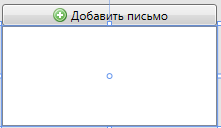
# Домашнее задание

1. Мы реализовали возможность отправлять серию писем в один день при помощи класса **SchedulerClass**. Задание: добавить такую возможность на весь проект.
   1. Убрать параметр **dtSend** из сигнатуры метода **SendEmails** класса **SchedulerClass**;
   2. Сделать возможность добавлять несколько вариантов времени отправления писем на одну дату и их текстов.
      1. Вместо поля **tbTimePicker** (для тех, кто смог заменить текстовое поле на элемент **TimePicker**) добавить элемент **ListView** и сверху кнопку «Добавить письмо». По клику на нее будет добавляться новый элемент в **ListView**. Выглядеть это должно так:



* + 1. В качестве элемента в **ListView** добавлять самописный контрол. Для него нужно создать проект **ListViewItem Scheduler**. Выглядеть контрол должен так:



Посередине — поле, куда записываются даты. Кнопка «-» удаляет элемент из списка. Кнопка с иконкой-карандашом вызывает окно с **RichTextBox**, в который можно записать текст письма.

* 1. При клике на кнопку «Отправить запланировано» забирать информацию из **ListView**, передавать ее в класс **SchedulerClass** в поле **Dictionary<DateTime, string>** и обрабатывать так, как прописано в уроке.



1. Добавить **UnitTest** для метода **getPassword** проекта **CodePasswordDLL**:
   1. Методы с тестами добавить в проект **CodePasswordDLL.Test** класса **CodePasswordTest** по аналогии с тестами для метода **getCodPassword**.
   2. Добавить в класс **CodePasswordTest** методы **TestInitialize** и **TestCleanup**.
   3. Расширить метод **Assert.AreEqual** параметром с сообщением для **output**.
   4. Добавить **Debug.WriteLine** внутри тестовых методов и посмотреть, как записываются текстовые сообщения.
   5. Добавить тестовые методы для этого же класса с использованием методов **StringAssert**.
2. Добавить тестовые методы для класса **SchedulerClassTest** с использованием тестовой коллекции **sc.DatesEmailTexts** и методов класса **CollectionAssert**.