

Правительство Российской Федерации

**Федеральное государственное автономное образовательное учреждение
высшего профессионального образования**

**«Национальный исследовательский университет
«Высшая школа экономики»**

**Факультет БИЗНЕС-ИНФОРМАТИКИ
Отделение ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

Программа дисциплины

Объектно-ориентированные CASE-технологии

**для направления 010400.62 «Прикладная математика и информатика»
подготовки бакалавров**

Автор Незнанов А.А. (aneznanov@hse.ru)

Рекомендована секцией УМС
«Прикладная математика
и информатика»

Председатель
_____ Кузнецов С.О.
« ____ » _____ 20 ____ г.

Одобрена на заседании кафедры
Анализа данных
и искусственного интеллекта

Зав. кафедрой
_____ Кузнецов С.О.
« ____ » _____ 20 ____ г.

Утверждена УС факультета
бизнес-информатики

Ученый секретарь
_____ Фомичев В.А.
« ____ » _____ 20 ____ г.

Москва

Программа дисциплины Объектно-ориентированные CASE-технологии
(Технологии разработки сложных программных систем) для подготовки бакалавров по направлению 010500.62 (бакалаврская программа «Прикладная математика и информатика»)

I. Пояснительная записка

Автор программы: кандидат технических наук А.А. Незнанов

Требования к студентам: Изучение курса «Объектно-ориентированные CASE-технологии» («Технологии разработки сложных программных комплексов») требует предварительных знаний по основам информатики (курс «Информатика»), объектно-ориентированному конструированию (курс «Практикум на ЭВМ»).

Аннотация. Дисциплина «Объектно-ориентированные CASE-технологии» предназначена для подготовки бакалавров 010500.62 – Прикладная математика и информатика.

Любой специалист, занятый в сфере информационных технологий, должен отчётливо представлять себе общие тенденции развития отрасли и современное состояние технологий разработки *программных продуктов* (ПО). Основным понятием, вокруг которого строится теория промышленного создания программных продуктов, является понятие «*жизненного цикла программного обеспечения* (ЖЦПО)». В настоящее время наиболее популярна объектно-ориентированная парадигма проектирования и конструирования программного обеспечения. В курсе рассматриваются основные стадии, этапы и процессы ЖЦПО; поддерживающие их объектно-ориентированные методологии; программные средства на основе платфомерно-ориентированного подхода.

Учебные задачи курса.

Данный курс позволит студентам понять принципы, положенные в основу современной программной инженерии, и овладеть приёмами работы с основными классами средств поддержки ЖЦПО. Студенты получат навыки объектно-ориентированного анализа, проектирования и конструирования ПО с использованием объектно-ориентированного подхода, осознанного выбора платформы разработки и развёртывания ПО, использования средств поддержки ЖЦПО.

II. Тематический план курса «Объектно-ориентированные CASE-технологии»

№	Название темы	Всего часов по дисциплине	Аудиторные часы		Самостоятельная работа
			Лекции	Сем. и практика	
1	Введение. Жизненный цикл программного обеспечения. Промышленное производство программных продуктов. Программные платформы.	18	4	2	12
2	Классификация ПО. Основные процессы ЖЦ ПО. Стандартизация ЖЦ ПО.	22	4	4	14
3	Объектно-ориентированный анализ и проектирование. Язык UML. Средства поддержки объектно-ориентированного анализа и проектирования.	46	8	8	30
4	Интегрированные системы поддержки ЖЦ ПО. Организация удалённой работы. Системы управления конфигурацией.	28	4	4	20
5	Средства поддержки процессов конструирования.	26	4	4	18
6	Средства поддержки процессов развёртывания и сопровождения.	22	2	4	16
	Итого	162	26	26	110

III. Источники информации

Базовый учебник по курсу – ридер «Технологии разработки сложных программных систем», составленный по следующим источникам:

