

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Методические рекомендации по выполнению
самостоятельной работы
по дисциплине

***Объектно-ориентированный анализ и
программирование***

для студентов направления

231000.62

«Программная инженерия»

Томск – 2012

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)**

Кафедра автоматизации обработки информации

Утверждаю:
Зав. каф. АОИ
профессор
_____ Ю.П. Ехлаков
«__» _____ 2012 г.

Методические рекомендации по выполнению
самостоятельной работы
по дисциплине

Объектно-ориентированный анализ и программирование

для студентов направления

231000.62

«Программная инженерия»

Разработчик:
ст. преподаватель каф. АОИ
_____ Н.В. Пермякова

СОДЕРЖАНИЕ

| | |
|--|-----------|
| ВВЕДЕНИЕ..... | 4 |
| 1. ОБЩАЯ ХАРАКТЕРИСТИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ | 5 |
| 2. СТРУКТУРА САМОСТОЯТЕЛЬНОЙ РАБОТЫ | 5 |
| 3. МЕТОДИКА РЕАЛИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО ТЕОРЕТИЧЕСКОМУ КУРСУ | 7 |
| 4. СОДЕРЖАНИЕ РАЗДЕЛОВ И ТЕМ ДИСЦИПЛИНЫ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ | 8 |
| Раздел 2. Объектно-ориентированный анализ и проектирование с использованием UML. | 9 |
| Раздел 3. Диаграммы UML. | 9 |
| Раздел 4. Основы управления проектами | 11 |
| Раздел 5. Основные понятия объектно-ориентированного программирования..... | 12 |
| Раздел 6. Применение библиотек и иерархий объектов при программировании..... | 13 |
| ОЦЕНКА ВЫПОЛНЕНИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТА .. | 14 |
| РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА | 15 |

ВВЕДЕНИЕ

Дисциплина «Объектно-ориентированный анализ и программирование» относится к базовому циклу профессиональных дисциплин подготовки бакалавров направления 231000.62 «Программная инженерия».

Целью курса является сформировать у студентов объектно-ориентированное мышление, научить их объектно-ориентированному (ОО) подходу к анализу предметной области и использованию объектно-ориентированной методологии программирования при разработке программных продуктов.

Процесс самостоятельной работы студентов направлен на формирование следующих профессиональных (ПК) компетенций:

1) выпускник должен демонстрировать понимание основных концепций, принципов, теорий и фактов, связанных с информатикой (ПК-1).

2) выпускник должен уметь применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов (ПК-10).

3) выпускник должен иметь навыки использования различных технологий разработки программного обеспечения (ПК-16).

Задачи изучения дисциплины следующие:

- изучение техники объектно-ориентированного анализа;
- изучение приемов объектно-ориентированного программирования;
- изучение технологии проектирования архитектуры информационных систем;
- изучение основ проектирования информационно-коммуникационных технологий (ИКТ) и основ управления ИКТ-проектами.

Данные методические указания предназначены для выполнения самостоятельной работы по дисциплине «Объектно-ориентированный анализ и программирование» подготовки бакалавров направления 231000.62 «Программная инженерия».

1. ОБЩАЯ ХАРАКТЕРИСТИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ

Самостоятельная работа является важной составляющей в изучении дисциплины и состоит из следующих видов деятельности: проработка лекционного материала с целью подготовки к тестовым опросам и контрольным работам, самостоятельное изучение теоретического материала, подготовка к лабораторным работам, написание реферата и подготовка устного доклада с последующим выступлением на лекции.

Самостоятельная работа над теоретическим материалом направлена на изучение основных понятий и принципов объектно-ориентированного анализа и программирования, основ проектирования информационных систем, основ управления проектами.

Самостоятельная работа по подготовке к выполнению лабораторных работ направлена на изучение инструментальных сред объектно-ориентированного анализа и программирования. В ходе лабораторной работы студенты самостоятельно выполняют решение индивидуального или группового варианта, отлаживают написанные программы и защищают результаты работы.

