

Практическое занятие №2

Знакомство с UML

UML, или Unified Modeling Language, — унифицированный язык моделирования. Это графический язык, который с помощью диаграмм и схем описывает разнообразные процессы и структуры. Это не язык программирования, но чаще всего UML применяют именно в IT — с его помощью можно автоматически генерировать код.

Где еще используется UML

Кроме IT, UML используется в проектировании, документировании и построении бизнес-процессов. Он помогает строить схемы, которые визуализируют сложные структуры, действия или понятия. Особенность таких схем в том, что они унифицированы, то есть одинаковые связи и обозначения будут означать одно и то же в разных диаграммах. Это значит в том числе, что любой знающий UML человек легко поймет любую схему, созданную на этом языке.

В целом UML — открытый стандарт, язык широкого профиля. Он нужен для графического описания и визуализации абстрактной модели, а на практике эта модель может быть чем угодно — от архитектуры программы до описания целей деятельности. Название читается как «юмл» или «ю-эм-эл».

Кто пользуется UML

Разработчики, которым важно продемонстрировать, как будет устроена программа или те или иные ее компоненты, и визуализировать это.

Системные аналитики и архитекторы, которым такой способ представления может помочь в создании логичной и грамотной структуры ПО или отдельных его частей.

Технические писатели, так как UML, помимо всего прочего, используется для составления документации и автоматической генерации технических описаний.

Дизайнеры, которые занимаются созданием интерфейсов или визуализацией чего-то сложного. Они могут пользоваться и другими инструментами, но знать UML им стоит.

Бизнес-аналитики и менеджеры по развитию, которым диаграммы нужны для визуализации бизнес-процессов и структур.

Для чего нужны UML-диаграммы

- Для создания «чертежей» программы, схем, которые показывают, как будет устроено программное обеспечение изнутри, — то есть для проектирования. Это может быть описание связей между компонентами, модулей или сервисов, программных процессов и многого другого. В качестве инструмента проектирования UML может использоваться и вне IT.
- Для визуализации уже имеющейся программной структуры. Ряд инструментов позволяет создать UML на основе существующего кода, в таком случае диаграмма сгенерируется автоматически. Это называется реверс-инжинирингом.
- Для автоматической генерации кода или технической документации по нему, так как UML поддерживает возможность создания продукта на основе диаграмм. Но с этой функцией нужно быть крайне аккуратным: автоматически сгенерированный код позже нуждается в доработке. А вот созданная таким образом документация обычно наглядная и понятная.
- Для внутренней и внешней коммуникации между сотрудниками, заказчиками и другими: картинки и диаграммы UML понятнее людям, чем текстовые описания.