1. Эрик Дж. Брауде. Технология разработки программного обеспечения. – Питер, 2004. – 656 с.
2. Роберт Т. Фатрелл, Дональд Ф. Шафер, Линда И. Шафер. Управление программными проектами. Достижение оптимального качества при минимуме затрат. – Вильямс, 2003. – 1136 с.
3. *Guide to the Software Engineering Body of Knowledge (SWEBOK)* (<http://www.computer.org/portal/web/swebok>) и его русский перевод (<http://swebok.sorlik.ru>)
4. С. А. Орлов. Технологии разработки программного обеспечения. Разработка сложных программных систем. Для студентов и преподавателей высших учебных заведений. – Питер, 2003. – 480 с.
5. Ф. Брукс. Мифический человеко-месяц или Как создаются программные системы. – Символ-Плюс, 2000. – 304 с.

6. С. Макконнелл. Совершенный код. Практическое руководство по разработке программного обеспечения. – Питер, 2005. – 896 с.
7. С. Макконнелл. Сколько стоит программный проект. – СПб.: Питер, 2007. – 297 с.
8. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. – М. : Бинум, 1998. – 2-е изд. – 560 с.
9. А. Якобсон, Г. Буч, Дж. Рамбо. Унифицированный процесс разработки программного обеспечения. – Питер, 2002. – 496 с.
10. Арлоу Дж. и Нейштадт А. UML 2 и Унифицированный процесс: практический объектно-ориентированный анализ и проектирование. – М. : Символ-Плюс, 2007. – 624 с.
11. Рамбо Дж. и Блаха М. UML 2.0. Объектно-ориентированное моделирование и разработка. – СПб : Питер, 2006. – 544 с.
12. Карл И. Вигерс. Разработка требований к программному обеспечению. – Русская Редакция, 2004. – 576 с.
13. Роберт К. Мартин, Джеймс В. Ньюкирк, Роберт С. Косс. Быстрая разработка программ. Принципы, примеры, практика. – Вильямс, 2004. – 752 с.
14. Э. Йордон. Путь камикадзе. Как разработчику программного обеспечения выжить в безнадежном проекте. – Лори, 2005. – 264 с.
15. Джю Ханк Рейнвотер. Как пасти котов. Наставление для программистов, руководящих другими программистами. – СПб.: Питер, 2006. – 256 с.
16. Кент Бек, Мартин Фаулер. Экстремальное программирование: планирование. – Питер, 2003. – 144 с.
17. Кент Бек. Экстремальное программирование. – Питер, 2002. – 224 с.
18. Кент Бек. Экстремальное программирование: разработка через тестирование. – Питер, 2003. – 224 с.
19. Кент Ауэр, Рой Миллер. Экстремальное программирование: постановка процесса. С первых шагов и до победного конца. – Питер, 2003 г. – 368 с.

Дополнительная литература по курсу

1. Сполски Д. Джоэл о программировании. – М. : Символ-Плюс, 2006. – 352 с.
2. Эндрю Хант, Дэвид Томас. Программист-практик. – М.: Лори. – 270с.
3. Фаулер М., Бек К., Брант Д., Робертс Д., Апдайк У. Рефакторинг: улучшение существующего кода. – Символ-Плюс, 2002. – 432 с.
4. Антон Элиенс. Принципы объектно-ориентированной разработки программ. – Вильямс, 2002 г. – 496 с.
5. Бертран Мейер. Объектно-ориентированное конструирование программных систем. – Русская Редакция, 2005. – 1204 с.
6. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приёмы объектно-ориентированного проектирования. Паттерны проектирования. – Питер, 2001. – 368 с.
7. Савин Р. Тестирование Дот Ком, или Пособие по жестокому обращению с багами в интернет-стартапах. – Издательство: Дело, 2007. – 312 с.
8. Роберт У. Себеста. Основные концепции языков программирования, 5 издание. – Вильямс, 2001. – 672 с.
9. В. В. Липаев. Выбор и оценивание характеристик качества программных средств. Методы и стандарты. – Синтег, 2001. – 228 с.
10. В. В. Липаев. Документирование и управление конфигурацией программных средств. Методы и стандарты. – Синтег, 1998. – 220 с.
11. В. В. Липаев. Системное проектирование сложных программных средств для информационных систем. – Синтег, 2002. – 268 с.
12. В. В. Липаев. Надежность программных средств. – Синтег, 1998. – 232 с.
13. Мэтт Тэллес, Юань Хсих. Наука отладки. – КУДИЦ-Образ, 2003. – 560 с.