На протяжении изучения дисциплины студент может выбрать тему, не вошедшую в лекционный курс и написать по выбранной теме реферат и (или) подготовить устный доклад для выступления на лекционных занятиях.

2. СТРУКТУРА САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Объем и виды самостоятельной работы в структуре дисциплины приведены в следующей таблице:

| Тематика самостоятельной работы (детализация) | Трудоемкость, час | Компетенции ОК, ПК | Контроль выполнения работы |
|--|-------------------|--------------------------------|-----------------------------------|
| Подготовка к лабораторной работе № 1 | 2 | ПК – 1, ПК – 10, ПК - 16 | Отчет, защита лабораторной работы |
| Проработка лекционного материала, подготовка к тестовым опросам: | | | |
| Подготовка к тестовому опросу по теме «Диаграмма состояний» | 1 | ПК – 1, ПК – 10, ПК - 16 | Тестовый опрос |
| Подготовка к тестовому опросу по теме «Диаграмма компонентов» | 1 | ПК – 1, ПК – 10, ПК - 16 | Тестовый опрос |
| Подготовка к тестовому опросу по теме «Диаграмма взаимодействия» | 1 | ПК – 1, ПК – 10, ПК - 16 | Тестовый опрос |

| | | | |
|--|---|--------------------------------|---------------------------|
| Самостоятельное изучение материала по теме «Диаграмма деятельности» | 3 | ПК – 1, ПК – 12, ПК - 16 | Тестовый опрос |
| Подготовка к тестовому опросу по теме «Диаграмма взаимодействия» | 1 | ПК – 1, ПК – 10, ПК - 16 | Тестовый опрос |
| Подготовка к лабораторной работе № 2 «Создание диаграммы прецедентов в Rational Rose» | 2 | ПК – 1, ПК – 10, ПК - 16 | Отчет, защита лаб. работы |
| Подготовка к лабораторной работе №3 «Создание диаграммы состояний в Rational Rose» | 3 | ПК – 1, ПК – 10, ПК - 16 | Отчет, защита лаб. работы |
| Подготовка к лабораторной работе №4 «Создание диаграмм взаимодействия в Rational Rose» | 3 | ПК – 1, ПК – 10, ПК - 16 | Отчет, защита лаб. работы |
| Подготовка к лабораторной работе №5 «Создание диаграммы классов в Rational Rose» | 2 | ПК – 1, ПК – 10, ПК - 16 | Отчет, защита лаб. работы |
| Подготовка и написание реферата | 1 | ПК – 10, ПК - 16 | Реферат |
| Подготовка устного доклада | 1 | ПК – 10, ПК - 16 | Выступление на лекции |
| Самостоятельное изучение тем: | | | |
| Этап анализа проблемы | 2 | ПК – 10, ПК - 16 | Контрольная работа №1 |
| Управление масштабом проекта. | 3 | ПК – 10, ПК - 16 | Контрольная работа №1 |
| Требования к программному обеспечению | 2 | ПК – 10, ПК - 16 | Контрольная работа №1 |
| Организация требований | 2 | ПК – 10, ПК - 16 | Контрольная работа №1 |
| Подготовка к контрольной работе №1 «Управление требованиями к ИС» | 1 | ПК – 10, ПК - 16 | Контрольная работа №1 |
| Подготовка и написание реферата | 6 | ПК – 10, ПК - 16 | Реферат |
| Подготовка устного доклада | 6 | ПК – 10, ПК - 16 | Выступление на лекции |
| Проработка лекционного материала, подготовка к кон- | | | |

