**Слайд 1: Разработка системы классов для приложения "Риэлторская компания"**  
Здравствуйте! Меня зовут Даниил Ротницкий, и сегодня я представляю проект по разработке системы классов для приложения, предназначенного для управления сделками с недвижимостью в риэлторской компании. В этом проекте был использован объектно-ориентированный подход, который позволяет эффективно организовать работу с данными о сделках, участниках и объектах недвижимости.  
Риэлторская деятельность включает в себя не только покупку и продажу недвижимости, но и оформление документов, взаимодействие с клиентами и юридическое сопровождение сделок. Эти процессы часто требуют больших временных затрат и могут быть подвержены ошибкам, если выполняются вручную. Цель приложения — автоматизировать и упростить эти задачи, повысив эффективность работы риэлторов и снизив риски ошибок.  
Приложение предоставляет риэлторам удобные инструменты для управления сделками, создания и генерации документов, а также хранения данных о сделках и участниках. В ходе презентации я расскажу о выбранных инструментах для разработки, архитектуре приложения и его перспективах.

**Слайд 2: Описание предметной области: Риэлторская компания**  
Риэлторская деятельность — это посредничество между продавцами и покупателями недвижимости. Задача риэлторов — обеспечить безопасность и прозрачность сделок. Это требует эффективного управления данными о сделках, участниках и объектах недвижимости. Работа риэлтора включает такие этапы, как:

1. Поиск клиентов и объектов недвижимости.
2. Организация и сопровождение сделок.
3. Составление и подписание договоров купли-продажи, аренды и других юридических документов.  
   Для эффективного выполнения этих задач необходимы инструменты, которые помогут риэлторам быстро и правильно обрабатывать информацию, а также обеспечат сохранность и доступность данных о сделках.

**Слайд 3: Выбор инструментов для разработки**  
Для реализации проекта был проведен анализ нескольких языков программирования и сред разработки.  
**Сравнение языков программирования**  
Рассматривались два языка программирования: C++ и Python.

1. C++ — мощный язык, известный своей высокой производительностью и гибкостью. Однако его синтаксис достаточно сложен, что делает разработку более трудоемкой. Для работы с графическими интерфейсами и текстовыми файлами в C++ потребуется больше усилий, а также использование сторонних библиотек, что увеличивает время разработки.
2. Python — язык с простым и понятным синтаксисом, что позволяет значительно ускорить разработку. Он имеет большое количество встроенных и сторонних библиотек, что позволяет легко работать с текстовыми данными, генерировать документы и создавать графические интерфейсы. Python идеально подходит для быстрого создания прототипов и приложений, что сделало его идеальным выбором для этого проекта.  
   После оценки по таким критериям, как простота сопровождения, наличие библиотек, понятность синтаксиса и скорость разработки, был выбран Python.

**Сравнение сред разработки**  
Также был проведен анализ двух популярных сред разработки: PyCharm и Microsoft Visual Studio Code (VS Code).

1. PyCharm — специализированная среда для Python-разработки, предоставляющая мощные инструменты для работы с кодом: автодополнение, отладка, интеграция с системами контроля версий. PyCharm оптимизирован для Python и значительно ускоряет процесс разработки.
2. VS Code — универсальная среда разработки, поддерживающая множество языков, включая Python. Она легкая и гибкая, поддерживает большое количество расширений, что делает ее хорошим выбором для многоязычных проектов. Однако для полноценной работы с Python потребуются дополнительные настройки.  
   В результате была выбрана PyCharm, так как она предлагает удобные и мощные инструменты для работы с Python, что ускоряет процесс разработки и тестирования.

**Слайд 4: Используемые библиотеки**  
В процессе разработки были использованы следующие ключевые библиотеки:

1. python-docx — для генерации документов в формате .docx. С помощью этой библиотеки создаются договора купли-продажи, которые автоматически заполняются данными из сделок.
2. Для работы с данными о сделках используется стандартная библиотека для работы с текстовыми файлами. Информация о сделках сохраняется в текстовом файле с использованием формата CSV, где данные о сделках хранятся как строки, разделенные запятыми.  
   Эти библиотеки обеспечивают необходимый функционал для работы с графическим интерфейсом, созданием документов и обработкой данных.

