Delivering Excellence in Software Engineering

# Диаграмма классов

## Общие сведения

® 2008. EPAM Systems. All rights reserved.

**Дать общие представления о диаграмме классов в нотации UML**


® 2008. EPAM Systems. All rights reserved.

**Курс включает в себя следующие презентации:**

- **UML. Введение**
- **Диаграмма вариантов использования**
- **Диаграмма деятельности**
- **Диаграмма последовательности**
- **Диаграмма классов**
- **Диаграмма состояний**


® 2008. EPAM Systems. All rights reserved.

1. Диаграмма классов
2. Классы
3. Отношения
4. Интерфейсы
5. Кооперации
6. Примеры

® 2008. EPAM Systems. All rights reserved.

1. Диаграмма классов
2. Классы
3. Отношения
4. Интерфейсы
5. Кооперации
6. Примеры

® 2008. EPAM Systems. All rights reserved.

**При проектировании системы информационная составляющая предметной области представляется в виде логической модели уровня сущностей (логическая модель)**

**Логическая модель описывает сущности (объекты данных) автоматизируемой предметной области и связи между ними**

**В унифицированном языке моделирования (UML) для представления логической модели можно использовать диаграмму классов**

**Диаграмма классов (class diagram) - это диаграмма, на которой показано множество классов, интерфейсов, кооперации и отношений между ними. Ее изображают в виде множества вершин и дуг**


® 2008. EPAM Systems. All rights reserved.

## Назначение диаграммы классов

- Диаграмма классов служит для представления статической структуры модели системы
- Диаграмма классов может отражать различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений

## Особенности диаграммы классов

- На диаграмме классов не указывается информация о временных аспектах функционирования системы


® 2008. EPAM Systems. All rights reserved.

## Диаграммы классов могут содержать

- Классы
- Отношения
- Интерфейсы
- Кооперации


® 2008. EPAM Systems. All rights reserved.

1. Диаграмма классов
2. Классы
3. Отношения
4. Интерфейсы
5. Кооперации
6. Примеры

® 2008. EPAM Systems. All rights reserved.

**Класс (class)** служит для обозначения множества объектов, которые обладают одинаковой структурой, поведением и отношениями с объектами из других классов. Графически класс изображается в виде прямоугольника, который дополнительно может быть разделен горизонтальными линиями на разделы или секции. В этих разделах могут указываться **имя класса**, **атрибуты** (переменные) и **операции** (методы).

Имя класса

Атрибуты класса

Операции класса

 ® 2008. EPAM Systems. All rights reserved.

# Имя класса

- **Имя класса** должно быть уникальным в пределах пакета, который описывается некоторой совокупностью диаграмм классов (возможно, одной диаграммой). Имя класса указывается в верхней секции прямоугольника полужирным шрифтом и должно начинаться с прописной буквы.

- **Абстрактный класс** – это класс, который не может иметь экземпляров (объектов). Для обозначения его имени используется наклонный шрифт.

 ® 2008. EPAM Systems. All rights reserved.

# Атрибуты класса

**Атрибут** - это именованное свойство класса, общее для всех его объектов, включающее описание множества значений, которые могут принимать экземпляры этого свойства

- Класс может иметь любое число атрибутов или не иметь их вовсе
- Атрибут является абстракцией данных объекта или его состояния

- В каждый момент времени любой атрибут объекта, принадлежащего данному классу, обладает вполне определенным значением
- Атрибуты представлены в разделе, который расположен под именем класса

При описании атрибута можно явным образом указывать его тип и значение, принимаемое по умолчанию

 ® 2008. EPAM Systems. All rights reserved.

# Операции класса

**Операция** – это реализация услуги, которую можно запросить у любого объекта класса для воздействия на поведение. Операция - это абстракция того, что может делать объект.

- У всех объектов класса имеется общий набор операций
- Класс может содержать любое число операций или не содержать их вовсе

- Как правило (хотя не всегда) обращение к операции объекта изменяет его состояние или его данные
- Операции класса изображаются в разделе, расположенном ниже раздела с атрибутами. При этом можно ограничиться только именами.

 ® 2008. EPAM Systems. All rights reserved.

# Советы

Изображая класс в UML, придерживайтесь следующих правил:

- показывайте только те свойства класса, которые важны для понимания абстракции в данном контексте
- разделяйте длинные списки атрибутов и операций на группы в соответствии с их категориями
- показывайте взаимосвязанные классы на одной и той же диаграмме

 ® 2008. EPAM Systems. All rights reserved.

1. Диаграмма классов
2. Классы
3. Отношения
4. Интерфейсы
5. Кооперации
6. Примеры

® 2008. EPAM Systems. All rights reserved.

# Отношения между классами

Помимо внутреннего устройства классов на диаграмме классов указываются различные отношения (связи) между ними. Совокупность типов отношений фиксирована.

Базовые отношениями в языке UML:

- Отношение зависимости (dependency relationship)
- Отношение ассоциации (association relationship)
- Отношение обобщения (generalization relationship)

 ® 2008. EPAM Systems. All rights reserved.

# Отношение зависимости

**Зависимость (dependency)** - отношение использования, согласно которому изменение в спецификации одного элемента может повлиять на другой элемент, его использующий, причем обратное не обязательно.

Графически зависимость изображается пунктирной линией со стрелкой, направленной от зависимого элемента на тот, от которого он зависит.

 ® 2008. EPAM Systems. All rights reserved.

# Отношение ассоциации

**Ассоциация (association)** - структурное отношение, показывающее, что объекты одного типа неким образом связаны с объектами другого типа.

Ассоциация, связывающая два класса, называется бинарной. Можно создавать ассоциации, связывающие сразу несколько классов; они называются n-арными.

