

Записная книжка NoteApp

Назначение приложения

Пользовательское приложение NoteApp, предназначено для ведения персональных записей и заметок. Приложение должно:

- 1) Обеспечивать стабильную работу приложения при порядке 200 заметок.
- 2) Обеспечивать категоризацию заметок, навигацию по созданным заметкам.
- 3) Предоставить инструменты для просмотра и редактирования заметок.
- 4) Сохранять и восстанавливать заметки между сессиями приложения.
- 5) Выполнять промежуточные сохранения заметок на машине пользователя на случай аварийного завершения программы, отключения компьютера и т.д. – для защиты от потери данных.

Приложение-референс: десктоп-версия программы Evernote

Пользовательский интерфейс

(лабораторная работа «Разработка пользовательского интерфейса»)

После запуска приложения перед пользователем появляется главное окно (рис. 1). Двухколоночная верстка главного окна содержит список всех созданных заметок в левой панели и отображает текущую выбранную заметку в правой панели. В списке заметок показаны названия заметок, в один момент времени может быть выбрана только одна заметка (далее – текущая заметка).

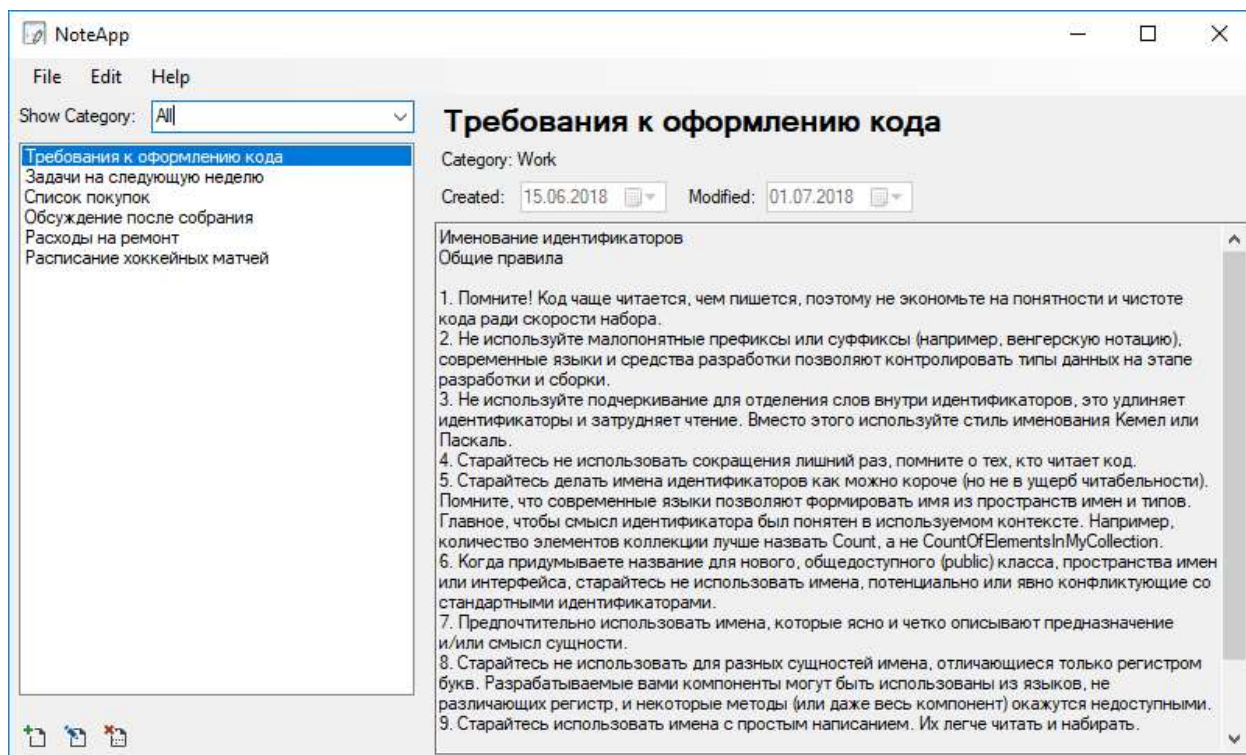


Рисунок 1 – Макет главного окна приложения NoteApp

На панели со списком заметок внизу располагаются три кнопки в виде пиктограмм: Add Note («Создать новую заметку»), Edit Note («Редактировать текущую заметку»), Remove Note («Удалить текущую заметку»).

