
catch

```

```

    {
        MessageBox.Show("Неудача."+Environment.NewLine+"Може
т быть вы не все выбрали?", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        connection.Close();
    }
}

private void excursion_FormClosed(object sender,
FormClosedEventArgs e)
{
    parent.Show();
}
}

```

ПРИЛОЖЕНИЕ Ж

(обязательное)

Листинг кода формы карты map.cs

```
public partial class map : Form
{
    public map(Menu par)
    {
        InitializeComponent();
        parent = par;
        earth.Parent = pictureBox1;

        air.Parent = pictureBox1;
        ocean.Parent = pictureBox1;
        cross.Parent = pictureBox1;
        pictureBox1.MouseMove += null;
        pictureBox1.MouseLeave += null;
        collect_info();
    }
    Menu parent;
    static MySqlConnection connection = new
    MySqlConnection("Server=207.154.220.247;UserId=popov
    ka;Password=tqh3t4wxmsgjq3k;Database=popovka_db");

    string earth_text;
    string ocean_text;
    string cross_text;
    string air_text;

    Label current = null;

    private void collect_info()
    {
```

```
        earth_text = "Корпус содержит огромное
        количество животных, проживающих на суше." +
        Environment.NewLine +
        "На данный момент в корпусе находится " +
        get_info("Amount", "Earth World") + " животных" +
        Environment.NewLine +
        "В корпусе так же имеется:" +
        Environment.NewLine +
        "Водоем: " + logic_int("Pool", "Earth
        World") + Environment.NewLine +
        "Подогрев: " + logic_int("Heating", "Earth
        World");
        ocean_text = "Этот корпус посвящен океану и
        его обитателям, которые поразят вас своим
        разнообразием." + Environment.NewLine +
        "На данный момент в корпусе находится " +
        get_info("Amount", "Ocean World") + " животных" +
        Environment.NewLine +
        "В корпусе так же имеется:" +
        Environment.NewLine +
        "Водоем: " + logic_int("Pool", "Ocean
        World") + Environment.NewLine +
        "Подогрев: " + logic_int("Heating", "Ocean
        World");
        air_text = "Здесь вы найдете разные виды птиц.
        К некоторым из которых можно прикоснуться." +
        Environment.NewLine +
        "На данный момент в корпусе находится " +
        get_info("Amount", "Sky World") + " животных" +
        Environment.NewLine +
        "В корпусе так же имеется:" +
        Environment.NewLine +
        "Водоем: " + logic_int("Pool", "Sky World")
        + Environment.NewLine +
        "Подогрев: " + logic_int("Heating", "Sky
        World");
```

```

        cross_text = "Животные разных видов и из
разных мест обитания собранные здесь." +
Environment.NewLine +
        " Лучший вариант для тех, кто хочет
посмотреть все и сразу." + Environment.NewLine +
        "На данный момент в корпусе находится " +
get_info("Amount", "Sky World") + " животных" +
Environment.NewLine +
        "В корпусе так же имеется:" +
Environment.NewLine +
        "Водоем: " + logic_int("Pool", "Sky World")
+ Environment.NewLine +
        "Подогрев: " + logic_int("Heating", "Sky
World");
    }

    private void CancelButton_Click(object sender,
EventArgs e)
    {
        parent.Visible = true;
        this.Close();
    }

    private void label4_Click(object sender,
EventArgs e)
    {
    }

    private string get_info(string which, string
name)
    {
        string com = "select " + which + " from
popovka_db.Facility where Name='" + name + "'";

```

```

        MySqlCommand command = new MySqlCommand(com,
connection);

        connection.Open();
        MySqlDataReader reader =
command.ExecuteReader();

        reader.Read();
        string answer = reader[0].ToString();
        connection.Close();
        return answer;
    }

    private string logic_int(string prop, string
world)
    {
        if (Boolean.Parse(get_info(prop, world)))
return "Да";
        else return "Нет";
    }

    private void generall_MouseLeave(object sender,
EventArgs e)
    {
        current.Dispose();
        current = null;
    }

    private void generall_MouseMove(object sender,
MouseEventArgs e)
    {
        if (current == null)
        {
            Label info = new Label();
            Label obj = sender as Label;
            if(obj.Name == "earth")

```