У ассоциаций могут быть: имя, роль, кратность и агрегирование.

 ® 2008. EPAM Systems. All rights reserved.

# Отношение ассоциации

**Имя** описывает природу отношения. Чтобы избежать возможных двусмысленностей в понимании имени, указывается направление, в котором оно должно читаться.

**Роль** – это «лицо», которым класс, находящийся на одной стороне ассоциации, обращен к классу с другой ее стороны. Один класс может играть в разных ассоциациях как одну и ту же роль, так и различные.

**Кратность** - количество объектов, которое может быть связано посредством одного экземпляра ассоциации.
Кратность записывается либо как выражение, значением которого является диапазон значений, либо в явном виде.
Кратность можно задать равной единице (1), можно указать диапазон: «ноль или единица» (0..1), «много» (0..*), «единица или больше» (1..*), «любое число объектов, кроме 2 и 5» (0 . . 1, 3..4, 6..*).

 ® 2008. EPAM Systems. All rights reserved.

# Отношение ассоциации

**Агрегирование (aggregation)** показывает отношение типа «часть/целое», в котором один класс состоит из нескольких меньших классов

**Композиция (composition)** служит для выделения специальной формы отношения «часть/целое». Специфика связи заключается в том, что части не могут выступать в отрыве от целого, то есть с уничтожением целого уничтожаются и все его составные части.

 ® 2008. EPAM Systems. All rights reserved.

# Отношение обобщения

**Обобщение (generalization)** - это отношение между общей сущностью (суперклассом, или родителем) и ее конкретным воплощением (подклассом, или потомком).

Данное отношение описывает иерархию классов и наследование их свойств и поведения. Предполагается, что класс-потомок обладает всеми свойствами и поведением класса-предка, а также имеет свои собственные свойства и поведение, которые отсутствуют у класса-предка.

 ® 2008. EPAM Systems. All rights reserved.

# Советы

При моделировании отношений в UML соблюдайте следующие правила:

- используйте зависимость, только если моделируемое отношение не является структурным
- используйте обобщение, только если имеет место отношение типа «является»
- избегайте множественного наследования
- иерархия наследования не должна быть ни слишком глубокой (желательно не более пяти уровней), ни слишком широкой
- применяйте ассоциации прежде всего там, где между объектами существуют структурные отношения

 ® 2008. EPAM Systems. All rights reserved.

1. Диаграмма классов
2. Классы
3. Отношения
4. Интерфейсы
5. Кооперации
6. Примеры

® 2008. EPAM Systems. All rights reserved.

**Интерфейс (interface)** - набор операций, используемый для специфицирования услуг, предоставляемых классом или компонентом.

Интерфейс должен иметь уникальное имя.
Интерфейс может включать любое число операций.

**Особенности**

- Интерфейсы не содержат атрибутов
- Интерфейсы не содержат реализующих операции методов

Интерфейс специфицирует контракт класса, но не накладывает никаких ограничений на свою реализацию.

Связь интерфейса с реализующим его элементом можно графически представить двумя способами:

- интерфейс представляют в виде стереотипного класса и связывают его с классом **отношением реализации**. Отношение реализации объединяет отношения обобщения и зависимости
- отношение между интерфейсом и его реализацией изображается кружочком с одной стороны класса.

 ® 2008. EPAM Systems. All rights reserved.

Интерфейсы могут принимать участие в отношениях обобщения, ассоциации и зависимости. Интерфейс представляет собой стыковочный узел в системе. Он определяет условия контракта, после чего обе стороны - клиент и поставщик - могут действовать независимо друг от друга, полностью полагаясь на взаимные обязательства.

 ® 2008. EPAM Systems. All rights reserved.

# Советы

Изображая интерфейс на языке UML, руководствуйтесь приведенными ниже правилами:

- используйте сокращенную нотацию, если надо просто показать наличие стыковочного узла в системе
- используйте форму, если надо визуализировать детали самого сервиса

 ® 2008. EPAM Systems. All rights reserved.

1. Диаграмма классов
2. Классы
3. Отношения
4. Интерфейсы
5. Кооперации
6. Примеры

® 2008. EPAM Systems. All rights reserved.

**Кооперация (collaboration)** - это сообщество классов, интерфейсов и других элементов, которые работают совместно для обеспечения кооперативного поведения, более значимого, чем сумма его составляющих.

 ® 2008. EPAM Systems. All rights reserved.

1. Диаграмма классов
2. Классы
3. Отношения
4. Интерфейсы
5. Кооперации
6. Примеры

® 2008. EPAM Systems. All rights reserved.

 ® 2008. EPAM Systems. All rights reserved.

 ® 2008. EPAM Systems. All rights reserved.

# Web-сайты

**UML® Resource Page**
www.uml.org

**UML Tutorial**
http://www.sparxsystems.com.au/uml-tutorial.html

**UML 2.1 Tutorial**
http://www.sparxsystems.com.au/resources/uml2_tutorial/

# Книги

**Г. Буч, Д. Рамбо, А. Джекобсон “Язык UML Руководство пользователя”**

**Леоненков А. “Самоучитель UML”**

**Martin Fowler “UML Distilled: A Brief Guide to the Standard Object Modeling Language”**

 ® 2008. EPAM Systems. All rights reserved.

Delivering Excellence in Software Engineering

# Диаграмма классов

For more information, please contact:

**Sergey Kalinov**
Software Engineering Team Leader

EPAM Systems, Inc.
Ryazan, Russia (GMT+3)

Phone: +7 920 635 02 13
Email: Sergey_Kalinov@epam.com
http://www.epam.com

® 2008. EPAM Systems. All rights reserved.