Федеральное агентство по образованию РФ

ГОУ ВПО Нижегородский государственный университет им. Н.И. Лобачевского

Факультет Вычислительной математики и кибернетики

Кафедра Математического обеспечения ЭВМ

УЧЕБНЫЙ КУРС

**«Объектно-ориентированный анализ и проектирование»**

для подготовки по направлению «Информационные технологии»

**ФУНКЦИОНАЛЬНАЯ СПЕЦИФИКАЦИЯ**

Нижний Новгород  
2006

**Содержание**

[1. Видение и рамки](#h.30j0zll)

[2. История проекта](#h.1fob9te)

[3. Цели дизайна](#h.3znysh7)

3.1. Требования пользователя

3.2. Системные требования

3.3. Сценарии использования

[4. Исключенные возможности и неподдерживаемые сценарии](#h.1t3h5sf)

[5. Предположения и зависимости](#h.4d34og8)

[6. Проект решения](#h.2s8eyo1)

6.1. Концептуальный проект

6.2. Логический проект

6.3. Физический проект

[7. Требования к инсталляции и деинсталляции](#h.lnxbz9)

1. **Видение и рамки**

Информация по видению/рамкам проекта должна сконцентрировать внимание читателя на ключевых элементах решения. Данная информация – стратегический элемент решения, который необходимо понять, прежде чем начинать вникать в детали функциональной спецификации. Включение этой информации формирует общее видение проекта и общие ожидания от его реализации.

Приведите здесь обзор концепции (видения) и рамок проекта.

Наше видение: удобный и эффективный WEB-сервис поиска и покупки алгоритмов. Многие разработчики нуждаются в простых решениях для написания и развития сложных систем, однако не всегда находятся ресурсы для реализации трудоёмких алгоритмов. В то же время другие разработчики уже имеют готовые решения и готовы поделиться ими за определенную плату.

Наша система является связующей компонентой между ними и даёт возможность: обмениваться реализациями алгоритмов (свободно/платно), проводить проверку корректности (компиляция, запуск тестов).

Успех проекта определяется выполнением всех поставленных задач в двухмесячный срок.

1. **История проекта**

Раздел «История проекта» перечисляет ключевые события в процессе создания решения в хронологической последовательности, так, как их видит команда. Перечисляются важные принятые решения. Данная информация может оказаться полезной для оценки того, почему проект был успешным (или, напротив, провальным) и почему. Подобный анализ будет крайне полезен при создании подобных решений в будущем.

Приведите здесь основные события и важные решения в процессе реализации проекта.

* Определение концепции проекта - **сделано**
* Разработка проектной документации **- в прогрессе**
  + Концепция проекта **- сделано**
  + Структура проекта **- сделано**
  + Функциональная спецификация **- в прогрессе**
  + Спецификации и сценарии тестов **- не начато**
  + Тестирование и отчеты об ошибках **- не начато**
  + Постпроектный анализ **- не начато**
* Разработка прототипа **- в прогрессе**
  + Настройка окружения разработчика **- сделано**
  + Создание простого web-сервера, в котором используется база данных, отображается web-страница с кнопкой, и реализована функция, принимающая запрос по нажатию кнопки. **- в прогрессе**
  + Дополнение простого web-сервера расширенным набором html-шаблонов для регистрации, авторизации, отображения списка алгоритмов, добавления и тестирования алгоритмов. Подразумевается, что на данном этапе такие страницы содержат минимум функционала. **- не начато**
* Разработка независимых компонент **- не начато**
  + Разработка детальных html-шаблонов **- не начато**
  + Обработка запросов с html страниц **- не начато**
  + Проработка архитектуры базы данных **- не начато**
  + Реализация компоненты, отвечающей за построение алгоритмов **- не начато**
    - построение алгоритмов, написанных на C/C++ **- не начато**
    - построение алгоритмов, написанных на C# **- не начато**
    - построение алгоритмов, написанных на Pascal **- не начато**
  + Реализация компоненты, отвечающей за тестирование алгоритмов **- не начато**
* Интеграция компонент **- не начато**
* Разработка тестов и тестирование **- не начато**

1. **Цели дизайна**

Данный раздел документа обобщает выполненный ранее анализ требований. Формулируются требования с точки зрения заказчика, пользователей, аппаратного и программного окружения. Данные требования, сформулированные ранее в общем виде, должны быть скорректированы в требования к решению и его отдельным компонентам в терминах команды разработчиков. В результате происходит уточнение целей проекта, сформулированных ранее в видении/рамках.