| | | | |
|---|----|--------------------------------|---------------------------|
| трольным работам: | | | |
| «Разработка простого класса» | 1 | ПК – 1, ПК – 10, ПК - 16 | Контрольная работа №2 |
| «Разработка производного класса» | 1 | ПК – 1, ПК – 10, ПК - 16 | Контрольная работа №3 |
| «Отношения между классами» | 1 | ПК – 1, ПК – 10, ПК - 16 | Контрольная работа №4 |
| Подготовка к лабораторной работе «Работа с классом Vector» | 1 | ПК – 1, ПК – 10, ПК - 16 | Отчет, защита лаб. работы |
| Подготовка к лабораторной работе «Создание класса графического изображения» | 1 | ПК – 1, ПК – 10, ПК - 16 | Отчет, защита лаб. работы |
| Подготовка к лабораторной работе «Наследование и полиморфизм. Создание класса наследника» | 1 | ПК – 1, ПК – 10, ПК - 16 | Отчет, защита лаб. работы |
| Подготовка к лабораторной работе «Перегрузка операций» | 1 | ПК – 1, ПК – 10, ПК - 16 | Отчет, защита лаб. работы |
| Подготовка к лабораторной работе «Создание массива объектов. Однородные объекты. Разнородные объекты» | 1 | ПК – 1, ПК – 10, ПК - 16 | Отчет, защита лаб. работы |
| Подготовка к лабораторной работе «Создание класса для работы с файлами. Система меню» | 1 | ПК – 1, ПК – 10, ПК - 16 | Отчет, защита лаб. работы |
| Подготовка к лабораторной работе «Разработка и создание программной минисистемы» | 2 | ПК – 1, ПК – 10, ПК - 16 | Отчет, защита лаб. работы |
| Подготовка к экзамену | 36 | | Экзамен |

3. МЕТОДИКА РЕАЛИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО ТЕОРЕТИЧЕСКОМУ КУРСУ

При самостоятельном изучении элементов теоретического курса студенты могут придерживаться следующего плана:

1. Самостоятельно изучить темы теоретического курса в соответствии с учебной программой дисциплины:

- Диаграмма деятельности в UML;
- Этап анализа проблемы;

- Управление масштабом проекта;
- Требования к программному обеспечению;
- Организация требований.

2. Проработать лекционный материал с целью подготовки к следующим тестовым опросам:

- Диаграмма состояний;
- Диаграмма компонентов;
- Диаграмма взаимодействия;
- Диаграмма деятельности.

и контрольным работам:

- Управление требованиями к ИС;
- Разработка простого класса;
- Разработка производного класса;
- Отношения между классами.

3. Самостоятельно изучить одну из указанных ниже тем, оформить результаты изучения рефератом и (или) подготовить устный доклад:

- Интеграция пакетов Rational RequisitePro и Microsoft Project. Составление графика реализации проекта;
- Объектно-ориентированное проектирование Web-приложений с использованием языка унифицированного моделирования;
- Расширение языка UML для проектирования систем реального времени;
- Проектирование структуры баз данных с использованием компонента Data Modeler;
- Изучение пакета Rational Test;
- Изучение систем поддержки групповой работы на этапах проектирования и управления требованиями;
- Управление конфигурацией программных систем с использованием Rational ClearCase;
- Моделирование организационных систем с использованием UML;
- Адаптация Rational Unified Process к нуждам конкретной организации. Использование компонента RUP Builder.

4. Подготовить устные ответы на контрольные вопросы, приведенные после каждой темы.

Самостоятельную работу выполняют студенты на основе учебно-методических материалов дисциплины и литературы, указанной в списке литературы данного методического пособия.

Самостоятельно изучаемые вопросы курса включены в экзаменационные билеты.

Варианты тестовых опросов и контрольных работ преподаватель выдает на лекционных занятиях в соответствии с графиком проведения.

4. СОДЕРЖАНИЕ РАЗДЕЛОВ И ТЕМ ДИСЦИПЛИНЫ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Раздел 2. Объектно-ориентированный анализ и проектирование с использованием UML.

Трудоемкость лекционного курса – 2 часа.

Трудоемкость лабораторных работ – 2 часа.

Самостоятельная работа – 2 часа.