При выборе заметки в списке, выбранная заметка отображается в правой панели. Главное окно не позволяет редактировать содержимое заметки – только просмотр.

При нажатии на кнопку Add Note и Edit Note появляется окно создания/редактирования заметки в диалоговом режиме (рис. 2). Для новой заметки окно изначально не заполнено (Установлены лишь название заметки по умолчанию, дата создания и дата редактирования). Для редактирования уже существующей заметки все поля должны быть предзаполнены данными текущей заметки.

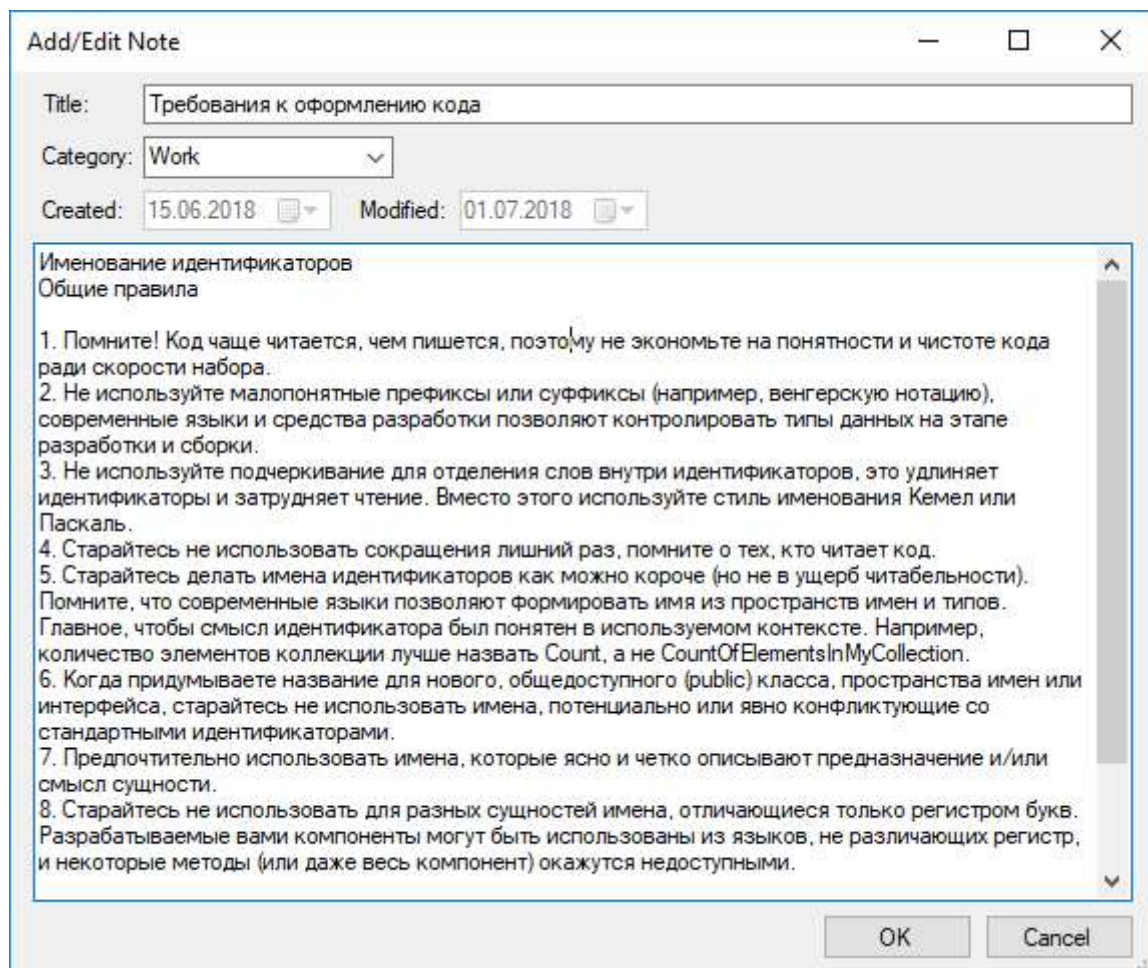


Рисунок 2 – Макет окна редактирования заметки в приложении NoteApp

При нажатии на кнопку ОК окно создания заметки закрывается, в список заметок главного окна добавляется новая заметка. При редактировании текущей заметки, нажатие на кнопку ОК должно обновить название заметки в списке заметок (если название текущей заметки было изменено), и обновить отображаемую заметку в правой панели приложения. При нажатии кнопки Cancel создание/редактирование заметки отменяется (новая заметка не добавляется, исходная заметка остается без изменений).