* 1. **Требования пользователя**

Раздел «Требование пользователя» перечисляет выявленные требования к решению с точки зрения заказчика и конечных пользователей.

Приведите здесь требования заказчика и конечных пользователей.

1. Регистрация и авторизация пользователей

2. Пользовательские данные защищены от несанкционированного доступа

3. Возможность загружать исходный код алгоритмов

4. Возможность задавать тестовые данные

5. Возможность протестировать работу алгоритма на заданных данных

6. Возможность разграничивать уровни доступа к алгоритму (платный/свободный)

7. Возможность задавать теги для поиска алгоритмов

8. Возможность скачать/приобрести выбранный алгоритм

9. Возможность менеджера модерировать данные системы

* требования к интерфейсам
  1. **Системные требования**

Сформулируйте здесь требования к аппаратному и программному окружению.

1. Сервер работает под управлением ОС Windows

2. C/C++ компилятор: MSVC 2010

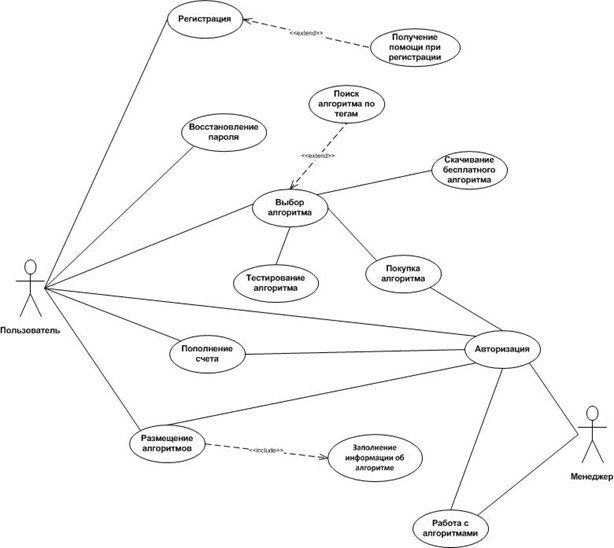
3. C# компилятор: csc.exe

4. FreePascal компилятор: (ppc386 или ppcrossx64)

5. HDD, RAM, CPU Cores

* 1. **Сценарии использования**

Приведите диаграмму вариантов использования. Прокомментируйте диаграмму. Перечислите варианты использования, опишите их суть. Для вариантов использования укажите сценарии. Для написания данного раздела рекомендуется язык UML.



* **добавить активити диаграмму для каждого сраного кружочка**

1. **Исключенные возможности и неподдерживаемые сценарии**

В данном разделе предполагается указание требований, которые не найдут отражение в текущем релизе проекта. При этом должны быть указаны как требования пользователей, требования к аппаратному и программному окружению. Для каждого из требований, которые не планируется удовлетворять, необходимо привести обоснование (почему это не делается). В данной секции можно сформулировать соображения относительно того, что необходимо сделать в будущих релизах, чтобы удовлетворить требованиям, и когда (и в каком случае) это может быть сделано.

Необходимо дополнительно отметить важность данного раздела. В любом случае важно честно сформулировать, какую функциональность мы создаем. Необходимо всеми силами избегать непонимания сторонами того, о чем конкретно они договорились. Соображения вида «сейчас все ОК, а там будет видно» на практике приводят к катастрофам на завершающих этапах.

Укажите здесь исключенные возможности и неподдерживаемые сценарии.