```

        {
            info.Text = earth_text;
        }
        else if (obj.Name == "cross")
        {
            info.Text = cross_text;
        }
        else if(obj.Name == "ocean")
        {
            info.Text = ocean_text;
        }
        else
        {
            info.Text = air_text;
        }

        info.Parent = pictureBox1;
        info.AutoSize = true;
        info.BackColor = Color.FromArgb(220, 35, 91,
121);
        info.ForeColor = Color.FromArgb(255, 193,
115);
        info.Font = new Font("Tahoma", 11);
        info.Location = e.Location;
        info.BringToFront();
        info.Show();
        current = info;
    }
}

private void map_FormClosed(object sender,
FormClosedEventArgs e)
{
    parent.Show();
}
}

```


ПРИЛОЖЕНИЕ 3

(обязательное)

Листинг кода формы билетов Tickets.cs

```
public partial class Tickets : Form
{
    public Tickets(Menu par, int id, int type)
    {
        InitializeComponent();
        parent = par;
        user_id = id;
        FinalSumm.Text = "0";
        if (type == 2) Export_button.Hide();
    }
    Menu parent;
    int user_id;
    static MySqlConnection connection = new
    MySqlConnection("Server=207.154.220.247;UserId=popov
    ka;Password=tqh3t4wxmsgqjq3k;Database=popovka_db");

    private void CancelButton_Click(object sender,
    EventArgs e)
    {
        parent.Visible = true;
        this.Close();
    }

    private void checkEarth_CheckStateChanged(object
    sender, EventArgs e)
    {
        CheckBox box = sender as CheckBox;
        if (box.Checked) EarthLabel.Show();
        else EarthLabel.Hide();
        value_count();
    }
```

```
    }

    private void checkOcean_CheckStateChanged(object
    sender, EventArgs e)
    {
        CheckBox box = sender as CheckBox;
        if (box.Checked) OceanLabel.Show();
        else OceanLabel.Hide();
        value_count();
    }

    private void checkSky_CheckStateChanged(object
    sender, EventArgs e)
    {
        CheckBox box = sender as CheckBox;
        if (box.Checked) SkyLabel.Show();
        else SkyLabel.Hide();
        value_count();
    }

    private void checkWorld_CheckStateChanged(object
    sender, EventArgs e)
    {
        CheckBox box = sender as CheckBox;
        if (box.Checked) CrossLabel.Show();
        else CrossLabel.Hide();
        value_count();
    }

    private void value_count()
    {
        double summ = 0;
        if (checkEarth.Checked) summ += 150;
        if (checkOcean.Checked) summ += 150;
        if (checkSky.Checked) summ += 150;
        if (checkWorld.Checked) summ += 200;
    }
```

```

        double mature = summ *
int.Parse(numericMature.Value.ToString());
        double children = summ * 0.5 *
int.Parse(numericChildren.Value.ToString());

        FinalSumm.Text = (mature +
children).ToString();
    }
    private void Value_changed(object sender,
EventArgs e)
    {
        value_count();
    }

    private void FinalSumm_Click(object sender,
EventArgs e)
    {

    }

    private int transform(bool stat)
    {
        if (stat) return 1;
        else return 0;
    }

    private void buttonBuy_Click(object sender,
EventArgs e)
    {
        int res =
DateTime.Compare(dateTimePicker1.Value,
DateTime.Now);
        if (Int32.Parse(FinalSumm.Text) != 0 &&
DateTime.Compare(dateTimePicker1.Value,
DateTime.Now) > 0)

```