14. Калбертсон Р., Браун К., Кобб Г. Быстрое тестирование. – Вильямс, 2002. – 384 с.
15. Джон Макгрегор, Дэвид Сайкс. Тестирование объектно-ориентированного программного обеспечения. Практическое пособие. – ТИД «ДС», 2002. – 432 с.
16. Л. Басс, П. Клементс, Р. Кацман. Архитектура программного обеспечения на практике. – Питер, 2006. – 576 с.
17. Сэм Канер, Джек Фолк, Енг Кек Нгуен. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. – ДиаСофт, 2001. – 544 с.
18. Б. Керниган, Р. Пайк. Практика программирования. – Невский Диалект, 2001. – 384 с.
19. Открытые стандарты и спецификации. – Общественная организация «Нетлэндер», 2008. (<http://doka.info/recommendations>)
20. Jordan Russell. Inno Setup (<http://www.jrsoftware.org/isinfo.php>)
21. Microsoft Windows Installer ([http://msdn.microsoft.com/en-us/library/cc185688\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/cc185688(VS.85).aspx))

IV. Формы контроля и структура итоговой оценки

Текущий контроль – 1 письменная контрольная работа (60 мин).

Итоговый контроль – 1 зачёт (120 мин.)

Итоговая оценка складывается из следующих элементов:

- работа на семинарах – 20%;
- устный зачет – 30%;
- 1 письменная контрольная работа – 20%;
- домашнее задание – 30%;

Таблица соответствия оценок по десятибалльной и системе зачет/незачет

Оценка по 10-балльной шкале	Оценка по 5-балльной шкале
1	незачет
2	
3	
4	зачет
5	
6	
7	
8	
9	
10	

Таблица соответствия оценок по десятибалльной и пятибалльной системе

По десятибалльной шкале	По пятибалльной системе
1 – неудовлетворительно	неудовлетворительно – 2
2 – очень плохо	
3 – плохо	
4 – удовлетворительно	удовлетворительно – 3
5 – весьма удовлетворительно	
6 – хорошо	хорошо – 4
7 – очень хорошо	
8 – почти отлично	отлично - 5
9 – отлично	
10 - блестяще	

V. Программа курса «Объектно-ориентированные CASE-технологии»

Тема 1. Введение. Жизненный цикл программного обеспечения. Промышленное производство программных продуктов. Программные платформы.

Инженерия программного обеспечения. Жизненный цикл программного обеспечения (ЖЦ ПО, SLCP) как объект исследования. ПО как система. ПО как проект. Процесс, задача, действие, фаза, стадия, этап, компетенция и др. Основные артефакты ЖЦ ПО, примеры.

Этапы ЖЦ: от анализа концепции до уничтожения. Итеративная природа каждого этапа ЖЦ. Процессы ЖЦ: от сбора требований до сопровождения.

Понимание ролей и интересов заказчика и исполнителя. Разделение ролей. Администраторы проекта, архитекторы, системные аналитики, аналитики предметной области, ведущие программисты, кодеры, проектировщики взаимодействия с пользователем, дизайнеры, тестеры, специалисты по внедрению и прочие.

Инфраструктура программной инженерии. Платформы разработки и развертывания. Основные современные платформы.

Основная литература

1. Эрик Дж. Брауде. Технология разработки программного обеспечения. – Питер, 2004. – 656 с.
2. Роберт Т. Фатрелл, Дональд Ф. Шафер, Линда И. Шафер. Управление программными проектами. Достижение оптимального качества при минимуме затрат. – Вильямс, 2003. – 1136 с.
3. С. А. Орлов. Технологии разработки программного обеспечения. Разработка сложных программных систем. Для студентов и преподавателей высших учебных заведений. – Питер, 2003. – 480 с.
4. Ф. Брукс. Мифический человеко-месяц или Как создаются программные системы. – Символ-Плюс, 2000. – 304 с.
5. *Guide to the Software Engineering Body of Knowledge (SWEBOK)* (<http://www.computer.org/portal/web/swebok>) и его русский перевод (<http://swebok.sorlik.ru>)

