

Приложение Телефонная книга.

Тема:

разработка приложения под Windows в технологии ООП.

Цель:

Объектно-ориентированный анализ, проектирование и реализация приложения под Windows «Телефонная книга».

В процессе выполнения работы студенты изучают:

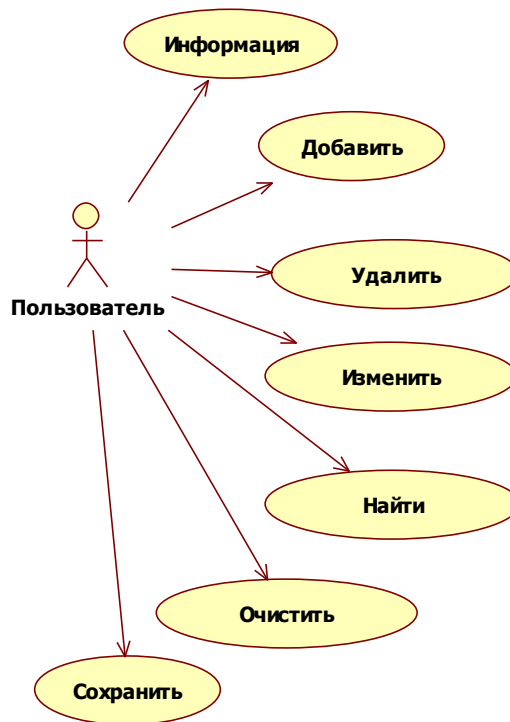
- этапы разработки приложений в технологии ООП;
- элементы технологии визуального программирования;
- ассоциативный контейнер `stl – multimap`;
- потоковые классы для работы с файлами на диске;
- диаграммы языка UML для документирования разработки.

Функциональные требования к приложению.

Приложение должно обеспечивать пользователю:

- ввод, редактирование и сохранение имён абонентов городской телефонной сети и номеров их телефонов
- записи должны храниться и отображаться в отсортированном по именам порядке;
- поиск по имени;
- удаление записи;
- очистку книги.

Функциональные требования представлены диаграммой прецедентов (use-case диаграммой) расположенной ниже.



Сценарий для прецедента «Добавить»

Основной поток событий

- 1) Пользователь вводит имя абонента.
- 2) Пользователь вводит номер абонента.
- 3) Пользователь нажимает команду Добавить.
- 4) Приложение добавляет абонента в книгу и отображает на экране запись по нему в списке.

Альтернативный поток событий 1. Одно из полей записи не заполнено.

- 3.1. Пользователь получает окно с сообщением.
- 3.2. Приложение переходит в режим Ввод и редактирования незаполненного поля.

Диаграмма классов модели объектно-ориентированного анализа.

Проанализировав прецеденты можно выделить следующий класс для реализации функциональности приложения. Он представлен на диаграмме классов ниже.

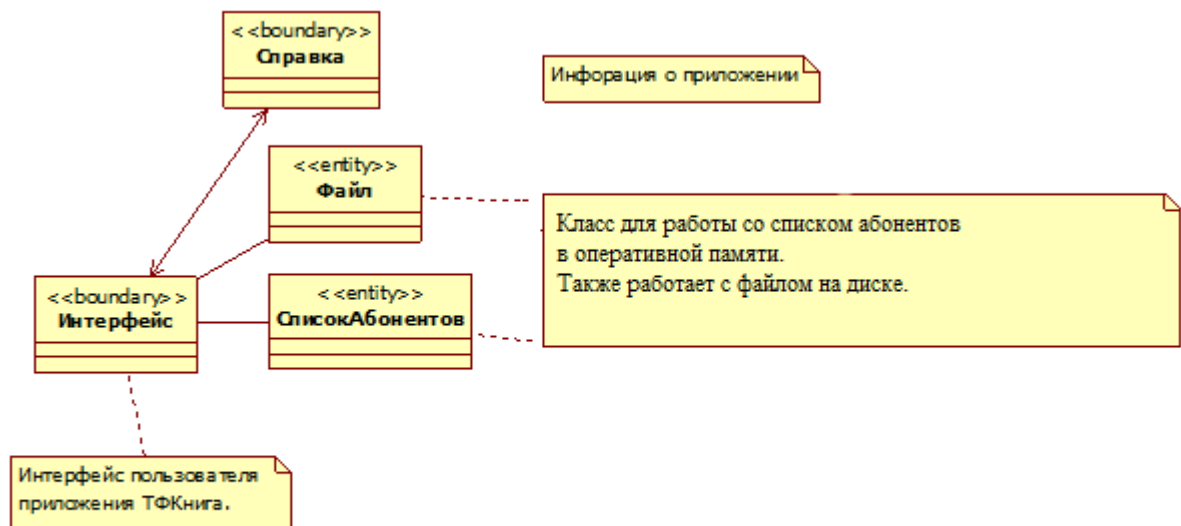
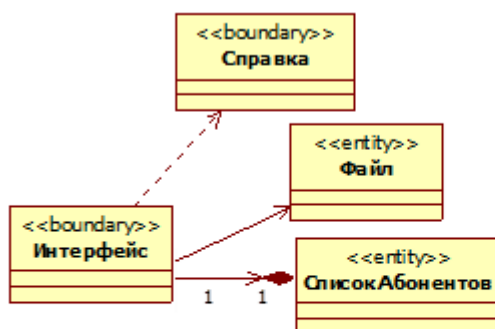