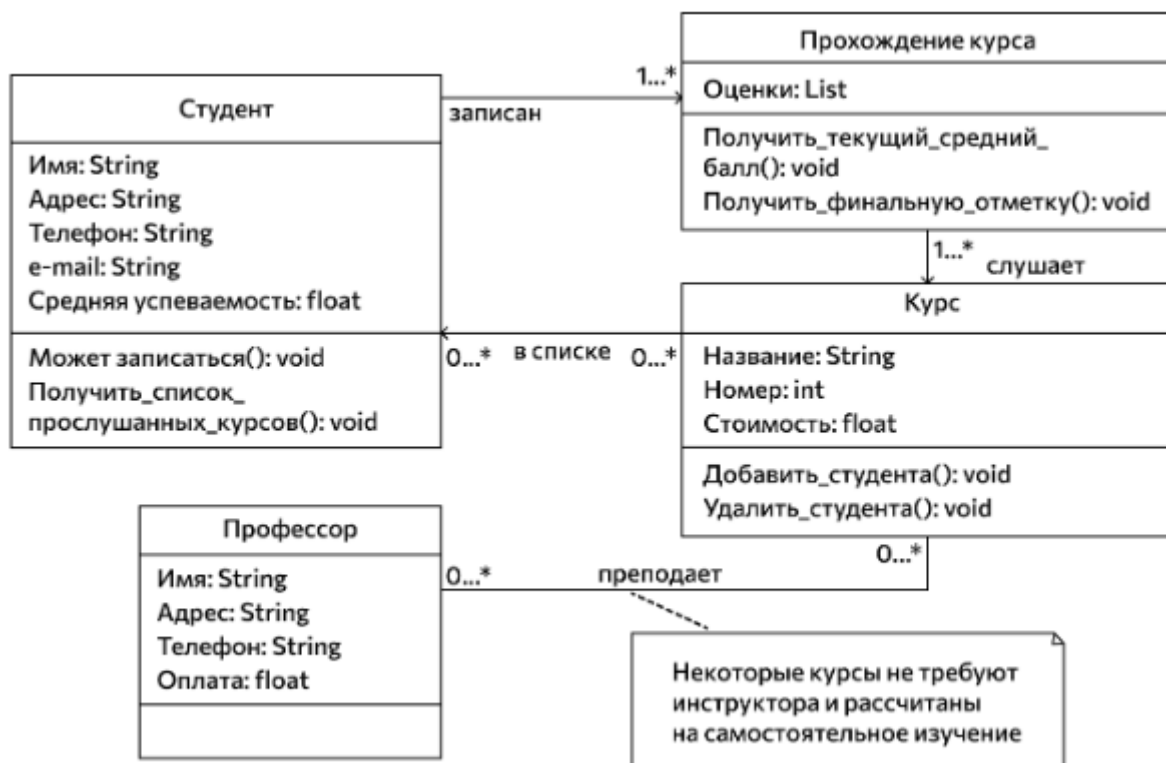


Рисунок 1: Пример UML-диаграммы классов

Важная особенность UML — этот язык поддерживает объектно-ориентированный подход, где все сущности представлены как объекты с определенными свойствами и методами. В диаграммах UML легко изобразить объекты, связи между ними, наследование и возможности передачи данных от одного объекта к другому.

Как устроена диаграмма UML

Схема UML — концептуальная: это значит, что она оперирует концепциями и связями между ними. Сама диаграмма состоит из фигур, значков, надписей, линий и контуров.

Фигуры обычно обозначают ту или иную концепцию: например, объект, класс, группу объектов или что-либо еще. Вариантов фигур в языке множество. Внутри одной фигуры могут находиться другие элементы, главное — чтобы они не пересекали границу.

Значки тоже обозначают разные сущности, но отличаются от фигур: внутри них нельзя ничего поместить. Это могут быть более мелкие автоматизированные структурные единицы, а могут быть служебные сущности, например для описаний.

Надписи могут быть обычными, подчеркнутыми, курсивными. Они именуют сущности, показывают, что есть что, и могут использоваться для описаний.

Линии могут быть прямыми, ломаными, изогнутыми, направленными и ненаправленными, штриховыми и какими-либо еще. Обычно они обозначают связи и зависимости сущностей друг от друга.

Контуры — это контейнеры, внутри которых помещаются концепции и связи между ними.

Виды UML-схем

Существует множество типов UML-диаграмм, и мы не будем описывать их все. Просто скажем, что условно они разделяются на три категории.

- **Объектные, или структурные** — диаграммы, которые описывают статическую структуру системы. Это, например, диаграммы классов в программе. В них показываются сами объекты и классы, связи между ними, зависимости и атрибуты.
- **Поведенческие** — схемы для описания поведения в динамике. Их еще называют динамическими. Это, например, диаграммы, которые описывают процессы.
- **Функциональные** — диаграммы, которые описывают особенности функционирования, показывают, как устроена работа того или иного объекта. Они демонстрируют пользователю функциональность так, чтобы это было понятно. Обычно это именно физические особенности реализации функций, поэтому еще их называют диаграммами реализаций. Разные функциональные диаграммы можно отнести к объектным или поведенческим.