Дополнительная литература

6. С. Макконнелл. Сколько стоит программный проект. – СПб.: Питер, 2007. – 297 с.
7. Э. Йордон. Путь камикадзе. Как разработчику программного обеспечения выжить в безнадежном проекте. – Лори, 2005. – 264 с.
8. Открытые стандарты и спецификации. – Общественная организация «Нетлэндер», 2008. (<http://doka.info/recommendations>)

Тема 2. Классификация ПО. Основные процессы ЖЦ ПО. Стандартизация ЖЦ ПО.

Классификация ПО в соответствии с *ISO 12182-2002* и в зависимости от параметров ЖЦ.

Этапы ЖЦ. Исследование задачи и предметной области. Сбор требований. Подтверждение задачи. Подготовка ТЗ. Высокоуровневое (логическое) проектирование и низкоуровневое (физическое) проектирование. Уточнение ТЗ. Конструирование. Внутреннее тестирование. Развёртывание. Внедрение. Сопровождение/поддержка. Эволюция или уничтожение.

Конкретные процессы ЖЦ в различных методологиях. Управление сбором требований, управление конфигурацией и изменениями, управление профилированием, отладкой и тестированием, управление сопровождением, управление рисками и др.

Модели жизненного цикла. Линейная модель. Каскадная модель. V-образная модель. Итерационные модели и эволюционное прототипирование. Инкрементная модель. Спиральная модель. Быстрая разработка приложений (*RAD*). Унифицированный процесс (*RUP*). Разработка в стиле экстремального программирования (*XP*). Адаптация моделей ЖЦ. Стандартизация ЖЦ ПО. Модель *CMM SEI*. Стандарт *IEEE 1074. ISO/IEC 12207. Mil STD 498. SWEBoK*.

Методологии управления ЖЦ ПО. Следствия из современных методологий: версии и редакции ПО, особенности сборки ПО.

Основная литература

1. Эрик Дж. Брауде. Технология разработки программного обеспечения. – Питер, 2004. – 656 с.
2. Роберт Т. Фатрелл, Дональд Ф. Шафер, Линда И. Шафер. Управление программными проектами. Достижение оптимального качества при минимуме затрат. – Вильямс, 2003. – 1136 с.
3. С. А. Орлов. Технологии разработки программного обеспечения. Разработка сложных программных систем. Для студентов и преподавателей высших учебных заведений. – Питер, 2003. – 480 с.
4. *Guide to the Software Engineering Body of Knowledge (SWEBOK)* (<http://www.computer.org/portal/web/swebok>) и его русский перевод (<http://swebok.sorlik.ru>)

Дополнительная литература

5. Открытые стандарты и спецификации. – Общественная организация «Нетлэндер», 2008. (<http://doka.info/recommendations>)

Тема 3. Объектно-ориентированный анализ и проектирование. Язык UML. Средства поддержки объектно-ориентированного анализа и проектирования.

Объектно-ориентированный анализ и проектирование. Парадигмы и подходы. Мир классов и объектов. Границы предметной области.

Сбор требований. Общая схема построения концептуальных, логических и физических моделей. Язык *Unified Modeling Language (UML)*. Базовые понятия *UML*. История создания. Основные виды диаграмм, различие между статическими и динамическими диаграммами. Выразительная сила диаграмм, общие правила их построения, используемые графические примитивы.

Сценарии использования и начало проектирования. Статические диаграммы: классов, объектов, пакетов, отношения между классами и объектами. Обозначение множественности и модальности. Динамические диаграммы: деятельности, состояний, синхронизации и др.

Оптимизация диаграмм и их автоматическая прорисовка.

Программные средства, поддерживающие язык *UML*. Прямая и обратная разработка.