Язык UML. Основные средства анализа и моделирования предметной области в языке UML. предпосылки возникновения и история создания UML; принципы моделирования с использованием UML; результат моделирования, понятие архитектуры системы.

Контрольные вопросы:

1. Определите понятие «сложность задачи». Приведите примеры сложных систем.
2. Охарактеризуйте основные задачи объектно-ориентированной декомпозиции системы.
3. Перечислите основные этапы создания программного продукта.
4. Дайте определение атрибутов, их типов и представления при объектном моделировании.
5. Дайте определение связей, их видов и способы формализации.
6. Как используется язык UML для объектно-ориентированного анализа?
7. Перечислите Рабочие продукты ОО анализа.

Раздел 3. Диаграммы UML.

Трудоемкость лекционного курса – 5 часов.

Трудоемкость лабораторных работ – 16 часов.

Самостоятельная работа – 19 часов.

Диаграмма прецедентов. Назначение диаграммы прецедентов; понятие прецедента и актера; абстрактные актеры и прецеденты; примечания; отношения ассоциации, расширения, включения, обобщения; различия между отношениями включения и расширения; установка границ системы; прецеденты и функции; пропорции прецедентов и актеров на диаграмме; потоки событий; идеальные и реальные прецеденты; рекомендации по разработке диаграмм прецедентов.

Диаграммы состояний и деятельности. Общие сведения о диаграмме состояний; понятие автомата; состояние; метки действия, переход; событие, выражение действия, составные состояния и вложенные состояния; параллельные состояния; рекомендации по разработке диаграммы состояний. Идентификация, описание объектов и их поведения.

Общие сведения о диаграмме деятельности;

Диаграмма классов. Назначение диаграммы; понятие класса; классы и объекты; атрибуты класса; методы класса; абстрактные классы; основные виды отношений - отношения зависимости, ассоциации, обобщения, агрегации,

композиции; отношение зависимости между пакетами; рекомендации по построению диаграммы классов.

Диаграммы взаимодействий. Назначение диаграммы последовательности; объекты; линия жизни объекта; ветвление потока управления; стереотипы сообщений; временные ограничения; рекомендации по построению диаграмм.

Назначение диаграммы кооперации; понятие кооперации; активные объекты; составные объекты; связи; сообщения; рекомендации по построению. Объектный подход к разработке ПО для распределенных систем.

Диаграммы компонентов и развертывания. Назначение диаграммы компонентов; понятие компонента; виды компонентов; интерфейсы; зависимости; рекомендации по построению диаграммы компонентов. Назначение диаграммы развертывания; понятие узла; соединения; рекомендации по построению диаграммы.

Самостоятельно изучите тему «Диаграмма деятельности». При изучении обратите внимание на следующие ключевые моменты темы:

Состояние действия и состояние деятельности;
переход, ветвление, разделение и слияние;
параллельные потоки работ; дорожки, объекты;
рекомендации по построению диаграмм деятельности.

Контрольные вопросы:

1. Что определяет диаграмма прецедентов?
2. Дайте определение прецедента.
3. Дайте определение актера.
4. Перечислите основные типы отношений.
5. Дайте определение потока событий.
6. Что определяет диаграмма состояния?
7. Для чего используется диаграмма деятельности.
8. Дайте определение понятия «автомат».
9. Охарактеризуйте составные, вложенные и параллельные состояния.
10. Дайте определение объекта.
11. Что понимается под «поведением объекта»?
12. Дайте определение класса.
13. Перечислите типы атрибутов класса.
14. Для чего используются абстрактные классы?
15. Для чего используется диаграмма последовательности?
16. Что такое «линия жизни объекта».
17. Приведите примеры ветвления потока управления.
18. Определите понятие кооперации.
19. Приведите пример активных объектов.
20. Приведите пример составных объектов.
21. Определите понятие компонента.
22. Перечислите виды компонентов.
23. Определите понятия узла и соединения.

Раздел 4. Основы управления проектами

Трудоемкость лекционного курса – 1 часа.

Лабораторные работы по разделу не предусмотрены.

