Одеський національний політехнічний університет

Кафедра «Комп'ютеризовані системи управління»

**КУРСОВА РОБОТА**

з дисципліни «Програмування та теорія алгоритмів»

на тему: «Iнформаційна база для автоматизації роботи автостоянки»

Варіант 8

Студента I курсу, групи АТ-183

спеціальності «Автоматизація та комп'ютерно-інтегровані технології»

Прізвище: Демиров А.О.

Керівник: доц. Сперанський В.О.

Національна шкала:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

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

(підпис) (прізвище та ініціали)

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

(підпис) (прізвище та ініціали)

м. Одеса – 2019 рік

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc10144558)

[Постановка задачи 4](#_Toc10144559)

[РАЗДЕЛ 1 ПРОЕКТИРОВАНИЕ И ИХ РЕАЛИЗАЦИЯ 6](#_Toc10144560)

[1.1 Календарное планирование проекта 6](#_Toc10144561)

1. Описание проектирования программы
   * 1. Создание интерфейса пользователя
     2. Основная форма
2. Краткое описание продукта

# ВВЕДЕНИЕ

**База данных** – это организованная структура, предназначенная для хранения информации. В те годы, когда формировалось понятие баз данных, в них хранились только данные, однако сегодня большинство систем управления базами данных (СУБД) позволяет размещать в своих структурах не только данные, но и методы (то есть программный код), с помощью которых происходит взаимодействие с потребителем или с другими программно-аппаратными комплексами. В современных базах данных хранятся не только данные, но и информация.

Под базой данных понимают хранилище структурированных данных. При этом данные должны быть непротиворечивы, минимально избыточны и целостны. База данных создаётся для хранения и доступа к данным, содержащим сведения о некоторой предметной области. Степень детализации информации определяется рядом факторов. Прежде всего, целью использования информации из баз данных и сложностью информационных процессов, существующих в пределах предметной области в конкретных условиях.

Постановка задачи

1. Спроектировать структуру информационной базы для своего варианта задания (Приложение А).
2. Создать основную таблицу информационной базы в виде динамического массива структур (использование списка из элементов типа структура оценивается выше).
3. Предусмотреть функции контроля корректности всех вводимых данных (например, неотрицательный возраст, дата продажи не ранее, чем 100 лет назад и т.п.).
4. Заполнить основную таблицу не менее чем 20 записями.
5. Обеспечить сохранение и загрузку данных во внешний файл/из файла.
6. Реализовать меню пользователя, соответствующие функции которого обеспечивают:
7. создание информационной базы той предметной области, которая определена постановкой задачи;
8. добавление новых записей в базу данных (с контролем корректности вводимых значений);
9. поиск записей по *ключевому* полю;
10. редактирование записей базы данных;
11. удаление ненужных записей;
12. выдачу отчётов по запросу.
13. Выполнить оригинальную обработку, указанную в постановке конкретной задачи для вашего варианта (Приложение А).
14. Организовать ввод пароля при выполнении операций изменения данных.
15. Добавить справочную систему пользователя по созданному приложению.
16. Подготовить презентацию по разработанному приложению.

**Постановка задачи варианта №8**

1. Разработать информационную базу для автоматизации работы автостоянки. На каждого владельца места коллективной автостоянки заводятся учётная карта, которая содержит:

* *номер места на автостоянке;*
* ФИО владельца автомобиля;
* марка автомобиля;
* отметка о наличии автомобиля на стоянке;
* отметка об оплате за текущий месяц.

*Предусмотреть:*

а) выдачу сведений о свободных автостоянках в данный момент;

б) выдачу сведений о владельцах автомобилей, не оплативших за пользование автостоянкой;

в) удаление из базы записей о владельце автомобиля с указанной фамилией;

г) добавление новой записи после записи с указанной фамилией.

# РАЗДЕЛ 1 ПРОЕКТИРОВАНИЕ И ИХ РЕАЛИЗАЦИЯ

## Календарное планирование проекта

Поэтапное календарное планирование для выполняемого курсового проекта представлено в табл. 1.1.