Основная литература

1. Арлоу Дж. и Нейштадт А. *UML 2* и Унифицированный процесс: практический объектно-ориентированный анализ и проектирование. – М. : Символ-Плюс, 2007. – 624 с.
2. Рамбо Дж. и Блаха М. *UML 2.0*. Объектно-ориентированное моделирование и разработка. – СПб : Питер, 2006. – 544 с.
3. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. – М. : Бином, 1998. – 2-е изд. – 560 с.
4. Карл И. Вигерс. Разработка требований к программному обеспечению. – Русская Редакция, 2004. – 576 с.

Дополнительная литература

5. Роберт К. Мартин, Джеймс В. Ньюкирк, Роберт С. Косс. Быстрая разработка программ. Принципы, примеры, практика. – Вильямс, 2004. – 752 с.
6. В. В. Липаев. Выбор и оценивание характеристик качества программных средств. Методы и стандарты. – Синтег, 2001. – 228 с.
7. Открытые стандарты и спецификации. – Общественная организация «Нетлэндер», 2008. (<http://doka.info/recommendations>)
8. *UML® Resource Page* (Официальные стандарты на язык *UML*). – Object Management Group, Inc., 2008. (<http://www.uml.org/>)
9. *Allen Holub's UML Quick Reference*. – Allen Holub, 2007. (<http://www.holub.com/goodies/uml/>)

Тема 4. Интегрированные системы поддержки ЖЦ ПО. Организация удалённой работы. Системы управления конфигурацией.

Связь показателей качества ПО, качества процессов ЖЦ ПО, методологии и выбранных средств поддержки ЖЦ ПО. Принципы интеграции средств поддержки. Интегрированные среды разработки ПО (*IDE*). Ролевая интеграция. Проблемы аутсорсинга. Проблемы разработки ПО в гетерогенных средах.

Базовые средства поддержки ЖЦ ПО: системы управления конфигурацией – *software configuration management (SCM)*.

Современные системы контроля версий. Централизованные и распределённые системы. История развития. *CVS, SVN, SourceSafe, ClearCase, Git, Mercurial* и др. Базовая методология работы с артефактами. Ветвление и тегирование.

Управление информацией об ошибках и пожеланиях.

Основная литература

1. Эрик Дж. Брауде. Технология разработки программного обеспечения. – Питер, 2004. – 656 с.
2. Роберт Т. Фатрелл, Дональд Ф. Шафер, Линда И. Шафер. Управление программными проектами. Достижение оптимального качества при минимуме затрат. – Вильямс, 2003. – 1136 с.
3. С. А. Орлов. Технологии разработки программного обеспечения. Разработка сложных программных систем. Для студентов и преподавателей высших учебных заведений. – Питер, 2003. – 480 с.

Дополнительная литература

4. Открытые стандарты и спецификации. – Общественная организация «Нетлэндер», 2008. (<http://doka.info/recommendations>)

Тема 5. Поддержка процессов конструирования.

Среды конструирования ПО. Управление исходным кодом. Версионный контроль кодовой базы.

Алгоритмизация, кодирование, отладка и сборка.

Модульное, интеграционное и функциональное тестирование.

Рефакторинг объектно-ориентированного кода. Цели и условия рефакторинга. Основные методы рефакторинга. Примеры.

Профилирование и оптимизация кода.

Реализация поддержки обсуждаемых процессов в современных интегрированных средах конструирования ПО.

Основная литература

1. С. Макконнелл. Совершенный код. Практическое руководство по разработке программного обеспечения. – Питер, 2005. – 896 с.

2. Фаулер М., Бек К., Брант Д., Робертс Д., Апдайк У. Рефакторинг: улучшение существующего кода. – Символ-Плюс, 2002. – 432 с.
3. Роберт Т. Фатрелл, Дональд Ф. Шафер, Линда И. Шафер. Управление программными проектами. Достижение оптимального качества при минимуме затрат. – Вильямс, 2003. – 1136 с.
4. С. А. Орлов. Технологии разработки программного обеспечения. Разработка сложных программных систем. Для студентов и преподавателей высших учебных заведений. – Питер, 2003. – 480 с.
5. Бертран Мейер. Объектно-ориентированное конструирование программных систем. – Русская Редакция, 2005. – 1204 с.
6. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приёмы объектно-ориентированного проектирования. Паттерны проектирования. – Питер, 2001. – 368 с.
7. Сэм Канер, Джек Фолк, Енг Кек Нгуен. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. – ДиаСофт, 2001. – 544 с.
8. В. В. Липаев. Надежность программных средств. – Синтег, 1998. – 232 с.
9. Мэтт Тэллес, Юань Хсих. Наука отладки. – КУДИЦ-Образ, 2003. – 560 с.

