Министерство образования и науки РФ

Государственное образовательное учреждение

Высшего образования

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых» (ВлГУ)

Кафедра «Физика и прикладная математика» (ВлГУ)

Курсовая работа

По дисциплине «Прикладное программирование»

«Программный модуль Прокат автомобилей»

Выполнил:

Ст. ПКсп-114

Приняла: Ст. преподаватель  
 Шишкина М. В.

Владимир 2017 г.

**Содержание.**

[**Введение. 2**](#_Toc477129037)

[**Глава1. Постановка задачи 3**](#_Toc477129038)

[**Глава 2. Выбор системы программирования 4**](#_Toc477129039)

[**Глава 3. Описание процесса проектирования приложения 5**](#_Toc477129040)

[**3.1. Взаимосвязь программных модулей 5**](#_Toc477129041)

[**3.1.1. Функция добавления нового авто. 6**](#_Toc477129042)

[**3.1.2. Функция меняющие параметры в списке, если авто взяли в прокат. 6**](#_Toc477129043)

[**3.1.3. Функция меняющие параметры в списке, если автомобиль вернули. 7**](#_Toc477129044)

[**3.1.4. Функция поиска на наличие авто. 8**](#_Toc477129045)

[**3.2. Схема работы программы 8**](#_Toc477129046)

[**3.3. Проектирование интерфейса. 9**](#_Toc477129047)

[**Глава 4. Инструкция для пользователя по работе с приложением 11**](#_Toc477129048)

[**Заключение. 12**](#_Toc477129049)

[**Список использованных информационных источников: 13**](#_Toc477129050)

[**Приложение 1 14**](#_Toc477129051)

# **Введение.**

В наш бурный век, когда все сферы жизни постоянно усовершенствуются и развиваются, трудно найти дом, офис или предприятие, в которых не было бы компьютера. Трудно найти и человека, даже среди детей дошкольного возраста, который не имел бы самых простых навыков работы с этой умной машиной. То, что еще несколько десятков лет назад казалось чем-то непонятным, неизвестным, сложным и почти фантастическим, теперь стало обычным и доступным даже ребенку. Многие уже не представляют своей работы, учебы и повседневной жизни без компьютера. Одна из самых утомляющих работ, которую может облегчить компьютер – ручное заполнение тысяч одинаковых форм, содержащих, к примеру, информацию о студентах. В большинстве государственных и частных учреждений, например, в библиотеке учет книг ведется с помощью ПО. В данной работе я рассмотрю все этапы создания программного модуля “Прокат автомобилей”.

# **Глава1. Постановка задачи**

В теме моей курсовой работы требуется создать приложение “Прокат автомобилей”, которое бы использовалось в компаниях по прокату автомобилей и облегчало учет автомобилей. Приложение должно иметь возможность добавлять новые автомобили, чтобы пополнять автопарк, следить у кого сейчас находится автомобиль и когда он был взят в прокат.

# 

# **Глава 2. Выбор системы программирования**

При выборе системы программирования я выбрал C#. Потому что, он поддерживает практически все современные концепции программирования: [объектно-ориентированное программирование](http://codingcraft.ru/oop.php), [обобщенное программирование (универсальные типы)](http://codingcraft.ru/c_sharp_coding/generics.php), [событийную модель](http://codingcraft.ru/c_sharp_coding/delegates.php), [исключения](http://codingcraft.ru/c_sharp_coding/exceptions.php) и многое другое. Одновременно он лишен многих недостатков, на которые указывают эксперты, анализирую процесс программирования на других языках. К таким недостаткам относят и сложность синтаксиса, и недостаточную гибкость и даже иногда чрезмерную свободу выбора.

C# для меня является самым удобным объектно-ориентированным языком программирования.

# **Глава 3. Описание процесса проектирования приложения**

# **3.1. Взаимосвязь программных модулей**

Итак, первым делом необходимо определиться с функционалом, которым будет обладать модуль. Во-первых, мы должны иметь возможность добавлять автомобили в автопарк компании, так как он будет пополнятся. Во-вторых, мы должны, иметь доступ к этому списку. В-третьих, в этом списке должны быть параметры, указывающие на наличие определенной автомобиле в наличии и возможность этот параметр редактировать. Информация о автомобиле содержит: Марку авто, Модель авто, Наличие в автопарке, если на руках, то у кого и дату выдачи.