Таблица 1.1 – Календарный план курсового проектирования

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование | Содержание | Дата окончания |
| 1 | Начало | Изучение постановочного задания, предметной области | 24.04.2019 |
| 2 | Теоретическая подготовка | Изучение литературы по дисциплины | 25.04.2019 |
| 3 | Проектирование модулей проекта | Разработка компонентов программы, их функций, свойств, а также связь между ними. | 30.04.2019 |
| 4 | Разработка программной реализации | Написание непосредственно самой программы с использованием изученных теоретических знаний | 03.05.2019 |
| 5 | Тестирование | Проверка созданного приложения на предмет некорректного поведения при вводе ошибочных значений, определение уровня эргономики, устойчивости разработки | 19.05.2019 |
| 5 | Написание документации | Написание документации с описанием выполненной работы и трудностями с которыми пришлось встретиться во время её выполнения | 22.05.2019 |
| 6 | Создание презентации | Выбор наиболее интересных этапов написания программы и подготовка презентации | 28.05.2019 |

1. Описание проектирования программы
2. Создание базы данных. Проектирование архитектуры приложения начинается с формирования базы данных согласно пунктам задания. Экземпляром базы данных в коде является класс «Parking», полями которого являются: строчные поля «Владелец», «Модель авто», «Оплата за текущий месяц», «Нахождение авто на стоянке», а так же целочисленные поля «Номер парко-места». Пример кода:

class Parking

{

//Поля класса

private int parkingSpaceNum;

private string carOwner;

private string carModel;

private string parkingMark;

private string paymentMark;

//Конструктор по умолчанию

public Parking()

{

}

//Конструктор с входными параметрами

public Parking(int parkingSpaceNum, string carOwner, string carModel, string parkingMark, string paymentPark)

{

this.ParkingSpaceNum = parkingSpaceNum;

this.CarOwner = carOwner;

this.CarModel = carModel;

this.ParkingMark = parkingMark;

this.PaymentMark = paymentPark;

}

// Инкапсуляция полей класса

public int ParkingSpaceNum { get => parkingSpaceNum; set => parkingSpaceNum = value; }

public string CarOwner { get => carOwner; set => carOwner = value; }

public string CarModel { get => carModel; set => carModel = value; }

public string ParkingMark { get => parkingMark; set => parkingMark = value; }

public string PaymentMark { get => paymentMark; set => paymentMark = value; }

//Метод записи свйоств класса в бинарный файл

public void Write(BinaryWriter bw)

{

// Все данные записываются по отдельности

bw.Write(ParkingSpaceNum);

bw.Write(CarOwner);

bw.Write(CarModel);

bw.Write(ParkingMark);

bw.Write(PaymentMark);

}

//Метод чтения свойств класса из бинарного файла

public static Parking Read(BinaryReader br)

{

Parking p = new Parking();

p.ParkingSpaceNum = br.ReadInt32();

p.CarOwner = br.ReadString();

p.CarModel = br.ReadString();

p.ParkingMark = br.ReadString();

p.PaymentMark = br.ReadString();

return p;

}

}

* + 1. Функции программы

При описании методов работы со списком были выполнены такие задачи как: очистить все записи, сведенья о пустых местах, редактирование списка, очистка списка, обновление таблицы, открытие базы, сохранение базы и загрузка базы. Пример кода предоставлен ниже

private void ОткрытьToolStripMenuItem\_Click(object sender, EventArgs e)

private void СохранитьToolStripMenuItem\_Click(object sender, EventArgs e)

private void ПоказатьВсеЗаписиToolStripMenuItem\_Click(object sender, EventArgs e)

Также в программе были созданы функции, предусмотренные постановкой задачи:

* Функция, которая удаляет все данные
* Функция вывода данных о свободных местах на парковке
* Функция вывода данных о не оплаченных месяцев
* Функция вывода все записей в файле
* Функция проверки корректности ввода. Проверка строки является ли она положительным числом.
* Функция, которая удаляет данные по указанному имени и фамилии
* Функция, которая изменяет данные по указанной строчке.