Дополнительная литература

10. Сполски Д. Джоэл о программировании. – М. : Символ-Плюс, 2006. – 352 с.
11. Эндрю Хант, Дэвид Томас. Программист-практик. – М.: Лори. – 270с.
12. Роберт У. Себеста. Основные концепции языков программирования, 5 издание. – Вильямс, 2001. – 672 с.
13. Открытые стандарты и спецификации. – Общественная организация «Нетлэндер», 2008. (<http://doka.info/recommendations>)

Тема 6. Поддержка процессов развёртывания и сопровождения.

Дистрибуция ПО. Подробнее о сборке, версиях и редакциях ПО. Основные каналы дистрибуции. Дистрибуция в сети Интернет.

Построение дистрибутивов. Дистрибутивы и операционные системы. Обеспечение кроссплатформенности. Управление развёртыванием.

Написание руководства пользователя и администратора.

Сопровождение и поддержка: взгляд с двух сторон. Управление информацией об ошибках и пожеланиях. Особенности сопровождения ПП различных классов.

Основная литература

1. Эрик Дж. Брауде. Технология разработки программного обеспечения. – Питер, 2004. – 656 с.
2. Роберт Т. Фатрелл, Дональд Ф. Шафер, Линда И. Шафер. Управление программными проектами. Достижение оптимального качества при минимуме затрат. – Вильямс, 2003. – 1136 с.
3. С. А. Орлов. Технологии разработки программного обеспечения. Разработка сложных программных систем. Для студентов и преподавателей высших учебных заведений. – Питер, 2003. – 480 с.
4. Jordan Russell. Inno Setup (<http://www.jrsoftware.org/isinfo.php>)
5. Microsoft Windows Installer ([http://msdn.microsoft.com/en-us/library/cc185688\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/cc185688(VS.85).aspx))

Дополнительная литература

6. Сполски Д. Джоэл о программировании. – М. : Символ-Плюс, 2006. – 352 с.
7. Открытые стандарты и спецификации. – Общественная организация «Нетлэндер», 2008. (<http://doka.info/recommendations>)

VI. Тематика заданий по различным формам текущего контроля

Темы контрольных работ, зачётов:

1. Базовые понятия инженерии ПО.
2. Классификация ПО (по *ISO 12182* и *SEI*).
3. Роли участников и процессы ЖЦ ПО.
4. Модели ЖЦ ПО, их достоинства и недостатки.
5. Классификация средств поддержки ЖЦ ПО.
6. Базовые понятия языка *UML 2.0*.
7. Правила построения *UML*-диаграмм.
8. Процесс *UML*-проектирования.
9. Оценка относительных затрат на этапы и процессы ЖЦ ПО.
10. Технология быстрой разработки ПО (*RAD*).
11. Интегрированные системы поддержки ЖЦ ПО.
12. Сравнение распространённых платформ разработки и развёртывания.
13. Принципы работы с системами версионного контроля.
14. Принципы работы с системами управления требованиями.
15. Принципы работы с системами объектно-ориентированного анализа и проектирования.
16. Принципы работы с системами автоматизированного тестирования.
17. Принципы работы с системами профилирования и оптимизации.
18. Принципы работы с системами построения дистрибутивов.

Темы домашних работ:

1. Объектно-ориентированный анализ и проектирование в выбранной предметной области с использованием нотации *UML 2.0*.
2. Правильное использование средств поддержки ЖЦ ПО в рамках заданного технологического процесса и методологии при коллективной разработке ПО.

VII. Вопросы для оценки качества освоения дисциплины

Тема 1.