В случае ввода пользователем некорректных данных (нарушение допустимой длины названия заметки), данная ситуация должна быть обработана соответствующим образом.

При нажатии на кнопку Remove Note главного окна текущая запись удаляется. Перед удалением должно появиться окно с запросом на разрешение записи: «Do you really want to remove this note: <Название текущей записи>». При нажатии на кнопку ОК происходит удаление, при нажатии на кнопку Cancel удаление отменяется.

Меню главного окна содержит следующие пункты:

- *File*:
 - *Exit* (Выйти из приложения – Alt+F4)
- *Edit*:
 - *Add Note* (Создать новую заметку)
 - *Edit Note* (Редактировать текущую заметку)
 - *Remove Note* (Удалить текущую заметку)
- *Help*
 - *About* (Вызвать окно «О программе» - F1) (см. рис. 3)

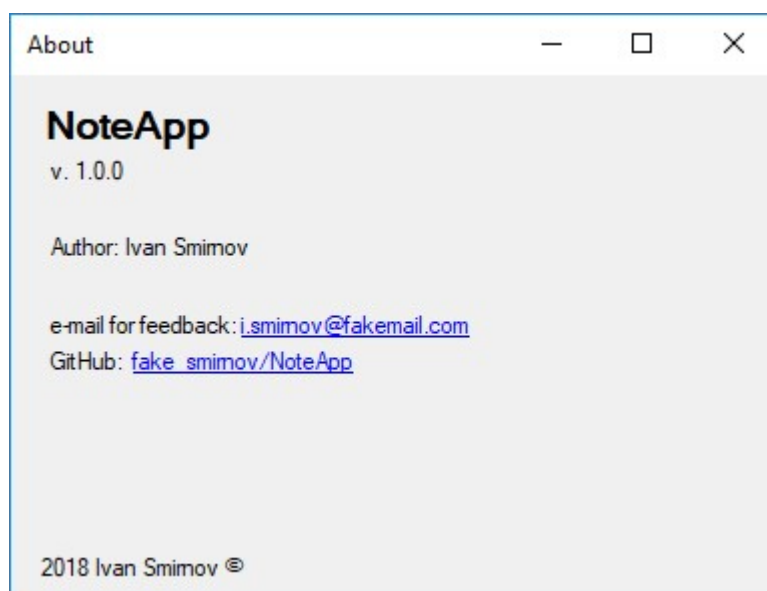


Рисунок 3 – Макет окна «About» приложения NoteApp

Таким образом, в программе должно быть реализовано три окна:

- Главное окно.
- Окно создания/редактирования заметки.
- Окно «About».

Верстка главного окна и окна создания/редактирования заметки должна быть адаптивной. Окно «About» имеет фиксированный размер.

Создание дополнительных элементов управления для уменьшения дублирования кода остается на усмотрение разработчика или руководителя.

Загрузка заметок осуществляется **при запуске программы** до вывода главного окна пользователю, сохранение заметок в файл должно выполняться в случаях: а) создания новой заметки; б) удаления заметки; в) закрытии приложения.

Допущения первой версии программы:

**В первой версии программы заметки в списке отображаются в порядке их создания или ином произвольном порядке.*

**В первой версии программы при запуске приложения текущей записью может быть первая запись в списке.*

**В первой версии программы необязательно реализовывать логику выпадающего списка с выбором категории заметок.*

**Все описанные допущения будут устранены в лабораторной работе «Расширение функциональности и релиз программы» (см. п. «Дополнительная функциональность»).*

Требования к бизнес-логике

(лабораторная работа «Разработка бизнес-логики приложения»)

Необходимо реализовать следующие типы данных:

- Перечисление «Категория заметки», содержащее значения «Работа», «Дом», «Здоровье и Спорт», «Люди», «Документы», «Финансы», «Разное».
- Класс «Заметка» с полями «Название», «Категория заметки», «Текст заметки», «Время создания», «Время последнего изменения». Название, категория и текст заметки доступны для изменений. Время создания инициализируется один раз при создании объекта «Заметка» и больше не модифицируется, доступна для чтения. Время последнего изменения меняется автоматически при изменении названия, категории или текста заметки. Название ограничено 50 символами. Название по умолчанию «Без названия». Допустимы заметки с одинаковыми названиями. Реализует интерфейс `ICloneable`.
- Класс «Проект». Содержит список(или словарь) всех заметок, созданных в приложении.
- Класс «Менеджер проекта». Реализует метод для сохранения объекта «Проект» в файл и метод загрузки проекта из файла. Сохранение и загрузка осуществляются в один и тот же файл «...\My Documents\NoteApp.notes», имя которого задано закрытой константой внутри класса. Формат данных – json, библиотека сериализации (преобразования данных в json-формат) – `Newtonsoft.JSON.NET`.