Пример кода:

private void ПоказатьВсеЗаписиToolStripMenuItem\_Click(object sender, EventArgs e)

private void ОткрытьToolStripMenuItem\_Click(object sender, EventArgs e)

private void ОчиститьВсеЗаписиToolStripMenuItem\_Click(object sender, EventArgs e)

Запись списка в файл осуществлялось с помощью цикла используя метод if():

private void Button1\_Click(object sender, EventArgs e)

{

if (InputIsPositiveDigit(textBox2.Text))

{

parking.Add(new Parking()

{

ParkingSpaceNum = Convert.ToInt32(textBox2.Text),

CarOwner = textBox1.Text,

CarModel = textBox3.Text,

ParkingMark = comboBox1.Text,

PaymentMark = comboBox2.Text

});

}

else

{

MessageBox.Show("Номер парко-места должен быть положительным числом!");

}

UpdateT();

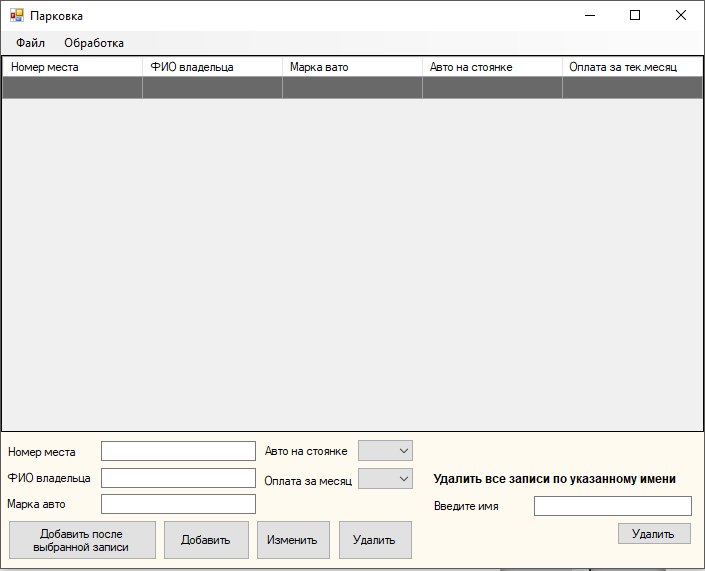
}

* + 1. Создание интерфейса пользователя

Интерфейс состоит из одной формы, это основная форма в стиле «Windows Form». Внешний вид стандартный.

* + 1. Основная форма

Основная форма является рабочей областью. Именно в ней происходят все операции над БД.

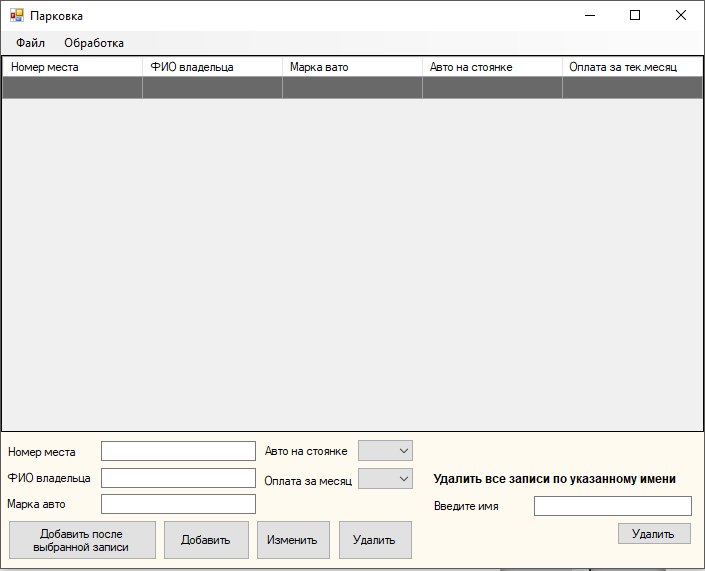


# РАЗДЕЛ 2 СПРАВОЧНАЯ ИНФОРМАЦИЯ ДЛЯ ПОЛЬЗОВАТЕЛЯ

1. Краткое описание продукта

Приложение представляет собой ПО, в котором разработана база данных «Автостоянки» или «Парковки». Функции программы:

* Номер места
* ФИО владельца
* Марка Авто
* Авто на стоянке
* Оплата за тек.месяц
  + 1. Панель управления