Самостоятельная работа – 22 часа.

Понятие требований; процесс управления требованиями; пирамида типов требований: запросы заинтересованных сторон, требования к программному обеспечению. Область проблемы и область решения. Экономическое обоснование необходимости формальных методов управления требованиями к ИС.

При *самостоятельном* изучении тем обратите внимание на следующие ключевые моменты:

1. Этап анализа проблемы -

Достижение соглашения об определении проблемы.

Выделение основных причин.

Выявление заинтересованных лиц и пользователей.

Определение границ системы-решения.

Выявление ограничений, налагаемых на решение.

2. Управление масштабом проекта -

Введение в проблему масштаба проекта.

Составляющие масштаба проекта.

Базовый уровень требований.

Установка приоритетов.

Оценка трудозатрат.

Добавление элемента риска.

Сокращение масштаба.

Функции системы и их атрибуты.

3. Требования к программному обеспечению –

Функциональные требования, функциональные и нефункциональные требования к программному обеспечению.

Требования практичности. Требования надежности.

Требования производительности.

Возможность обслуживания.

Ограничения проектирования.

4. Организация требований -

Использование "дочерних" требований для повышения уровня конкретизации.

Организация дочерних требований.

Пакет Modern SRS Package.

Документ видения. Понятие трассировки и верификации.

Качество требований и методы его проверки.

Контрольные вопросы:

1. Дайте определение понятия «требование».
2. Перечислите основные уровни требований.
3. Перечислите свойства требований к ИС.
4. Что понимается под процессом управления требованиями?
5. Что включает в себя процесс анализа проблемы?
6. Назовите способы визуализации процесса выделения основных причин.
7. Какими методами можно определить количественный вклад выделенных проблем в основную проблему?
8. Перечислите основные типы заинтересованных лиц.
9. Какие вопросы могут задаваться заинтересованным лицам на этапе анализа проблемы?
10. Дайте понятие актера системы.
11. Перечислите типы источников ограничений.
12. Перечислите виды требований к программному обеспечению по уровням.
13. Перечислите виды требований к программному обеспечению по характеру.
14. Каковы источники требований к программному обеспечению (ПО)?
15. Перечислите методологии разработки требований к ПО.
16. Дайте определение трассировки.
17. Дайте определение верификации.
18. Каким образом можно проверить качество требований?

Раздел 5. Основные понятия объектно-ориентированного программирования

Трудоемкость лекционного курса – 5 часов

Трудоемкость лабораторных работ – 10 часов

Самостоятельная работа – 8 часов.

Объектно-ориентированная методология программирования. Объектно-ориентированные языки. Понятия объекта, класса, метода, сообщения. Отношения простого и множественного наследования, виртуальные базовые классы. Абстракция данных, наследование и полиморфизм. Абстрактные классы. Полиморфизм параметрический и динамический. Совместимость типов в объектно-ориентированном программировании. Перекрывание методов. Методы реализации различных конструкций объектно-ориентированного программирования.

Контрольные вопросы:

1. Зачем нужны конструкторы и деструкторы классов?
2. Какие виды отношений существуют между классами?
3. Что представляют собой атрибуты и методы класса?

4. Каким образом можно управлять видимостью элементов класса?
5. Какие методы класса являются встроенными (inline)?
6. Какие спецификаторы доступа используются в C++?
7. Какие типы конструкторов существуют?
8. В каких случаях вызываются конструкторы копирования?
9. Какие конструкторы используются по умолчанию?
10. Для чего нужен деструктор? в каком случае вызывается деструктор?
11. На что указывает указатель this?
12. Какие операции нельзя перегружать?
13. Какой оператор используется при перегрузке операций?
14. Какие способы существуют для перегрузки операций?
15. Чем отличается перегрузка компонентной и глобальной унарной операции?
16. Что обозначает спецификатор friend?
17. Что представляет собой отношение наследования?
18. Что представляет собой открытое наследование? Привести пример.
19. Что такое принцип подстановки. Привести пример.
20. Что представляет собой закрытое наследование? Привести пример.
21. Что такое связывание?
22. Что такое полиморфные функции?
23. Чем полиморфизм отличается от принципа подстановки?
24. В каких случаях используется механизм позднего связывания?
25. Что такое чисто виртуальная функция?
26. Какие методы рекомендуется делать виртуальными?
27. Что такое таблица виртуальных методов? Для чего она используется?
28. Каким образом выполняется выбор виртуальной функции при позднем связывании?
29. Почему деструкторы рекомендуется делать виртуальными?
30. Какие методы нужно делать виртуальными?