Форма была разделена на 3 вкладки: 1 – Список автомобилей и информация о их наличии. 2 - Учет. 3 - Расширение автопарка.

# **3.1.1. Функция добавления нового авто.**

После нажатия кнопки управление передается обработчику события “btnAddAuto\_Click”. Программа создает новый элемент в lvAuto, далее выдается сообщение о том, что данные добавлены, и программа переходит в функцию записи элементов lvAuto в файл, что переписать изменения. Ниже представлен фрагмент кода:

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

{

ListViewItem ListItem = new ListViewItem(tbAddModel.Text);

ListItem.SubItems.Add(tbAddMarka.Text);

ListItem.SubItems.Add("В наличии");

ListItem.SubItems.Add("Находится в автопарке");

ListItem.SubItems.Add("Находится в автопарке");

ListItem.SubItems.Add(cbAutoPrice.Text);

lvAuto.Items.Add(ListItem);

MessageBox.Show("Автомобиль добавлен!", "Автопрокат");

save();

tbAddMarka.Text = "Марка";

tbAddModel.Text = "Модель";

cbAutoPrice.Text = "Стоимость проката";

# **3.1.2. Функция меняющие параметры в списке, если авто взяли в прокат.**

После нажатия кнопки управление передается обработчику события “btnOutAuto\_Click”. Программа проверят написано ли что-то в tbOutModel.Text и tbOutMarka.Text, если нет, то программа выходит из этого обработчика события. Далее осуществляется поиск по элементам, когда программа находит авто, требуемую пользователем, она заменит значения о наличии авто. Потом программа переходит в функцию записи элементов lvAuto в файл, что переписать изменения. Ниже представлен фрагмент кода:

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

{

if (tbOutModel.Text == string.Empty)

return;

if (tbOutMarka.Text == string.Empty)

return;

string date = mtbDateOut.Text;

foreach (ListViewItem item in this.lvAuto.Items)

{

if ((item.SubItems[0].Text == tbOutModel.Text) && (item.SubItems[1].Text == tbOutMarka.Text))

{

item.SubItems[2].Text = "В прокате";

item.SubItems[3].Text = date;

item.SubItems[4].Text = tbFIO.Text;

}

}

save();

tbFIO.Text = "ФИО арендатора";

mtbDateOut.Text = "";

tbOutModel.Text = "Модель";

tbOutMarka.Text = "Марка"; }

# **3.1.3. Функция меняющие параметры в списке, если автомобиль вернули.**

После нажатия кнопки управление передается обработчику события “btnBackAuto\_Click”. Программа проверят написано ли что-то в tbModelBack.Text и tbMarkaBack.Text, если нет, то программа выходит из этого обработчика события. Далее осуществляется поиск по элементам, когда программа находит авто, требуемую пользователем, она заменит значения о наличии авто. Потом программа переходит в функцию записи элементов lvAuto в файл, что переписать изменения. Ниже представлен фрагмент кода:

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

{

{

if (tbModelBack.Text == string.Empty)

return;

if (tbMarkaBack.Text == string.Empty)

return;

foreach (ListViewItem item in this.lvAuto.Items)

{

if ((item.SubItems[0].Text == tbModelBack.Text) && (item.SubItems[1].Text == tbMarkaBack.Text))

{

item.SubItems[2].Text = "В наличии";

item.SubItems[3].Text = "Находится в автопарке";

item.SubItems[4].Text = "Находится в автопарке";

}

}

}

save();

tbModelBack.Text = "Модель";

tbMarkaBack.Text = "Марка"; }

# **3.1.4. Функция поиска на наличие авто.**

После нажатия кнопки управление передается обработчику события “btnSeach\_Click”. Программа инициализирует логическую переменную и придает ей значение false. Создаются строковые переменные m и l в них заносятся значения из tbModelSeach.Text и tbMarkaSeach.Text соответственно. Далее осуществляется перебор всех элементов ListViewItem на поиск нужной книги и значения item.SubItems[2].Text == “В наличии”, если авто отвечает критериям поиска логическая переменная принимает значение true. В зависимости от значения лог.переменной выводится сообщение о наличии авто в автопарке компании. Ниже представлен фрагмент кода:

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

{

bool seach = false;

string l = tbMarkaSeach.Text, m = tbModelSeach.Text;

foreach (ListViewItem item in this.lvAuto.Items)

{

if ((item.SubItems[0].Text == m) && (item.SubItems[1].Text == l) && (item.SubItems[2].Text == "В наличии"))

{ seach = true; }

}

if (seach == true)

MessageBox.Show("Модель - " + m + ", Марки - " + l + "; В наличии ", "Прокат автомобилей");

else

MessageBox.Show("Модель - " + m + ", Марки - " + l + "; Взята в прокат ", "Прокат автомобилей");

tbMarkaSeach.Text = "Марка";

tbModelSeach.Text = "Модель";

# **3.2. Схема работы программы**

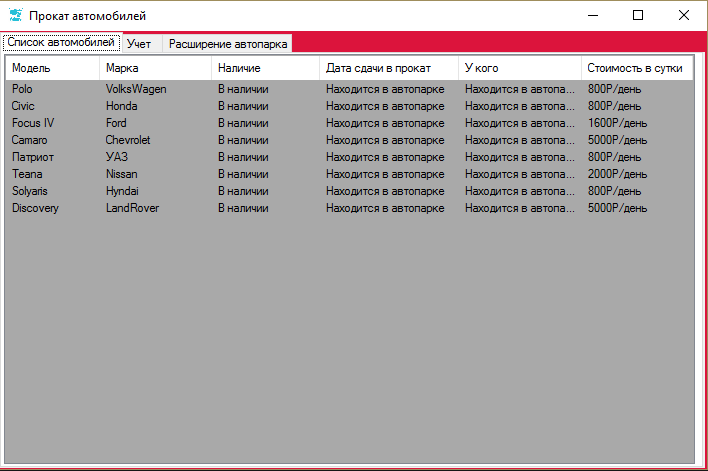
Прокат автомобилей

Арендатор

# **3.3. Проектирование интерфейса.**

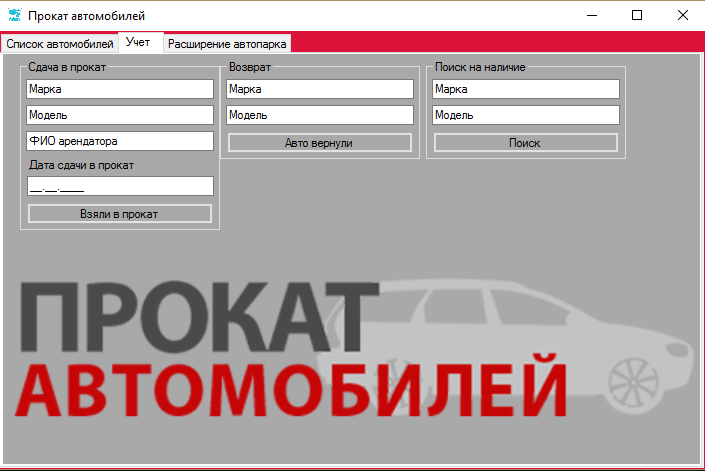
Проектирование интерфейса – не самая простая задач. Интерфейс должен быть интуитивен и понятен не только программисту, создавшему его, но и конечному пользователю, для которого собственно проект и разрабатывается. В ходе разработки интерфейс несколько раз менялся, пока не принял, на мой взгляд, достаточно приятный и понятный для стороннего пользователя вид:

На представленном ниже окне формы пользователю доступен список автомобилей и информация о их наличии.



