

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра вычислительной техники

“УТВЕРЖДАЮ”

ДЕКАН АВТФ

к.т.н. Рева И. Л.

“ ____ ” _____ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Проектирование программных систем

Образовательная программа: 09.03.04 Программная инженерия, профиль: Технологии разработки программного обеспечения

Курс: 3, семестр : 5

Факультет автоматики и вычислительной техники,

		Семестр
№	Вид деятельности	5
1	Всего зачетных единиц (кредитов)	3
2	Всего часов	108
3	Всего занятий в контактной форме, час.	45
4	Лекции, час.	0
5	Практические занятия, час.	18
6	Лабораторные занятия, час.	18
7	из них в активной и интерактивной форме, час.	8
8	Аттестация, час.	2
9	Консультации, час.	7
10	Самостоятельная работа, час.	63
11	Виды самостоятельной работы (курсовой проект, курсовая работа, РГЗ, подготовка к контрольной работе)	РГЗ
12	Вид аттестации	ДЗ

Рабочая программа составлена на основании федерального государственного образовательного стандарта высшего образования по направлению (специальности): 09.03.04 Программная инженерия

ФГОС введен в действие приказом №229 от 12.03.2015 г. , дата утверждения: 01.04.2015 г.

Место дисциплины в структуре учебного плана: Блок 1, вариативная

Рабочая программа разработана на основе компетентностной модели выпускника по направлению (специальности): 09.03.04 Программная инженерия

Рабочая программа обсуждена на заседании кафедры

ВТ, протокол заседания кафедры №6 от 20.06.2017

Утверждена на совете факультета автоматики и вычислительной техники, протокол № 6 от 21.06.2017

Программу разработал:

доцент, к.т.н. Романов Е. Л.

Заведующий кафедрой:

доцент, к.т.н. Якименко А. А.

Ответственный за образовательную программу:

доцент Романов Е. Л.

1. Внешние требования

Таблица 1.1

Компетенция ФГОС: ОПК.3 готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов; в части следующих результатов обучения:	
у1. владеть объектно-ориентированным подходом к программированию	
Компетенция ФГОС: ПК.1 готовность применять основные методы и инструменты разработки программного обеспечения; в части следующих результатов обучения:	
з1. знать методы и средства проектирования программного обеспечения	
з2. знать основные методы измерения и оценки характеристик программного обеспечения	
з3. знать принципы построения архитектуры программного обеспечения и виды архитектур программного обеспечения	
з5. знать типовые метрики программного обеспечения	
з6. знать возможности существующей программно-технической архитектуры	
у5. уметь использовать возможности имеющейся технической и/или программной архитектуры	
Компетенция ФГОС: ПК.14 готовность обосновать принимаемые проектные решения, осуществлять постановку и выполнение экспериментов по проверке их корректности и эффективности; в части следующих результатов обучения:	
у7. уметь проводить оценку и обоснование рекомендуемых решений	
Компетенция ФГОС: ПК.16 способность формализовать предметную область программного проекта и разработать спецификации для компонентов программного продукта; в части следующих результатов обучения:	
з1. знать языки формализации функциональных спецификаций	
у1. уметь разрабатывать технические спецификации на программные компоненты и их взаимодействие	
Компетенция ФГОС: ПК.19 владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения; в части следующих результатов обучения:	
з5. знать нотации моделирования	
у5. уметь моделировать бизнес-процессы	
у6. уметь применять соглашение о моделировании	
Компетенция ФГОС: ПК.22 способность создавать программные интерфейсы; в части следующих результатов обучения:	
з6. знать методы и средства проектирования программных интерфейсов	
Компетенция ФГОС: ПК.5 владение стандартами и моделями жизненного цикла; в части следующих результатов обучения:	
з1. знать методологии разработки программного обеспечения	
Компетенция ФГОС: ПК.7 владение методами управления процессами разработки требований, оценки рисков, приобретения, проектирования, конструирования, тестирования, эволюции и сопровождения; в части следующих результатов обучения:	
з1. знать методы выявления требований	
у1. уметь выбирать средства реализации требований к программному обеспечению	
у4. уметь вырабатывать варианты реализации требований	

2. Требования НГТУ к результатам освоения дисциплины

Таблица 2.1

Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)	Формы организации занятий
<i>Проектирование программных систем</i>	
ПК.1.з1 знать методы и средства проектирования программного обеспечения	
1. знать методы и средства проектирования программного обеспечения	Практические занятия; Самостоятельная работа
ОПК.3.у1 владеть объектно-ориентированным подходом к программированию	