Дополнительная функциональность

(лабораторная работа «Расширение функциональности и релиз программы»)

- В класс «Проект» добавить метод, который возвращает список заметок, отсортированный по дате изменения. Добавить перегруженный метод сортировки заметок, принимающий на вход категорию заметок – метод возвращает отсортированный по дате изменения список заметок, принадлежащих только этой категории. В случае отсутствия заметок данной категории, метод возвращает пустой список.
- В левой панели главного окна обеспечить сортировку списка заметок по дате изменения. Таким образом, последние измененные записи должны находиться вверху списка, а записи с более ранними датами изменения – внизу.
- В левой панели главного окна над списком заметок реализовать выпадающий список категорий заметок. При выборе пользователем определенной категории, в списке заметок должны отобразиться только заметки выбранной категории. При выборе пункта «All» отобразить все заметки.
- В класс «Проект» добавить свойство «Текущая заметка». Свойство текущей заметки меняется в случае просмотра пользователем какой-либо заметки в программе. Значение свойства должно также сохраняться в файл. При запуске программы значение свойства должно загружаться из файла, и пользователю в главном окне должна отобразиться последняя просмотренная им заметка.
- Реализовать удаление текущей заметки по нажатию на клавишу `Delete`. Перед удалением должно появиться окно с запросом на разрешение записи.

Приёмочное тестирование

Для приёмочного тестирования выполните следующую последовательность действий:

1. Установите приложение на компьютер с помощью собранного установочного пакета.
2. Запустите приложение. Окно программы должно быть пустым – в приложении не должно быть заметок.
3. Создайте три заметки в приложении разных категорий.
4. Переключитесь между заметками, показав, что смена текущей заметки происходит корректно.
5. Переключите отображаемую категорию заметок – в списке заметок должны остаться только заметки целевой категории. Снова отобразите все категории заметок – список заметок должен восстановиться.
6. Выберите вторую заметку и нажмите кнопку редактирования. Должно открыться окно редактирования заметки.
7. Введите название заметки более 50 символов. Элемент управления с названием заметки должен указать на некорректное значение.
8. Введите название заметки менее 50 символов. Элемент управления с названием должен стать корректным.
9. Поменяйте текст заметки. Нажмите «ОК». Отредактированная заметка должна подняться в списке заметок на первую позицию, время изменения заметки должно поменяться, отображаемый текст заметки также должен измениться.
10. Выберите вторую заметку и нажмите кнопку редактирования. Должно открыться окно редактирования. Измените название заметки, её текст и категорию. Нажмите «Cancel». Исходная заметка должна остаться без изменений.
11. Удалите третью заметку.
12. Закройте приложение. Должно произойти сохранение заметок в целевой файл.
13. Запустите приложение. В программе должны восстановиться заметки, созданные в предыдущую сессию.
14. Дайте руководителю провести исследовательское тестирование программы.

Варианты развития программы (по желанию студента, вне учебного курса дисциплины или для повышения оценки за пройденный курс)

* Создать элемент управления NoteControl для просмотра и редактирования заметок. Созданный элемент управления должен использоваться в правой панели главного окна и в окне создания/редактирования заметки. У элемента управления должно быть булево свойство ReadOnly, с помощью которого задается режим работы элемента управления (режим просмотра или режим редактирования заметки). Для удобства работы с элементом управления, элемент должен иметь свойство типа Note (класс «Заметка»), через которое

* Переделать текст записи на формат RichText вместо обычного текста string. Преимущество RichText по сравнению с обычным текстом – он может хранить рисунки, таблицы, разные шрифты и т.д. Для работы с RichText уже существует готовый элемент управления RichTextBox. Разберитесь с данным элементом управления и переделайте записи под формат RichText.

* Добавьте в программу функцию экспорта текущей записи в doc-файл – найти библиотеку для работы с doc-файлами предлагается самостоятельно.

* Добавьте в программу функцию печати текущей записи – для работы с принтером на платформе .NET существуют готовые классы.