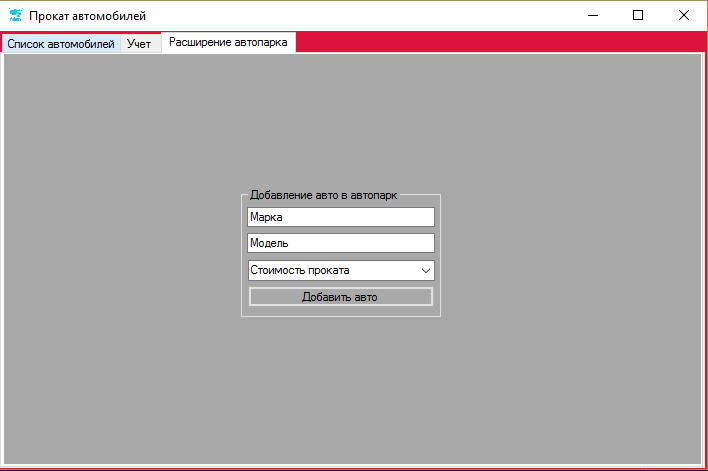
*Рис. 1. Интерфейс программы. Первая вкладка.*

На представленном ниже окне формы располагается окно учета автомобилей, где пользователь может оформить сдачу автомобиля в прокат, его возврат и узнать о наличие автомобиля в автопарке.



*Рис. 2. Интерфейс программы. Вторая вкладка.*

На представленном ниже окне формы производится добавления автомобилей в автопарк. Пользователь вводит информацию о новом автомобиле.



*Рис. 3. Интерфейс программы. Третья вкладка.*

# **Глава 4. Инструкция для пользователя по работе с приложением**

После запуска программы, вы создаете список автомобилей, которые находятся в автопарке фирмы, чтобы в дальнейшем с ним работать, перейдя на первой вкладке можно ознакомиться с этим списком, на второй вкладке ведется учет автомобилей, если авто взяли в прокат, вы вводите информацию о авто, ФИО арендатора и дату выдачи машины в пользование, вся эта информация заменяется в списке автомобилей. Если авто вернули, вам достаточно ввести модель и марку автомобиля и нажать кнопку “Авто вернули”, авто снимется с учета. Так же можно узнать находится ли машина в автопарке, для этого надо воспользоваться поиском.

# **Заключение.**

В ходе выполнения этой работы был разработан программный модуль, предоставляющий полный функционал картотеки для учёта литературы в библиотеки, который полностью соответствует поставленной задаче. Я научился работать с файлами и объектом listView. Также были закреплены мои знания объектно-ориентированного языка С#.

# **Список использованных информационных источников:**

1. Язык программирования C#. Классика Computer Science. 4-е изд.

Авторы: А. Хейлсберг, М.Торгерсен, С.Вилтамут, П. Голд

1. [www.habrahabr.ru/hub/csharp/](http://www.habrahabr.ru/hub/csharp/) - Хабрахабр, информация по языку C#
2. [https://msdn.microsoft.com/](https://msdn.microsoft.com/ru-RU/library/mt472912(v=vs.110).aspx) - Microsoft Developer Network
3. "Полный справочник по C#" Автор: Герберт Шилдт
4. "Советы программистам" Автор: А. Климов
5. Язык программирования C# 5.0 и платформа .NET 4.5 Автор: Эндрю Троелсен..

# **Приложение 1**

Код программы:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Diagnostics;

using System.IO;

using System.Xml.Linq;

using System.Globalization;

namespace AutoProkat

{

public partial class Form1 : Form

{

string UsersDirectory = Environment.CurrentDirectory;

public Form1()

{

InitializeComponent();

parseItems();

}

bool isWrite = false;

private void save()

{

isWrite = true;

StringBuilder listViewContent = new StringBuilder();

foreach (ListViewItem item in this.lvAuto.Items)

{

foreach (ListViewItem.ListViewSubItem subItem in item.SubItems)

{

listViewContent.Append(subItem.Text);

listViewContent.Append('|');

}

listViewContent.Append(Environment.NewLine);

}

using (TextWriter tw = new StreamWriter(UsersDirectory + "\\prokat.txt"))

{

tw.WriteLine(listViewContent.ToString());

tw.Close();

}

}

private void parseItems()

{

lvAuto.Items.Clear();

using (StreamReader sReader = new StreamReader("prokat.txt"))

{

string line;

while ((line = sReader.ReadLine()) != null)

{

string[] values = line.Split('|');

lvAuto.Items.Add(new ListViewItem(values));