Поддержка реальных платёжных систем для покупки и продажи алгоритмов. Эта функциональность остаётся за рамками решения по следующим причинам: на данном этапе это студенческий проект, не подразумевающий реального использования, следовательно не рационально тратить ресурсы на поддержку данной функциональности. + расширить

1. **Предположения и зависимости**

Раздел «Предположения и зависимости» перечисляет и определяет предположения и зависимости, ориентированные на проект и сделанные в рамках создания функциональной спецификации.

Укажите здесь предположения и зависимости.

● Поддержка браузера Google Chrome.

● Алгоритмы, добавляемые в систему должны работать не более 10 минут на предоставляемых тестовых наборах.

● Объем тестовых данных не должен превышать 1 мегабайта

● Исходный код алгоритма должен быть помещен в единственный файл

● Привязка к реальным платёжным системам реализована не будет

+ расширить

1. **Проект решения**

Проект решения обобщает документы, созданные в рамках проектирования будущего решения, в краткой сжатой форме. При этом указываются назначение и важность для проекта указанных документов. Эта информация способствует выработки у читателя ясного представления о концепции проектирования решения.

* 1. **Концептуальный проект**

Концептуальный проект – стратегический документ, содержащий информацию о том, как решение будет реализовывать рассмотренные ранее сценарии вариантов использования. Сценарии описывают всех пользователей решения, а также их возможные действия. Данный раздел содержит проект системы в терминах пользователя на основе имеющейся диаграммы вариантов использования и сценариев вариантов использования.

Данный раздел должен быть кратким. В противном случае, он будет трудным для понимания, что существенно снизит его полезность. Необходимо обратить особое внимание на то, что часть читателей данного раздела не владеет технической терминологией и знаниями. Поэтому важно сформулировать положения дизайна, по возможности не привлекая специальной терминологии.

Применение визуальных средств (UML) может существенно упростить понимание и всячески приветствуется.

Приведите здесь концептуальный проект решения (архитектура решения, важные проектные решения, возможные варианты, их плюсы и минусы) в терминах, понятных заказчику.

описание проекта на человеческом олбанском езыге [TB]

* 1. **Логический проект**

Создание логического проекта – следующая стадия в проектировании решения. Так, в концептуальном проекте мы показываем будущее решение в терминах заказчика, не привлекая технологии и ничего не говоря о внутреннем устройстве. В логическом проекте мы должны выделить основные структурные элементы будущего решения, разобраться с их иерархией, поведением, атрибутами, взаимосвязями.

В результате логического проектирования мы должны получить абстрактную модель решения. При этом речь о привлечении каких-то технологий реализации данной модели пока не идет.

На этапе логического проектирования могут быть решены следующие важные задачи:

1) Сбор откликов заинтересованных лиц. При этом мы можем обнаружить ошибки проектирования на ранней стадии.

2) Проверка соответствия проекта требованиям.

3) Создание базиса для последующей разработки системы тестов.

Приведите здесь логический проект решения (объекты[[1]](#footnote-0), атрибуты, поведение, связи). Активно используйте нотацию UML.

Диаграмма классов+паблик методы

* 1. **Физический проект**

Физический проект представляет собой проекцию логического проекта на имеющееся аппаратное, программное и технологическое окружение. Так, нам надлежит выбрать те или иные технологии для реализации идей, заложенных в концептуальном и уточненных в логическом дизайне.

Физический проект фактически представляет собой документацию с четким указанием параметров реализуемой функциональности в терминах разработчиков программного обеспечения.

Кроме того, решаются как вопросы создания компонентов, допускающих повторное использование, так и вопросы применения написанного ранее кода (или сторонних разроботок).

Приведите здесь физический проект решения.

по компам\серверам

1. **Требования к инсталляции и деинсталляции**

Приведите здесь информацию по тому, как будет осуществляться инсталляция/деинсталляция решения. Сформулируйте требования к этим процессам.

1. Не то же самое, что «объект» в терминах объектно-ориентированного программирования. [↑](#footnote-ref-0)