

Отчёт №3

Тема: «Модульное тестирование методов настольного приложения»

Задание: разработать каркас приложения и реализовать функционал, где применить ранее созданный метод согласно инд. варианту. Разработать тест-кейс(ы) и провести тестирование.

1 Спецификация метода

Таблица 1 – Спецификации требований к методу

Наименование проекта	WpfAppAddCars
Наименование класса	PageAddCars
Наименование тестируемого метода	btnSave_Click()
Входные параметры	tbBrand.Text, tbModel.Text, tbPrice.Text, tbAmount.Text
Выходные параметры	Сообщения о успешном\неуспешном добавлении данных

В таблице1 представлена спецификация тестируемого метода.

2 Тестирование

					ККЭП 09.02.07 №3 №2 642_06		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.		Дядюшкин А. П.			Модульное тестирование методов настольного приложения		
Провер.		Головки Р.А.					
					Гр. 642-КД9-4ИСП		
					Лит.	Лист	Листов
					КП	1	6

```

private void btnSave_Click(object sender, RoutedEventArgs e)
{
    Cars _currentCars = new Cars();
    StringBuilder errors = new StringBuilder();

    if (tbBrand.Text.Length == 0)
        errors.AppendLine("Укажите марку автомобиля");
    if (!(tbBrand.Text.Intersect("#$%^&_").Count() == 0))
        errors.AppendLine("В названии марки автомобиля не должно быть специальных символов");
    if (tbModel.Text.Length == 0)
        errors.AppendLine("Укажите модель автомобиля");

    try
    {
        _currentCars.Price = Convert.ToDecimal(tbPrice.Text);
    }
    catch
    {
        if (tbPrice.Text.Length == 0)
            errors.AppendLine("Укажите цену автомобиля");
    }

    try
    {
        _currentCars.Amount = Convert.ToInt32(tbAmount.Text);
    }
    catch
    {
        if (tbAmount.Text.Length == 0)
            errors.AppendLine("Укажите количество автомобилей");
    }

    _currentCars.Brand = tbBrand.Text;
    _currentCars.Model = tbModel.Text;

    if (errors.Length > 0)
    {
        MessageBox.Show(errors.ToString());
        return;
    }
    else
    {
        if (_currentCars.Id == 0)
            bdAvtoLider2Entities.GetContext().Cars.Add(_currentCars);
        bdAvtoLider2Entities.GetContext().SaveChanges();
        MessageBox.Show("Информация сохранена!");
        ClassManager.BaseFrame.GoBack();
    }
}

```

Рисунок 1 – Тестируемый метод

На рисунке 1 представлен метод, который будет тестироваться в рамках настольного приложения. “tbBrand.text”, “tbModel.text”, “tbPrice.text”, “tbAmount.text” это параметры, которые принимают значения, вводимые в поля ввода пользователем. Данные параметры проходят проверки, и, если они были успешно пройдены, данные добавляются в базу данных и выводится соответствующее сообщение, иначе выведется ошибка с уточняющей информацией. Метод представлен в обработчике события нажатия кнопки “Save”.

Таблица 2 – Тест кейс для тестирования данных

Test Case #	TC_UT_1
Приоритет теста	Средний
Название тестирования/Имя	Тестирование метода проверки и добавления данных в базу данных

Продолжение таблицы 2

Резюме испытания	<p>1) Нужно достигнуть успешного добавления данных в базу при вводе корректных данных, и вывода соответствующего сообщения;</p> <p>2) Нужно достигнуть вывода ошибки при вводе некорректных данных (оставить незаполненные поля ввода);</p> <p>3) Нужно достигнуть вывода ошибки при вводе некорректных данных (в названии марки автомобиля прописать специальный символ("#\$%^&_"))</p>
Шаги тестирования	<p>1) Скачать с репозитория https://github.com/Alexandr33333/Dyadyushkin642_Practic проект;</p> <p>2) Запустить программу;</p> <p>3) Нажать «Добавить», ввести данные и нажать сохранить</p>
Данные тестирования	<p>1) Проверка успешного добавления:</p> <ul style="list-style-type: none"> • Марка: "Mercedes"; • Модель: "CLS"; • Цена: 7000000; • Количество: 6 <p>2) Проверка на незаполненные поля: оставить поля ввода незаполненными</p> <p>3) Проверка на ввод спец символов:</p> <ul style="list-style-type: none"> • Марка: "Mercedes\$"; • Модель: "CLS"; • Цена: 7000000; • Количество: 6
Ожидаемый результат	<p>1) При вводе корректных данных, будет выведено сообщение об успешном добавлении данных;</p> <p>2) При попытке добавления данных с каким-либо незаполненным полем программа выдаст ошибку с просьбой заполнить конкретное пустое поле ввода;</p> <p>3) При попытке добавления данных с специальным символом в названии марки программа выдаст ошибку с информацией о том, что в названии марки не должно быть спец символов</p>
Фактический результат	<p>1) При вводе корректных данных, было выведено сообщение об успешном добавлении данных;</p> <p>2) При попытке добавления данных с каким-либо незаполненным полем программа выдала ошибку с просьбой заполнить конкретное пустое поле ввода;</p> <p>3) При попытке добавления данных с специальным символом в названии марки программа выдала ошибку с информацией о том, что в названии марки не должно быть спец символов</p>
Предпосылки	Проверить подключение к базе данных
Постусловия	Программа работает без ошибок
Статус (Pass/Fail)	Pass
Комментарии	Не требуются