}

}

}

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

{

Close();

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

if (isWrite)

{

StreamReader sReader = new StreamReader("prokat.txt");

string str = sReader.ReadToEnd();

sReader.Close();

TextWriter tw = new StreamWriter(UsersDirectory + "\\prokat.txt");

tw.Write(str.Remove(str.Length - 4, 4));

tw.Close();

}

}

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

{

ListViewItem ListItem = new ListViewItem(tbAddModel.Text);

ListItem.SubItems.Add(tbAddMarka.Text);

ListItem.SubItems.Add("В наличии");

ListItem.SubItems.Add("Находится в автопарке");

ListItem.SubItems.Add("Находится в автопарке");

ListItem.SubItems.Add(cbAutoPrice.Text);

lvAuto.Items.Add(ListItem);

MessageBox.Show("Автомобиль добавлен!", "Автопрокат");

save();

tbAddMarka.Text = "Марка";

tbAddModel.Text = "Модель";

cbAutoPrice.Text = "Стоимость проката";

}

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

{

bool seach = false;

string l = tbMarkaSeach.Text, m = tbModelSeach.Text;

foreach (ListViewItem item in this.lvAuto.Items)

{

if ((item.SubItems[0].Text == m) && (item.SubItems[1].Text == l) && (item.SubItems[2].Text == "В наличии"))

{ seach = true; }

}

if (seach == true)

MessageBox.Show("Модель - " + m + ", Марки - " + l + "; В наличии ", "Прокат автомобилей");

else

MessageBox.Show("Модель - " + m + ", Марки - " + l + "; Взята в прокат ", "Прокат автомобилей");

tbMarkaSeach.Text = "Марка";

tbModelSeach.Text = "Модель";

}

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

{

if (tbOutModel.Text == string.Empty)

return;

if (tbOutMarka.Text == string.Empty)

return;

string date = mtbDateOut.Text;

foreach (ListViewItem item in this.lvAuto.Items)

{

if ((item.SubItems[0].Text == tbOutModel.Text) && (item.SubItems[1].Text == tbOutMarka.Text))

{

item.SubItems[2].Text = "В прокате";

item.SubItems[3].Text = date;

item.SubItems[4].Text = tbFIO.Text;

}

}

save();

tbFIO.Text = "ФИО арендатора";

mtbDateOut.Text = "";

tbOutModel.Text = "Модель";

tbOutMarka.Text = "Марка";

}

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

{

{

if (tbModelBack.Text == string.Empty)

return;

if (tbMarkaBack.Text == string.Empty)

return;

foreach (ListViewItem item in this.lvAuto.Items)

{

if ((item.SubItems[0].Text == tbModelBack.Text) && (item.SubItems[1].Text == tbMarkaBack.Text))

{

item.SubItems[2].Text = "В наличии";

item.SubItems[3].Text = "Находится в автопарке";

item.SubItems[4].Text = "Находится в автопарке";

}

}

}

save();

tbModelBack.Text = "Модель";

tbMarkaBack.Text = "Марка";

}

private void tbAddMarka\_MouseDown(object sender, MouseEventArgs e)

{

tbAddMarka.Text = "";

}

private void tbAddModel\_MouseDown(object sender, MouseEventArgs e)

{

tbAddModel.Text = "";

}

private void tbOutMarka\_MouseDown(object sender, MouseEventArgs e)

{

tbOutMarka.Text = "";

}

private void tbOutModel\_MouseDown(object sender, MouseEventArgs e)

{

tbOutModel.Text = "";

}

private void tbFIO\_MouseDown(object sender, MouseEventArgs e)

{

tbFIO.Text = "";

}

private void tbMarkaBack\_MouseDown(object sender, MouseEventArgs e)

{

tbMarkaBack.Text = "";

}

private void tbModelBack\_MouseDown(object sender, MouseEventArgs e)

{

tbModelBack.Text = "";

}

private void tbModelSeach\_MouseDown(object sender, MouseEventArgs e)

{

tbModelBack.Text = "";

}

private void tbMarkaSeach\_MouseDown(object sender, MouseEventArgs e)

{

tbMarkaSeach.Text = "";

}

}

}