# Краткие указания по подготовке отчета по практическим работам по дисциплине «Проектирование ИС»

## Цель работы

Целью практической работы является закрепление изученного теоретического материала и практических навыков, полученных в ходе освоения курса «Проектирование информационных систем». Для достижения указанной цели, в ходе выполнения работы студенты должны пройти все этапы одного из процессов проектирования и решить задачи, стоящие на каждом из этапов.

## Общая структура отчета

В качестве процесса проектирования для практической работы используется Rational Unified Process (RUP), один из современных итеративных процессов разработки ИС. Выбор данного процесса объясняется, во-первых, его достаточной формальной проработанностью, а во-вторых, тесной связкой с унифицированным языком моделирования (Unified Modeling Language, UML), что позволяет одновременно освоить использование данного языка для проектирования информационных систем. UML фактически является отраслевым стандартом для построения объектно-ориентированных моделей и владение им полезно для специалиста в сфере информационных технологий.

Таким образом, курсовая работа структурируется в соответствии с процессами, протекающими внутри проекта разработки ИС, выполняющегося с использованием RUP, а именно:

* бизнес-моделирование;
* управление требованиями;
* анализ предметной области;
* проектирование;
* реализация;
* тестирование;
* развертывание (в данной работе не затрагивается).

Каждому из процессов должен быть отведен раздел отчета о практической работе.

## Бизнес-моделирование

Так как дисциплина «Проектирование ИС» выполняется в рамках общего учебного процесса, на основе выполненных ранее практических и курсовых работ, то в разделе бизнес-моделирования следует разместить модели процессов предметной области, если они выполнялись ранее.

В рамках дисциплины «Проектирование ИС» построение моделей бизнес-процессов не рассматривается и как самостоятельная задача бизнес-моделирование не ставится. Поэтому, если модели бизнес-процессов ранее не строились, в данном разделе следует разместить общее описание предметной области, ссылки на нормативные документы, регулирующие деятельность в предметной области и так далее.

Состав и количество диаграмм и иных материалов для данного раздела не регламентируется. Цель его разработки — сформировать общее представление о выбранной предметной области, протекающих в ней процессах и её окружении. Объем раздела — до 5 страниц.

## Управление требованиями

Необходимо выявить как функциональные, так и нефункциональные требования к проектируемой системе. Поскольку основой является RUP и UML, то функциональные требования должны оформляться в виде вариантов использования.

Для данного раздела должны быть построены диаграммы двух видов — диаграмма вариантов использования (Use-Case Diagram), а также диаграммы деятельности (Activity Diagram).

Диаграмма вариантов использования может быть одна, но если она получается слишком большой, её можно разделить на несколько более мелких. Ключевым требованием является использование всех трех типов отношений между вариантами использования: включения, расширения и обобщения. Если какой-то из типов применить не представляется возможным, следует пояснить это в тексте записки.

Диаграмм деятельности должно быть не менее двух. Одна должна представлять укрупненный взгляд на систему, схему «рабочего цикла». Вторая должна описывать один из ключевых вариантов использования.

Также в пояснительную записку следует включить текстовое описание одного из вариантов использования (не того, для которого построена диаграмма) в формате, предложенном Алистером Коберном (описание формата находится в приложении 1).

Нефункциональные требования следует выделить исходя из общих соображений о функционировании предметной области. В настоящий момент не ставится задача формирования подробных нефункциональных требований, тем не менее, следует попытаться обозначить требования к объемам хранимых и обрабатываемых данных, аппаратным ресурсам (в первую очередь, к постоянной и оперативной памяти), быстродействию и надежности.

## Анализ предметной области

На этапе анализа предметной области следует выявить классы, существующие в ней, определить отношения между ними, а также описать взаимодействие в рамках реализации вариантов использования. Кроме того, необходимо определить архитектуру будущей системы.

В разделе анализа должны быть построены три типа диаграмм: диаграмма классов этапа анализа (Class diagram), диаграмма коммуникации (Communication diagram) и диаграмма развертывания (Deployment diagram).

Диаграмма классов этапа проектирования предназначена для отображения классов, выявленных в предметной области (классы сущностей), а также граничных классов и управляющих классов.

Для выявления классов сущностей следует использовать подходы, рассмотренные в курсе «Объектно-ориентированное программирование». Для упорядочения выявленных классов нужно применить анализ по категориям (перечень категорий находится в приложении 2).

Диаграмма коммуникации используется для отображения взаимодействия между выявленными классами. Должна быть разработана как минимум одна диаграмма этого типа для описания работы ключевого варианта использования.

Диаграмма развертывания предназначена для отображения составных частей системы и содержит информацию об аппаратных узлах и работающих на них компонентах.

Таким образом, раздел анализа должен содержать два подраздела – анализ предметной области, описывающий выявленные классы и их взаимодействие и содержащий перечень классов по категориям, диаграмму классов и одну или несколько диаграмм коммуникации.