Преимущества UML

Стандартизация. Элементы UML-диаграммы в разных схемах обозначают одно и то же — это все же унифицированный язык. Поэтому специалисты из разных сфер деятельности или из разных стран поймут друг друга, если будут демонстрировать продукт с помощью UML-диаграмм.

Наглядность. Изображение понятнее и нагляднее, чем текст. Сложную структуру проще описать в виде схемы, чем как-либо еще. UML-

диаграммы хорошо показывают, как устроена та или иная структура, и помогают ничего не пропустить.

Полнота описания. Обычно UML-диаграммы полные и подробные, то есть показывают не просто существование какой-то концепции, но и ее поведение и возможности. С их помощью можно понять, как будет вести себя сущность в тех или иных ситуациях, рассмотреть ее с разных сторон.

Широкое применение. UML активно используется во множестве отраслей, не только в IT. Благодаря широкому применению по модели много материалов, и высок шанс, что описанные с ее помощью диаграммы поймут специалисты из другой сферы. К тому же потенциальная широта применения дает возможность удобно описать большинство моделей, где бы вы ни работали.

Возможность реверс-инжиниринга. Если у какого-то уже существующего проекта нет документации или вам нужно срочно и наглядно объяснить другим людям, как он работает, реверс-инжиниринг поможет быстро решить проблему. Генерация UML на основе кода помогает и в других ситуациях, например при написании документации.

Удобная генерация. Речь идет и о генерации кода из UML, и о создании документации на основе диаграмм, и о реверс-инжиниринге. Все эти действия довольно простые, их не так сложно освоить.

Снижение риска ошибки. Человеческий фактор — проблема, которая может привести к ошибкам в описаниях, схемах или непосредственно в программных реализациях. Благодаря наглядности и высокому уровню автоматизации UML снижает риск возникновения таких ошибок.

Недостатки UML

Избыточность. Ранние версии UML критиковали из-за того, что некоторые обозначения и сущности избыточны и практически не используются в реальных проектах — в них просто нет необходимости, а язык они делают более громоздким. Со временем проблема стала менее заметной, но при начале работы с UML все равно стоит быть готовым к тому, что многие сущности понадобятся вам далеко не сразу — или не понадобятся вовсе.

Большое количество обозначений. Чтобы работать с UML правильно и корректно использовать составляющие диаграммы, понадобится выучить как минимум около ста обозначений. Но это важно, только если вы составляете UML-диаграмму вручную. При автоматической генерации нужные обозначения подставляются сами собой.

Как построить UML-схему

Существует два варианта: с помощью любого сервиса для построения диаграмм или благодаря программным модулям.

- **Сервисы и редакторы.** Сервисы дают возможность рисовать диаграммы вручную. Они похожи на графические редакторы, только вместо кистей и ручек в них есть уже готовые компоненты, из которых «собирается» схема. Человек проектирует диаграмму, подписывает элементы, оставляет заметки — такие редакторы предоставляют все необходимое. Некоторые из них могут связываться с сервисами для редактирования документов, хранения данных и других действий. Например, веб-приложение `diagrams.net` совместимо с облачными сервисами Google и с системами контроля версий Git. Подробнее о Git можно прочесть в нашей статье.
- **Модули и библиотеки.** Модули для IDE и библиотеки для популярных языков программирования позволяют создавать UML с помощью программного кода. Тут тоже есть две версии использования: сгенерировать диаграмму на основе имеющегося кода проекта или запрограммировать ее вручную. Можно описывать на языке программирования элементы и связи между ними, а потом «рисовать» визуализацию с помощью других инструментов. Такие утилиты и библиотеки существуют, например, для Java, JavaScript, PHP и других распространенных языков.

Практическая часть:

1. Установите программу StarUML. Для этого перейдите по ссылке и следуйте указаниям. Ссылка: <https://staruml.io/>
2. При первом запуске программы нажмите кнопку «Evaluate».
3. В меню выберите Model – Add Diagram – Use Case Diagram.
4. Попробуйте создать диаграмму.