

## Прикладное программирование

### Лабораторная работа №3

#### Оконное приложение и тестирование

**Задача:** Необходимо реализовать оконный интерфейс к программе, разработанной на предыдущих лабораторных работах и написать автоматические тесты

#### Реализация оконного приложения

К моменту 3-ей лабораторной работы у вас уже должно быть:

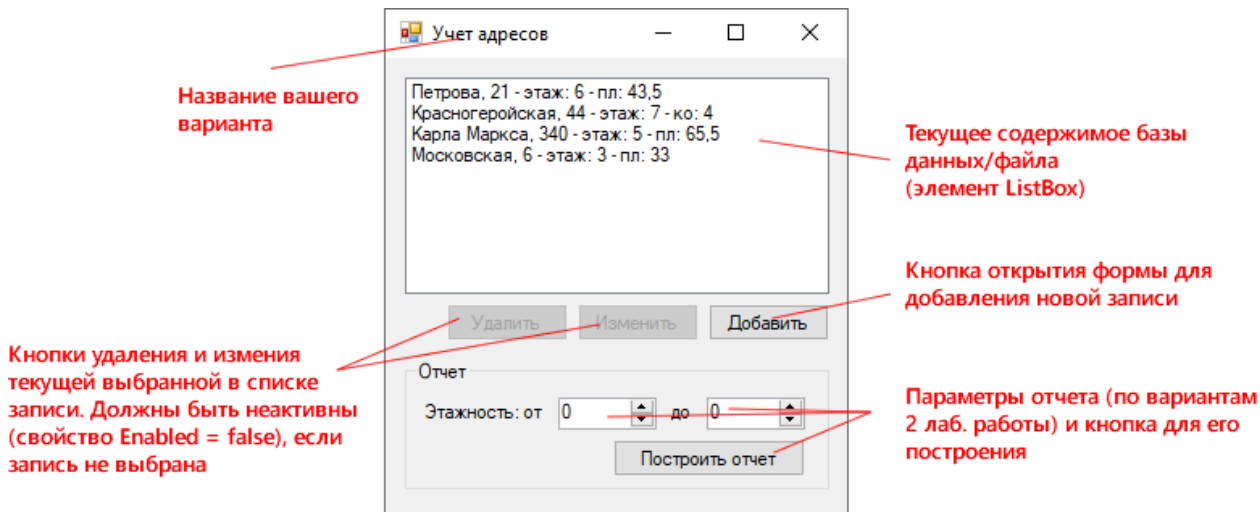
- Класс FileDataSource, который отвечает за хранение внесенных в программу данных в файле
- Классы бизнес-логики, которые отвечают за выполнение основных функций программы без привязки к конкретному пользовательскому интерфейсу

Вы должны скопировать эти модули без изменений и заменить лишь модуль, отвечавший за ввод/вывод в консоль.

В программе у вас будет две формы:

- Основное окно
- Форма добавления/изменения записи

#### Основное окно



- При загрузке программы в ListBox должны подставляться данные из базы через метод получения отсортированного списка из класса бизнес-логики. Для того, чтобы поместить полученный список в ListBox, присвойте его в свойство DataSource.

Для форматирования вывода записей реализуйте стандартный метод ToString в классах записей

- При нажатии кнопки «Добавить» должна отображаться форма добавления/изменения записи с помощью метода [ShowDialog](#) у отображаемой

формы. После успешного добавления перезагрузите содержимое списка (вновь вызовите метод получения отсортированного списка и присвойте в DataSource)

- При нажатии на кнопку «Изменить» также отображается форма добавления/изменения записи, но в нее изначально подставляются данные из изменяемой записи. После изменения перезагрузите содержимое списка
- При нажатии на кнопку «Удалить» должно появиться окно подтверждения (используйте стандартный `MessageBox.Show` с кнопками «Да» и «Нет») и, если пользователь подтвердил удаления, то вызываться метод бизнес-логики для удаления записи. После удаления перезагрузите содержимое списка и сбросьте выделение у `ListBox`.
- При нажатии на кнопку «Построить отчет» должно открыться стандартное окно сохранения файла (`SaveFileDialog`) и далее вызваться метод бизнес-логики построения отчета из второй лаб. работы.