1. В чём особенность создания ПО как инженерной деятельности?
2. Что такое жизненный цикл программного обеспечения?
3. В чём различие понятий «программный продукт» и «программное обеспечение»?
4. По каким критериям классифицируются программные продукты?
5. Чем системное ПО отличается от прикладного?
6. Как оплачиваются ПП? Что такое свободное ПО и в чём его отличия от бесплатного?
7. Какие артефакты возникают в результате процессов ЖЦПО?
8. Назовите основные роли участников ЖЦПО.
9. Что такое платформа? Чем платформа разработки отличается от платформы развёртывания?

Тема 2.

10. Что такое модель жизненного цикла?
11. Каковы основные этапы, процессы, роли участников ЖЦПО?
12. Как связан объём разработки с выбором модели ЖЦПО?
13. Какими стандартами и соглашениями регламентируются ЖЦПО?
14. Где проходит граница между анализом и проектированием?
15. В чём особенности объектно-ориентированного подхода к анализу, проектированию и конструированию ПО?

16. Охарактеризуйте линейную модель ЖЦПО.
17. Охарактеризуйте V-образную модель ЖЦПО.
18. Что такое инкрементная разработка?
19. Охарактеризуйте спиральную модель ЖЦПО.
20. Что такое быстрая разработка приложений?

Тема 3.

21. Дайте общую характеристику процесса объектно-ориентированного проектирования.
22. Как выявляются необходимые сущности и связи между ними?
23. Что такое требование? Каковы его основные характеристики?
24. Дайте характеристику языку *UML*. Каковы его главные достоинства и недостатки?
25. Какие виды диаграмм специфицирует язык *UML*?
26. Какие виды отношений между объектами способен выразить язык *UML*?
27. Опишите назначение и правила составления *UML*-диаграмм.
28. Какие виды ассоциаций можно представить на языке *UML*?
29. С какого вида диаграмм нужно начинать описание модели?
30. Приведите пример тернарной ассоциации.
31. Какие программные средства используются для упрощения процессов объектно-ориентированного анализа, проектирования и конструирования?
32. Как объектно-ориентированный подход стыкуется с различными моделями данных при создании схем баз данных?
33. Что такое объектно-реляционное отображение?
34. Что такое обратный инжиниринг?
35. Охарактеризуйте отличия концептуального, логического и физического проектирования.
36. Когда требуется формально описанные требования к ПО?
37. Что специфицирует техническое задание?
38. Как техническое задание связано с проведением приёмо-сдаточных испытаний?
39. Какие методы применяются при сборе требований?
40. Чем отличаются друг от друга интервью, тестирование и анкетирование?
41. Как связана методология проектирования с поддерживающими её программными средствами?

Тема 4.

42. Каковы основные характеристики интегрированных сред разработки ПО?
43. В чём особенности многопользовательской работы со средствами поддержки ЖЦ ПО?
44. В чём заключается сложность поддержки ЖЦ гетерогенных программных систем?
45. Что такое аутсорсинг и аутстаффинг?
- 46.

Тема 5.

47. Перечислите основные возможности систем управления исходным кодом (версионного контроля).
48. Как правильно сравнивать изменения в исходный код?
49. В чём заключаются основные проблемы слияния версий?
50. Перечислите основные возможности систем профилирования.
51. Дайте обзор уровней оптимизации.
52. Когда нужно начинать оптимизацию, а когда нельзя её заниматься?
53. Что такое рефакторинг исходного кода?
54. Как формализуется надёжность программной системы?
55. В чём главная проблема функционального тестирования?
56. Что такое интеграционное тестирование?
57. Имеет ли смысл заниматься оптимизацией без профилирования?

Тема 6.

58. Что является целью процесса развёртывания?

59. Чем отличаются понятия поддержки и сопровождения?
60. Что такое эволюция программного продукта?
61. Перечислите основные возможности систем построения дистрибутивов.
62. Как связаны процессы сборки и развёртывания?
63. Как повышается сложность процессов сборки с ростом числа версий продуктов?

Автор программы: _____ / А.А. Незнанов /