В таблице 2 представлен тест-кейс для тестирования метода обработчика события нажатия кнопки “Save”.

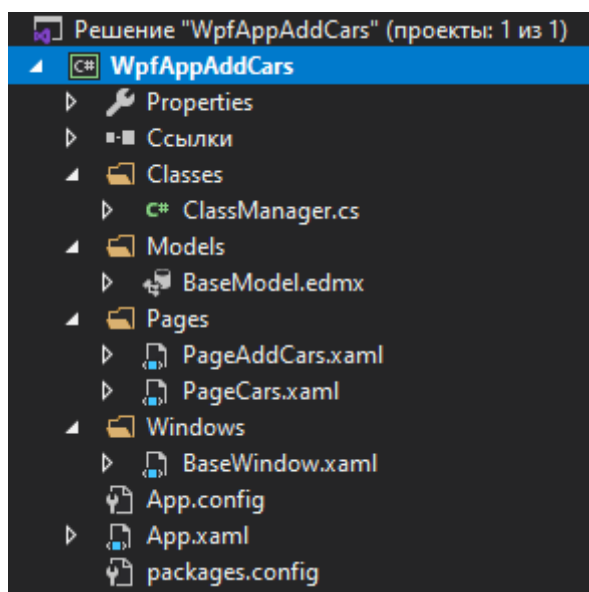


Рисунок 1 – Обзорщик решений

На рисунке 1 отображена структура проекта.

Автосалон АвтоЛидер

Добавление (изменение) данных об автомобилях

Марка

Модель

Цена

Количество

X
 Информация сохранена!
 OK

Сохранить
Отмена

Рисунок 2 – Успешное добавление данных

Автосалон АвтоЛидер

Список автомобилей

Марка	Модель	Цена	Количество
BMW	x5	6 000 000,00 Р	9
Hyundai	Genesis	1 500 000,00 Р	12
Audi	A8	7 500 000,00 Р	6
Lexus	GX	5 800 000,00 Р	9
LADA	XRAY	800 000,00 Р	15
VAZ	Patriot	900 000,00 Р	10
Haval	Jolion	1 189 000,00 Р	10
Lada	Kalina	500 000,00 Р	10
BMW	Exodus	1 500 000,00 Р	8
Mercedes	Benz	3 000 000,00 Р	8
Mercedes	CLS	7 000 000,00 Р	6

Выход
Добавить

Рисунок 3 – Данные добавлены в базу

На рисунках 2 и 3 представлен сценарий успешного добавления данных об автомобиле в базу данных.

Автосалон АвтоЛидер

Добавление (изменение) данных об автомобилях

Марка

Модель

Цена

Количество

×

Укажите марку автомобиля
Укажите модель автомобиля
Укажите цену автомобиля
Укажите количество автомобилей

ОК

Сохранить
Отмена

Рисунок 4 – Неуспешное добавление данных (пустые поля ввода)

На рисунке 4 представлен сценарий неуспешного добавления данных, при котором поля ввода не заполнены.

Автосалон АвтоЛидер

Добавление (изменение) данных об автомобилях

Марка

Mercedes\$

Модель

CLS

Цена

7000000

Количество

6

Сохранить

Отмена

В названии марки автомобиля не должно быть специальных символов

OK

Рисунок 5 – Неуспешное добавление данных (спец символ в названии марки)

На рисунке 5 представлен сценарий неуспешного добавления данных, при котором введен недопустимый спец символ в поле ввода названия марки.