

Задание по МДК 02.02 (Инструментальные средства разработки программного обеспечения)

группе ИС-21

Преподаватель Кузнецова Ольга Владимировна

1). Изучить тему «Проектирование на языке UML», выполнить краткий конспект в тетради:

ПРОЕКТИРОВАНИЕ НА ЯЗЫКЕ UML

Язык UML – это язык визуального моделирования, позволяющий разрабатывать концептуальные, логические и физические модели сложных систем.

Он предназначен для визуализации, анализа, спецификации, проектирования и документирования предметных областей, сложных систем вообще и ПС в частности.

Язык UML основан на следующих принципах объектно-ориентированного анализа и проектирования:

- *принцип абстрагирования* – предписывает включать в модель только те аспекты предметной области, которые имеют непосредственное отношение к выполнению проектируемой системой своих функций; абстрагирование сводится к формированию абстракций, определяющих основные характеристики внешнего представления объектов;
- *принцип инкапсуляции* – предписывает разделять элементы абстракции на секции с различной видимостью, что позволяет отделить интерфейс абстракции от его реализации; обычно скрываются структура объектов и реализация их методов;
- *принцип модульности* – определяет возможность декомпозиции проектируемой системы на совокупность сильно связанных и слабо сцепленных модулей; определение модулей выполняется при физической разработке системы, определение классов и объектов – при логической разработке;
- *принцип иерархии* – означает формирование иерархической структуры абстракций; принцип предписывает выполнять иерархическое построение моделей сложных систем на различных уровнях детализации;
- *принцип многомодельности* – обозначает, что при моделировании предметной области необходимо разрабатывать различные модели проектируемой сложной системы, отражающие различные аспекты ее поведения или структуры.

Для моделирования различных аспектов предметной области или проектируемой системы в языке UML предусмотрены следующие виды диаграмм:

- диаграмма вариантов использования (Use Case Diagram);

- диаграмма классов (Class Diagram);
- диаграммы поведения (Behavior Diagram), в том числе:
- диаграмма состояний (Statechart Diagram);
- диаграмма деятельности (Activity Diagram);
- диаграммы взаимодействия (Interaction Diagram), в том числе:
- диаграмма последовательности (Sequence Diagram);
- диаграмма кооперации (Collaboration Diagram);
- диаграммы реализации (Implementation Diagram), в том числе:
- диаграмма компонентов (Component Diagram);
- диаграмма развертывания (Deployment Diagram).

Модели языка UML подразделяются на два вида:

- *структурные модели (статические модели)* описывают структуру сущностей предметной области или компонентов моделируемой системы, включая их классы, атрибуты, связи, интерфейсы; к данному виду моделей относятся диаграммы вариантов использования, классов, компонентов, развертывания;
- *модели поведения (динамические модели)* описывают функционирование сущностей предметной области или компонентов системы во времени, включая их методы, взаимодействия между ними, изменение состояний отдельных сущностей, компонентов и системы в целом; к данному виду моделей относятся диаграмма состояний, диаграмма деятельности, диаграмма последовательности, диаграмма кооперации.

Все модели языка UML подразделяются на три уровня:

- *концептуальные модели* представляют собой верхний, наиболее общий и абстрактный уровень описания моделируемой системы; к данному уровню относится диаграмма вариантов использования; с уровня концептуальной модели должно начинаться моделирование предметной области или проектируемой системы (программного средства);
- *логические модели* представляют собой второй уровень описания моделируемой системы; элементы моделей данного уровня не имеют физического воплощения и отражают логические аспекты структуры и поведения реальной предметной области или системы; логические модели должны строиться после концептуальных моделей; к данному уровню относятся диаграммы классов, состояний, деятельности, последовательности, кооперации;

- *физические модели* представляют собой нижний уровень описания моделируемой системы; элементы моделей данного уровня представляют собой конкретные материальные сущности физической системы; данные модели рекомендуется строить в последнюю очередь; к данному уровню относятся диаграммы компонентов и развертывания.

Процесс объектно-ориентированного анализа и проектирования, базирующийся на построении различных типов диаграмм UML, получил название рационального унифицированного процесса RUP (Rational Unified Process).

Основы данного процесса разработаны одним из авторов языка UML Джекобсоном.

Диаграмма вариантов использования описывает функциональное назначение моделируемой предметной области или системы.

Диаграмма классов является основной для создания кода приложения. Она описывает внутреннюю структуру программного средства, наследование и взаимное положение классов.

Диаграмма состояний описывает возможные последовательности состояний и переходов, выполняемых в ответ на некоторые события. Данная диаграмма характеризует поведение элементов модели в течение их жизненного цикла.

Диаграмма деятельности моделирует алгоритмическую и логическую реализации выполнения операций в системе и является аналогом схем алгоритмов, предназначенным для использования в объектно-ориентированных приложениях.

Диаграмма последовательности отображает синхронные процессы, описывающие взаимодействие объектов модели во времени. Время в данной модели присутствует в явном виде.

Диаграмма кооперации описывает структурные связи между взаимодействующими объектами модели.

Диаграмма компонентов описывает физическое представление проектируемой системы и позволяет определить ее архитектуру в терминах модулей, исходных и исполняемых кодов, файлов.

Диаграмма развертывания (диаграмма размещения) отображает общую конфигурацию и топологию распределенной системы. Данная диаграмма представляет распределение программных компонентов по отдельным узлам системы и маршруты передачи информации между ними.

Диаграмма вариантов использования является первой из диаграмм, разрабатываемых при моделировании предметной области, системы или программного

средства. Она является базой при разработке спецификации функциональных требований и имеет основополагающее значение с точки зрения полноты и корректности дальнейшего моделирования проектируемой системы (программного средства).