```

    {
        string comm = "insert into
popovka_db.ticket(user_id, Date, Cost, Earth, Sky,
Ocean, Crossed) values('" +
        user_id + "',''" +
dateTimePicker1.Value.ToString("yyyy-MM-dd") + "',''"
+ FinalSumm.Text + "',''" +
transform(checkEarth.Checked) + "',''" +
        transform(checkSky.Checked) + "',''" +
transform(checkOcean.Checked) + "',''" +
transform(checkWorld.Checked) + "')";
        MySqlCommand command = new
MySqlCommand(comm, connection);
        connection.Open();
        command.ExecuteNonQuery();
        connection.Close();
        MessageBox.Show("Покупка совершена.");
    }
    else
if(DateTime.Compare(dateTimePicker1.Value,
DateTime.Now) < 0) MessageBox.Show("Дата должна быть
установлена не раньше сегодняшнего дня!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        else MessageBox.Show("Вы еще ничего не
выбрали!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
    }

    private void Tickets_FormClosed(object sender,
FormClosedEventArgs e)
    {
        parent.Show();
    }

    private void Tickets_Load(object sender,
EventArgs e)

```

```
{  
  
}  
  
    private void Export_button_Click(object sender,  
EventArgs e)  
    {  
        ExportForm win = new ExportForm(this);  
        this.Hide();  
        win.Show();  
    }  
}
```

ПРИЛОЖЕНИЕ И

(обязательное)

Листинг кода формы выгрузки данных в Excel ExportForm.cs

```
public partial class ExportForm : Form
{
    public ExportForm(Tickets par)
    {
        InitializeComponent();
        parent = par;
        load_data();
    }
    Tickets parent;

    private void load_data()
    {
        MySqlConnection connection = new
        MySqlConnection("Server=207.154.220.247;UserId=popov
        ka;Password=tqh3t4wxmsgqjq3k;Database=popovka_db");
        connection.Open();
        MySqlDataAdapter ad = new
        MySqlDataAdapter("select * from popovka_db.ticket",
        connection);
        DataSet set = new DataSet();
        ad.Fill(set);
        BindingSource bs = new BindingSource(set,
        set.Tables[0].TableName);
        dataGridView1.DataSource = bs;
        connection.Close();
    }

    private void CancelButton_Click(object sender,
    EventArgs e)
    {
```

```
        parent.Visible = true;
        this.Close();
    }

    private void ExportForm_FormClosed(object
    sender, FormClosedEventArgs e)
    {
        parent.Show();
    }

    private void copyAlltoClipboard()
    {
        dataGridView1.SelectAll();
        DataObject dataObj =
        dataGridView1.GetClipboardContent();
        if (dataObj != null)
            Clipboard.SetDataObject(dataObj);
    }

    private void Export_Click(object sender,
    EventArgs e)
    {
        copyAlltoClipboard();
        Microsoft.Office.Interop.Excel.Application
        xlexcel;
        Microsoft.Office.Interop.Excel.Workbook
        xlWorkBook;
        Microsoft.Office.Interop.Excel.Worksheet
        xlWorkSheet;
        object misValue =
        System.Reflection.Missing.Value;
        xlexcel = new
        Microsoft.Office.Interop.Excel.Application();
        xlexcel.Visible = true;
        xlWorkBook = xlexcel.Workbooks.Add(misValue);
```

```
xlWorksheet =  
(Microsoft.Office.Interop.Excel.Worksheet)xlWorkbook  
.Worksheets.get_Item(1);  
    Microsoft.Office.Interop.Excel.Range CR =  
(Microsoft.Office.Interop.Excel.Range)xlWorksheet.Cells[1, 1];  
    CR.Select();  
    xlWorksheet.PasteSpecial(CR, Type.Missing,  
Type.Missing, Type.Missing, Type.Missing,  
Type.Missing, true);  
}  
  
}
```

Сведения о самостоятельности выполнения работы

Работа _____ выполнена мной
самостоятельно.

Используемые в работе материалы и концепции из публикуемой литературы
и других источников имеют ссылки на них.

Один печатный экземпляр работы и электронный вариант работы на
цифровом носителе переданы мной на кафедру.

«___» июня 2018г.

(подпись)

К. А. Попов
(инициалы, фамилия)