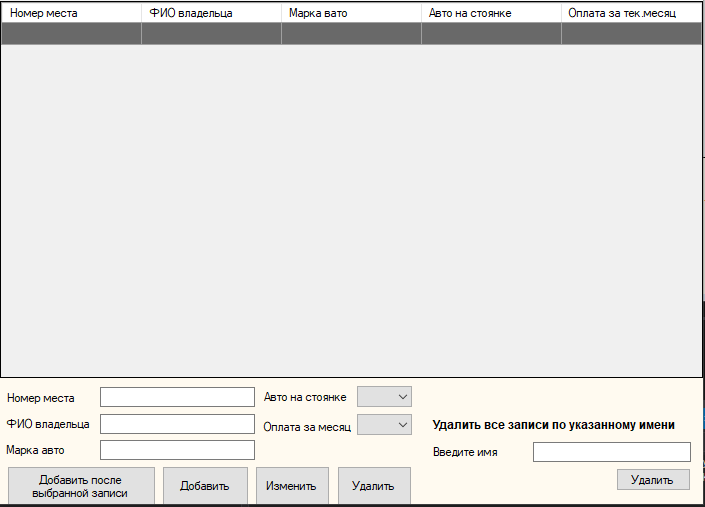
Меню приложения



Заголовки столбцов



Содержимое вкладки



Кнопка «Добавить после выбранной записи»



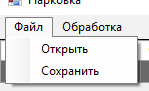
Кнопка «Добавить»

 Кнопка «Удалить и Изменить»

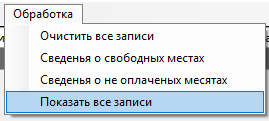


Меню имеет 2 вкладки: «Файл» и «Обработка»

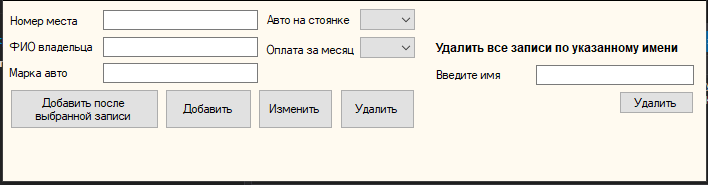
Через вкладку «Файл»: нужно загрузить данные из файла для работы приложение или сохранить в новый фал или уже существующий.



Вкладка «Обработка» отвечает за обработку записей, с помощью этой вкладки можно удалить все записи в файле, вывести даные о неоплаченых парко-мест в тек. Месц., и вывести даные про авто которые находятся на парковке, а также кнопка «Показать все записи» отвечает за выведение всех записей из файла на экран.



В нижней части приложения есть поле где мы вписываем новые данные и добавляем их, или мы можем вписать новые данные кликнуть на нужную строчку и заменить их с помощью кнопки «Изменить»



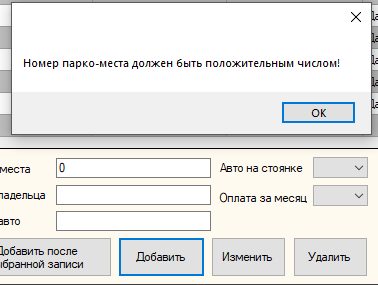
Ввод данных

Ввод некоторых данных производится с помощью специальных полей:

* Номер места
* ФИО владельца
* Марка авто
* Авто на стоянке
* Оплата за месяц

При вписывании новых данных таблица автоматически обновляется

При вводе в поле «Номер места» значении меньшего от 1 то выведется сообщение



# ВЫВОДЫ

На основании знаний, приобретенных на курсе «Основы технологии .Net», был спроектирован программный продукт, который автоматизирует управление базой данных, упрощая поиск, редактирование и удаление информации, и экономя время на ее обработку. В ходе выполнения курсовой работы были закреплены навыки ООП, создания пользовательских интерфейсов.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Классы и объекты [Электронный ресурс] // 27.07.2018. – 2018. – Режим доступа к ресурсу: <https://bit.ly/2Wcpboe>.
2. Список List [Электронный ресурс] // 31.10.2015. – 2015. – Режим доступа к ресурсу: <https://bit.ly/2VTuzxn>.
3. Оформление дизайна программы [Электронный ресурс] – Режим доступа к ресурсу: <https://bit.ly/1EUK3Mq>.
4. Настройка таблицы [Электронный ресурс] – Режим доступа к ресурсу: <https://bit.ly/2LTviK5>.
5. Шилдт Г. C# 4.0: полное руководство / Герберт Шилдт., 2010.
6. Работа с Windows Forms [Электронный ресурс] – Режим доступа к ресурсу:

<https://bit.ly/2YvYvfA>.