**Слайд 5: Сценарий использования**  
Сценарий использования приложения включает следующие шаги:

1. Просмотр списка текущих сделок.
2. Добавление новой сделки с указанием данных о продавце, покупателе и объекте недвижимости.
3. Генерация договора купли-продажи на основе данных сделки.
4. Сохранение сделки в базе данных (текстовый файл).
5. Удаление завершенных или устаревших сделок..

**Слайд 6: Диаграмма классов**  
Диаграмма классов представляет структуру системы и взаимосвязи между ключевыми классами:

* **Person** — класс, представляющий участника сделки (покупателя или продавца).
* **RealEstate** — класс, представляющий объект недвижимости.
* **Deal** — класс, представляющий сделку, включающий объекты Person и RealEstate.
* **FileOperations** — модуль для работы с файлами.
* **ContractSaver** — модуль для генерации договоров.  
  Диаграмма показывает, как классы взаимодействуют друг с другом для выполнения основных функций приложения.

**Слайд 7: Блок-схема главного модуля**  
Блок-схема главного модуля отображает основные этапы работы приложения:

1. Запуск приложения.
2. Инициализация данных (загрузка сделок из файла).
3. Отображение главного меню.
4. Выбор пользователем действия (просмотр, добавление, удаление сделки, генерация договора).
5. Выполнение выбранного действия.
6. Сохранение изменений и завершение работы.  
   Блок-схема наглядно демонстрирует последовательность действий, выполняемых приложением.

**Слайд 8: Ключевые классы приложения**  
Приложение организовано с помощью нескольких ключевых классов:

1. **Класс Person** — представляет участника сделки (покупателя или продавца). Включает поля для имени, фамилии, отчества и телефона.
2. **Класс RealEstate** — представляет объект недвижимости. Хранит информацию о цене, площади, адресе и количестве комнат.
3. **Класс Deal** — представляет сделку. Включает данные о продавце, покупателе и объекте недвижимости, а также город, в котором происходит сделка.  
   Каждый класс играет важную роль в организации данных и взаимодействует с другими классами для обеспечения эффективной работы приложения.

**Слайд 9: Дополнительные модули**  
Программа также включает несколько дополнительных модулей для выполнения вспомогательных задач:

1. **Модуль file\_operations.py** — содержит функции для сохранения и загрузки данных о сделках в текстовом формате (например, CSV).
2. **Модуль contract\_saver.py** — используется для создания и сохранения договоров купли-продажи в формате .docx.  
   Эти модули обеспечивают дополнительный функционал для работы с данными и документами.

**Слайд 10: Результат реализации программного модуля**  
Результатом реализации является система, которая позволяет риэлторам эффективно управлять сделками с недвижимостью. В приложении реализованы следующие ключевые функции:

1. Просмотр сделок.
2. Добавление новой сделки.
3. Генерация договора.
4. Удаление сделок.  
   Данные о сделках сохраняются в текстовом файле, что позволяет работать с ними между сеансами работы программы.

**Слайд 11: Тестирование и отладка**  
После завершения разработки системы было проведено тестирование всех ключевых функций программы:

* Проверка корректности добавления и удаления сделок.
* Тестирование загрузки и сохранения данных из текстовых файлов.
* Проверка корректности генерации договоров.  
  В процессе тестирования были выявлены и исправлены ошибки, что позволило достичь стабильной работы всех функций.

**Слайд 12: Заключение и перспективы**  
Разработанная система позволяет риэлторам эффективно управлять сделками с недвижимостью, генерировать документы и хранить данные о сделках. Приложение простое в использовании и легко масштабируемое. В будущем планируется расширение функциональности:

* Добавление аналитики для анализа рынка недвижимости.
* Интеграция с внешними базами данных.
* Реализация картографической визуализации объектов недвижимости.  
  Приложение готово к использованию и может быть адаптировано для решения различных задач в сфере риэлторской деятельности.

**Слайд 13: Спасибо за внимание!**  
Спасибо за внимание! Готов ответить на ваши вопросы.