### Форма добавления/изменения записи

- В зависимости от выбранного типа записи должны быть активны (`Enabled = true`) разные специфичные поля
- Кнопка «Отмена» устанавливает у формы `DialogResult = DialogResult.Cancel` и закрывает форму с помощью метода `Close`
- Кнопка «Сохранить» формирует запись из полей формы и вызывает метод сохранения в бизнес-логике. В случае ошибки отображает ее с помощью `MessageBox.Show`. В случае успеха устанавливает у формы `DialogResult = DialogResult.OK` и закрывает форму

### Автоматическое тестирование:

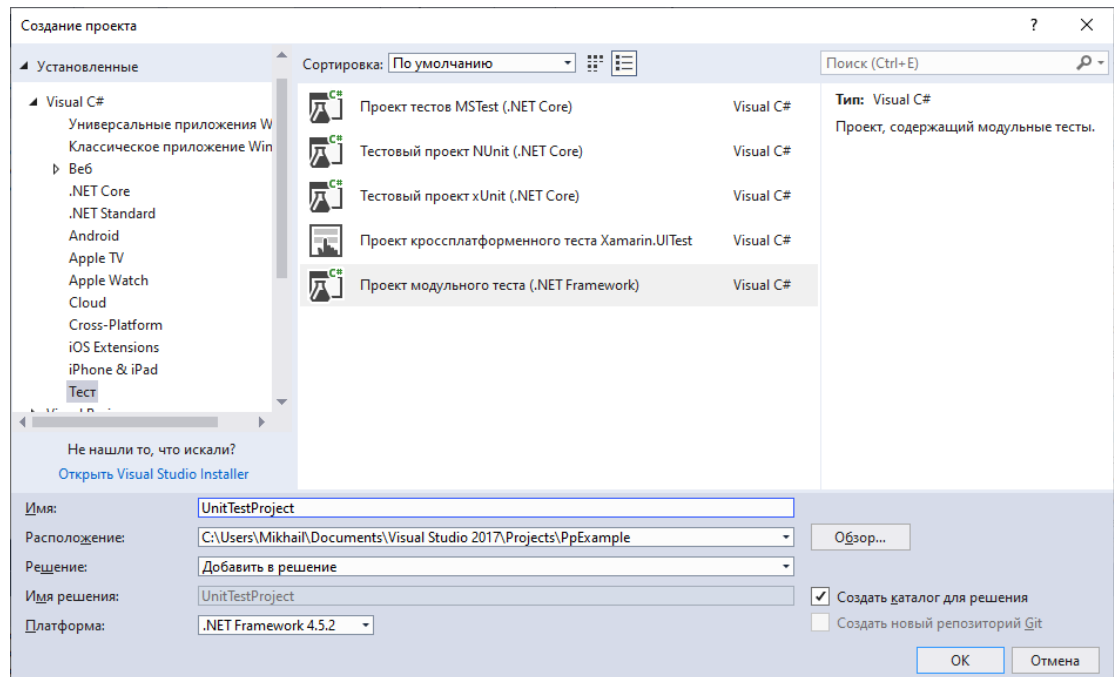
Вам необходимо написать юнит-тесты для проверки правильности работы основных методов класса бизнес-логики:

- Добавление записи
- Изменение записи
- Проверка реакции на ошибочный ввод
- Вывод отсортированного списка

- Удаление записи

Юнит-тесты должны работать с классом `MemoryDataSource`, разработанным в рамках первой лабораторной работы

Для создания юнит-тестов добавьте в решение новый проект типа «Проект модульного теста».



### Контрольные вопросы:

1. Чем разработка консольного приложения отличается от разработки оконного?
2. Что такое событие? Как их обрабатывать?
3. Что такое элемент управления?
4. Что такое свойство элемента управления?
5. Приведите примеры элементов управления и кратко их охарактеризуйте
6. Что такое форма?
7. Какими способами можно отобразить форму?
8. Как осуществляется взаимодействие между формами?
9. Для чего применяется тестирование?
10. Что такое автоматическое тестирование?
11. Что такое модульное (юнит-) тестирование?
12. Что обозначает `[TestClass]` перед классом?
13. Что обозначает `[TestMethod]` перед методом?
14. Как запустить автоматические тесты на выполнение?