Раздел 6. Применение библиотек и иерархий объектов при программировании

Трудоемкость лекционного курса – 4 часа

Трудоемкость лабораторных работ – 8 часов

Самостоятельная работа – 3 часа.

Потоки. Установка потока. Чтение и запись потока. Вывод в поток. Ввод из потока. Удаление потока. Использование объектов с потоком. Механизм потоков. Процедуры обмена информации в потоках. Шаблоны. Библиотека стандартных шаблонов. Основные концепции STL. Последовательные и ассоциативные контейнеры. Итераторы STL. Общие свойства контейнеров. Использование последовательных контейнеров. Адаптеры контейнеров. Алгоритмы. Инструментальные средства объектно-ориентированного программирования.

Контрольные вопросы:

1. Абстрактные типы данных. Контейнеры.
2. Примеры реализации (вектор, матрица, стек, очередь)
3. Создание шаблонов классов
4. Использование шаблонов классов
5. Специализация шаблонов.
6. Классификация потоков.
7. Подключение потоков
8. Операции ввода-вывода
9. Файловые потоки
10. Контейнеры
11. Итераторы
12. Последовательные контейнеры
13. Адаптеры контейнеров
14. Ассоциативные контейнеры
15. Обобщенные алгоритмы

ОЦЕНКА ВЫПОЛНЕНИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТА

Самостоятельная работа студента оценивается преподавателем по результатам выполнения:

- тестовых опросов;
- контрольных работ;
- защиты лабораторных работ;
- устного доклада;
- реферата;
- ответов на экзамене.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Павловская Т. А. С/С++. Программирование на языке высокого уровня : Учебник для вузов / Т. А. Павловская. - СПб.: Питер, 2007.- 460 с
2. Крачтен Филипп. Введение в Rational Unified Process. 2-е изд.. : Пер. с англ. — М: Издательский дом “Вильямс”, 2002. — 240 с. : ил.
3. Леоненков А. В. Самоучитель UML. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2006. - 432 с.
4. Буч Грейди, Рамбо Джеймс, Джекобсон Айвар. Язык UML. Руководство пользователя: Пер. с англ. - М.: ДМК,2000. - 432с.
5. Розенберг Дуг, Кендалл Скот. Применение объектного моделирования с использованием UML и анализ прецедентов. Пер. с англ. – М.: “ДМК-Пресс”, 2002. – 160 с.
6. Ларман Крэг. Применение UML и шаблонов проектирования. Введение в объектно-ориентированный анализ и проектирование. Пер. с англ.: Уч. Пос. — М: Издательский дом “Вильямс”, 2001. — 496 с.
7. Боггс Уэнди, Боггс Майкл. UML и Rational Rose. Пер. с англ. — М: Издательство “Лори”, 2000. — 582 с.: ил.
8. Леффингуэл Дин, Уидриг Дон. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. Пер. с англ. — М: Издательский дом “Вильямс”, 2002. — 448 с.
9. Брайен А. Уайт. Управление конфигурацией программных средств. Практическое руководство по Rational ClearCase. Пер. с англ. – М.: “ДМК-Пресс”, 2002. – 272 с.
10. Коналлен Джим. Разработка Web-приложений с использованием UML. Пер. с англ. – М.: Издательский дом “Вильямс”, 2001. – 288 с.