

Курсовое проектирование

Курсовые работы выполняются в соответствии с теми задачами, которые определены для включения в выпускную квалификационную работу.

Текст курсовой работы рекомендуется печатать:

шрифт – Times New Roman, размер – 14, интервал – 1,5, все поля – 2 см.

Объем курсовой работы составляет 30 – 50 стр.

Структура курсовой работы

Введение

1. Описание предметной области

2. Проектирование подсистемы

2.1. Функциональная модель предметной области

2.2. Концептуальная модель подсистемы

2.3. Логическая модель подсистемы

2.4. Динамическое представление логической модели подсистемы

2.5. Физическая модель подсистемы

Заключение

Список использованных источников

Во *введении* раскрывается актуальность избранной тематики, формулируется цель, задачи, направления исследования; указывается объект и предмет исследования

В пункте 1 (*Анализ предметной области*) следует привести:

- характеристику предприятия, деятельность которого является объектом рассмотрения;

- описание рассматриваемой деятельности и ее свойств как объекта управления;

- список функций управления, выполнение которых должно быть автоматизировано.

Пункт 2.1 (*Функциональная модель предметной области*)

Методология функционального моделирования представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели какой-либо предметной области.

Функциональная модель предметной области разрабатывается с целью выявления базовых функций и их взаимосвязи. Для этих целей посредством программного продукта BPWin 4.0 строятся диаграммы.

Для построения функциональной модели используется стандарт IDEF0 (Icam DEFinition). Функциональная модель по стандарту IDEF0 представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Модель содержит три типа диаграмм: контекстная; декомпозиции; дерева узлов.

Контекстная диаграмма является вершиной древовидной структуры и представляет общую функцию системы и её взаимосвязь с внешним миром.

На основе контекстной диаграммы выполняется функциональная декомпозиция системы до достижения необходимого уровня подробности описания.

Пункт 2.2 (*Концептуальная модель подсистемы*)

Диаграмма вариантов использования (use case diagram) описывает функциональное назначение системы или, другими словами, то, что система будет делать в процессе своего функционирования. Диаграмма вариантов использования является исходным концептуальным представлением, или концептуальной моделью системы, в процессе ее проектирования и разработки.

Разработка диаграммы вариантов использования преследует цели:

- определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы;
- сформулировать общие требования к функциональному поведению проектируемой системы;
- разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;
- подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Пункт 2.3 (*Логическая модель подсистемы*)

Центральное место в объектно-ориентированном анализе и проектировании занимает разработка логической модели системы в виде *диаграммы классов*. Диаграмма классов (class diagram) служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений. На данной диаграмме не указывается информация о временных аспектах функционирования системы. С этой точки зрения диаграмма классов является дальнейшим развитием концептуальной модели проектируемой системы.

Диаграмма состояний описывает процесс изменения состояний только одного класса, а точнее – одного экземпляра определенного класса, т.е. моделирует все возможные изменения в состоянии конкретного объекта.

При этом изменение состояния объекта может быть вызвано внешними воздействиями со стороны других объектов или извне. Именно для описания реакции объекта на подобные внешние воздействия и используются диаграммы состояний.

Главное предназначение этой диаграммы - описать возможные последовательности состояний и переходов, которые в совокупности характеризуют поведение элемента модели в течение его жизненного цикла. Диаграмма состояний представляет динамическое поведение сущностей на

основе спецификации их реакции на восприятие некоторых конкретных событий. Диаграмма состояний, по существу, является графом специального вида, который представляет некоторый автомат.

Диаграмма деятельности (англ. activity diagram) — UML-диаграмма, на которой показаны действия, состояния которых описано на диаграмме состояний.

Пункт 2.4 (*Динамическое представление логической модели подсистемы*)

Для моделирования взаимодействия объектов в языке UML используются соответствующие диаграммы взаимодействия. Говоря об этих диаграммах, имеют в виду два аспекта взаимодействия. Во-первых, взаимодействия объектов можно рассматривать во времени, и тогда для представления временных особенностей передачи и приема сообщений между объектами используется *диаграмма последовательности*. Временной аспект поведения может иметь существенное значение при моделировании синхронных процессов, описывающих взаимодействия объектов. Именно для этой цели в языке UML используются диаграммы последовательности.

Во-вторых, можно рассматривать структурные особенности взаимодействия объектов. Для представления структурных особенностей передачи и приема сообщений между объектами используется *диаграмма кооперации*.

Пункт 2.5 (*Физическая модель подсистемы*)

Различные элементы логического представления, такие как классы, ассоциации, состояния, сообщения, не существуют материально или физически. Они лишь отражают наше понимание структуры физической системы или аспекты ее поведения. Для создания конкретной физической системы необходимо некоторым образом реализовать все элементы логического представления в конкретные материальные сущности. Для описания таких реальных сущностей предназначен другой аспект модельного представления, а именно физическое представление модели.

Для того чтобы пояснить отличие логического и физического представлений, рассмотрим в общих чертах процесс разработки некоторой программной системы. Ее исходным логическим представлением могут служить структурные схемы алгоритмов и процедур, описания интерфейсов и концептуальные схемы баз данных. Однако для реализации этой системы необходимо разработать исходный текст программы на некотором языке программирования (C++, Pascal, Basic/VBA, Java). При этом уже в тексте программы планируется такая организация программного кода, которая предполагает его разбиение на отдельные модули.

Тем не менее исходные тексты программы еще не являются окончательной реализацией проекта, хотя и служат фрагментом его физического представления. Очевидно, программная система может считаться реализованной в том случае, когда она будет способна выполнять функции своего целевого предназначения. А это возможно, только если программный код системы будет реализован в форме исполняемых модулей,

библиотек классов и процедур, стандартных графических интерфейсов, файлов баз данных. Именно эти компоненты являются необходимыми элементами физического представления системы.

Диаграмма компонентов, в отличие от ранее рассмотренных диаграмм, описывает особенности физического представления системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код.

Диаграмма развертывания предназначена для визуализации элементов и компонентов программы, существующих лишь на этапе ее исполнения (runtime). При этом представляются только компоненты-экземпляры программы, являющиеся исполнимыми файлами или динамическими библиотеками. Те компоненты, которые не используются на этапе исполнения, на диаграмме развертывания не показываются. Так, компоненты с исходными текстами программ могут присутствовать только на диаграмме компонентов. На диаграмме развертывания они не указываются.

Заключение содержит перечень основных полученных в работе результатов и сделанных выводов.

Список использованных источников должен содержать не менее 20 источников.