Диаграмма классов проекта

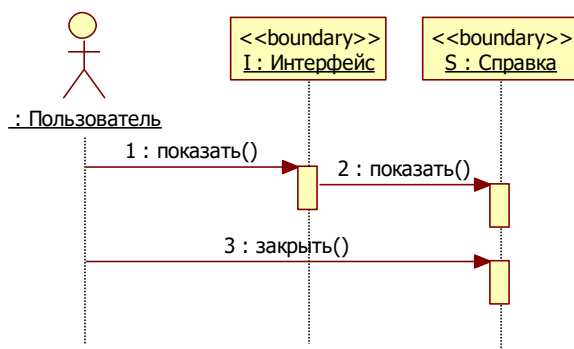
Проанализировав сообщения, которыми обмениваются классы в процессе выполнения прецедентов можно построить следующую диаграмму классов проекта.



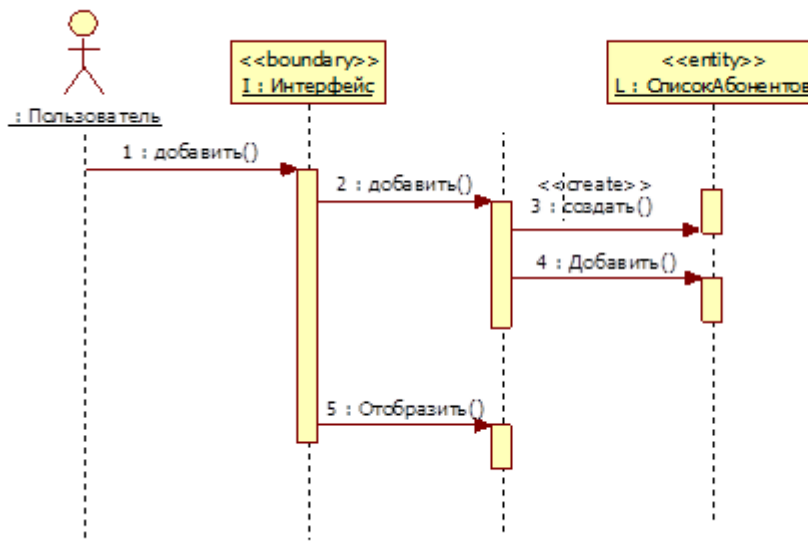
Из диаграммы классов видно, что объект класса «Интерфейс» вызывает методы класса «Список абонентов», «Справка», «Файл».

Обмен сообщениями между объектами. Диаграмма последовательностей.

Давайте спроектируем обмен сообщениями между объектами в процессе выполнения прецедента «Информация». На диаграмме последовательностей приведённой ниже приведёна последовательность сообщений между объектами в процессе реализации прецедента «Информация».



На диаграмме последовательностей приведённой ниже приведёна последовательность сообщений между объектами в процессе реализации прецедента «Добавить».



Интерфейс приложения может выглядеть так:

The screenshot shows a window titled 'Телефонная книга'. It contains a list of contacts with columns for name and phone number. Below the list are buttons for 'Очистить', 'Сохранить', 'Удалить', and 'Изменить'. At the bottom, there is a 'Создать' button and a form with fields for 'ФИО' (Name) and 'Номер' (Number), with a 'Добавить' button next to them. A 'Найти' button is also present at the bottom right.

Имя	Номер
Саша	т-ф. 2333555
Маша	т-ф. 2555777
Коля	т-ф. 3777888

Порядок выполнения разработки

- Реализуйте по заданной спецификации класс «Список абонентов» и протестируйте его в режиме консольного приложения. Для этого создайте новое консольное приложение и добавьте в него модуль и назовите его UAbonentList. При разработке класса используйте ассоциативный контейнер multimap.
- Реализуйте по заданной спецификации класс «Интерфейс» и протестируйте его.
- Объедините исходные модули для создания итогового приложения. Протестируйте его.