Второй подраздел – выбор и описание архитектуры системы. Должно быть определено разбиение системы на слои и для каждого слоя, по возможности, выбраны и описаны архитектурные шаблоны. Также должны быть выбраны средства реализации системы – целевая платформа, язык программирования и СУБД, библиотеки для доступа к данным и организации пользовательского интерфейса. Результаты разработки архитектуры следует оформить с помощью одной или нескольких диаграмм компонентов (Component diagram), если они требуются. Обязательно построение диаграммы развертывания (Deployment diagram), где должны быть отображены и компоненты системы.

## Проектирование системы

В ходе проектирования системы должно быть выполнено детальное описание классов, выявленных на этапе анализа – должны быть уточнены типы данных для свойств исходя из выбранной платформы и языка, определено поведение.

Также модель должна быть дополнена классами, входящими в состав выбранной среды, а также классами, обеспечивающими работу системы.

В ходе проектирования должны быть построены диаграммы трех видов: диаграмма классов этапа проектирования (Class diagram), диаграмма последовательности (Sequence diagram) и диаграмма состояний (Statechart diagram).

Диаграмма классов этапа проектирования должна содержать классы, которые уже будут реализованы в системе. Выявленные ранее классы должны быть увязаны с классами выбранной платформы, как при помощи наследования и реализации, так и при помощи использования. Возможно, придется построить несколько диаграмм классов для каждой из частей системы.

Диаграмма последовательности служит для подробного описания взаимодействия объектов между собой в ходе выполнения функций системы. Нужно построить две-три диаграммы, описывающих работу разных уровней системы.

Диаграмма состояний предназначена для отображения жизненного цикла объекта. Должна быть построена хотя бы одна диаграмма для какого-то из ключевых классов сущностей.

## Реализация системы

В разделе реализации необходимо поместить описание физической структуры системы – на какие модули система разбита, в каких проектах эти модули реализуются, и примеры программного кода разработанных модулей. Полученная структура должна соответствовать построенной ранее диаграмме компонентов.

Также, если для системы используется какой-то механизм сборки, то следует описать и его.

## Тестирование системы

Для системы могут быть разработаны различные типы тестов. Обязательным является применение модульного тестирования. Какой именно подход и фреймворк для модульного тестирования будет применен, зависит от выбранных средств реализации системы.

Должны быть сформированы модульные тесты для двух-трёх ключевых классов системы.

Если будет возможность задействовать другие виды автоматизированного тестирования (интеграционное тестирование, тестирование пользовательского интерфейса, тесты производительности и т. д.), следует их также описать в этом разделе и привести примеры тестов.

# Приложение 1. Формат текстового описания варианта использования.

<Название должно быть в виде краткой фразы с глаголом в неопределенной форме совершенного вида и отражать цель>

**Контекст использования:** <более длинное предложение цели, при необходимости условия ее нормального достижения>

**Область действия:** <область действия проектирования, в которой разрабатываемая система рассматривается как «черный ящик»>

**Уровень:** <один из следующих: обобщенный, цели пользователя, подфункции>

**Основное действующее лицо:** <ролевое имя для основного действующего лица или описание>

**Участники и интересы:** <список участников и ключевых интересов в данном варианте использования>

**Предусловие:** <то, что, как ожидается, уже имеет место>

**Минимальные гарантии:** <как защищаются интересы участников при всех исходах>

**Гарантии успеха:** <что имеет место, если цель достигнута>

**Триггер:** <то, что запускает вариант использования, возможно, временное событие>

**Основной сценарий:**

<запишите здесь шаги сценария от триггера до достижения цели и затем какую-либо очистку> <номер шага> <описание действия>

**Расширения:**

<запишите здесь по одному расширению, каждое из которых обращается к шагу основного сценария>

<изменяемый шаг> <условие>: <действие или подчиненный вариант использования>

<изменяемый шаг> <условие>: <действие или подчиненный вариант использования>

**Список изменений в технологии и данных:**

<запишите здесь изменения, которые в итоге вызовут ветвление в сценарии>

<номер шага или изменения> <список изменений>

<номер шага или изменения> <список изменений>

**Вспомогательная информация:**

<то, что необходимо для проекта в качестве дополнительной информации>

# Приложение 2. Перечень категорий классов для модели анализа.

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Категория** | **Возможные классы** |
| 1 | Физические или материальные объекты | Товар, материал |
| 2 | Спецификации, описания объектов | Характеристики товара |
| 3 | Места | Склад, магазин, цех |
| 4 | Транзакции | Продажа, поставка, оплата |
| 5 | Элементы транзакций | Проданный товар |
| 6 | Роли людей | Продавец, клиент |
| 7 | Контейнеры других объектов | Склад |
| 8 | Содержимое контейнеров | Товары на складе |
| 9 | Другие компьютеры и прочие системы, внешние по отношению к создаваемой | ККМ, система обработки платежей по пластиковым картам |
| 10 | Организации | Контрагент, государственный орган |
| 11 | События | Поступление товара, отгрузка товара |
| 12 | Процессы | Заключение договора, выполнение ремонта |
| 13 | Каталоги | Прайс-лист, меню |
| 14 | Записи о деятельности | Чек, договор |