2. владеть объектно-ориентированным подходом к программированию	Практические занятия
ПК.5.31 знать методологии разработки программного обеспечения	
3. знать методологии разработки программного обеспечения	Практические занятия; Лабораторные работы; Самостоятельная работа
ПК.7.31 знать методы выявления требований	
4. знать методы выявления требований	Практические занятия
ПК.7.y1 уметь выбирать средства реализации требований к программному обеспечению	
5. уметь выбирать средства реализации требований к программному обеспечению	Практические занятия; Лабораторные работы; Самостоятельная работа
ПК.7.y4 уметь вырабатывать варианты реализации требований	
6. уметь вырабатывать варианты реализации требований	Практические занятия; Лабораторные работы; Самостоятельная работа
ПК.14.y7 уметь проводить оценку и обоснование рекомендуемых решений	
7. уметь проводить оценку и обоснование рекомендуемых решений	Практические занятия; Лабораторные работы; Самостоятельная работа
ПК.16.31 знать языки формализации функциональных спецификаций	
8. знать языки формализации функциональных спецификаций	Практические занятия; Самостоятельная работа
ПК.16.y1 уметь разрабатывать технические спецификации на программные компоненты и их взаимодействие	
9. уметь разрабатывать технические спецификации на программные компоненты и их взаимодействие	Практические занятия; Лабораторные работы; Самостоятельная работа
ПК.19.35 знать нотации моделирования	
10. знать нотации моделирования	Практические занятия; Самостоятельная работа
ПК.19.y5 уметь моделировать бизнес-процессы	
11. уметь моделировать бизнес-процессы	Практические занятия; Лабораторные работы; Самостоятельная работа
ПК.19.y6 уметь применять соглашение о моделировании	
12. уметь применять соглашение о моделировании	Практические занятия; Лабораторные работы; Самостоятельная работа
ПК.22.36 знать методы и средства проектирования программных интерфейсов	
13. знать методы и средства проектирования программных интерфейсов	Практические занятия; Лабораторные работы; Самостоятельная работа
ПК.1.33 знать принципы построения архитектуры программного обеспечения и виды архитектур программного обеспечения	
14. знать принципы построения архитектуры программного обеспечения и виды архитектур программного обеспечения	Практические занятия; Самостоятельная работа
ПК.1.36 знать возможности существующей программно-технической архитектуры	
15. знать возможности существующей программно-технической архитектуры	Практические занятия
ПК.1.y5 уметь использовать возможности имеющейся технической и/или программной архитектуры	
16. уметь использовать возможности имеющейся технической и/или программной архитектуры	Практические занятия; Лабораторные работы

ПК.1.32 знать основные методы измерения и оценки характеристик программного обеспечения	
17.знать основные методы измерения и оценки характеристик программного обеспечения	Лабораторные работы
ПК.1.35 знать типовые метрики программного обеспечения	
18.знать типовые метрики программного обеспечения	Практические занятия

3. Содержание и структура учебной дисциплины

Таблица 3.1

Темы лабораторных работ	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 5				
Дидактическая единица: Предмет программной инженерии, жизненный цикл проекта, методологии проектирования и модели процессов ЖЦ				
1. Анализ бизнес-процессов и разработка видения проекта	1	4	11, 16, 17, 7	Для выбранного варианта разработать следующие документы фазы исследования проекта (ТП моделирование предметной области и разработка требований). - Глоссарий для значимых элементов предметной области; - Бизнес-требования - особенности проекта, обеспечивающие его привлекательность, предполагаемые отличия от аналогов, возможности коммерческого использования, способы монетизации - Границы проекта - перечень бизнес-процессов, поддерживаемых и не поддерживаемых системой - Перечень пользователей проекта Словесное описание бизнес-процессов предметной области, разработка моделей бизнес-процессов предметной области в виде диаграмм потоков данных и моделей на языке BPMN
2. Моделирование предметной области	1	4	11, 12, 3, 7	Разработка и верификация модели предметной области (диаграммы классов) с учетом существующих бизнес-процессов и представления предметной области в информационной системе

3. Разработка модели прецедентов.	2	2	12, 13, 3, 5, 6	Разработать модель прецедентов, кратко описать роли и содержание прецедентов, расписать сценарии 2-3 наиболее значимых прецедентов, основываясь на модели предметной области
4. Разработка требований.	2	4	12	Определить полный перечень функциональных и нефункциональных требований к системе. На его основе разработать документ "Спецификация требований к ПО" и древовидную спецификацию требований
6. Разработка архитектуры системы	2	4	12, 16, 7, 9	. Разработать архитектуру системы, оформив результат в виде документа, включающего в себя: - Общее архитектурное решение: архитектурные модели, компоненты (подсистемы), слои. Стили реализации отдельных компонент и подсистем; - Структуру программного кода (языки реализации, пакеты, основные классы) - диа-грамма пакетов и классов реализации; - Структуру программных компонент и артефактов - БД, файлов данных, конфигурационных файлов, служб и сторонних библиотек (диаграмма компонентов); - Средства коммуникаций, протоколы, форматы и стандарты обмена и хранения данных. - Описание внутреннего параллелизма и синхронизации программных компонент.

Таблица 3.2

Темы практических занятий	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 5				
Дидактическая единица: UML как средство поддержки процесса проектирования				

1. Структура UML, статическая и динамическая составляющие модели	0	2	10	Структура UML, статическая и динамическая составляющие модели. Составляющие элементы, механизмы, архитектура. Составные элементы: сущности, отношения, диаграммы. Виды сущностей: структурные сущности - класс, интерфейс, кооперация, прецедент, активный класс, компонент, узел; поведенческие сущности - взаимодействия, деятельности, автоматы; группирующая сущность - пакет, аннотационная сущность - примечание. Виды отношений: зависимость, ассоциация, агрегация, композиция, включение, обобщение, реализация. Отношения. Связи - отношения между объектами. Направленность связи, Сообщения. Ассоциации - отношения между классами. Свойства ассоциации: имя, кратность, навигация, атрибуты. Рефлексивные ассоциации, деревья и сети. Классы атрибутов ассоциаций (классы-ассоциации). Зависимости. Зависимости использования "use", "call", "parameter", "send" и "instantiate". Зависимости абстракции. Зависимости доступа. Виды диаграмм. Спецификации. Дополнения. Принятые деления: классификатор-экземпляр, интерфейс-реализация. Стереотипы
--	---	---	----	---

2. Диаграммы UML. Структуры и поведение	0	2	10, 12, 8	Диаграммы взаимодействия. Линия жизни, активности, сообщения, их виды. Комбинированные фрагменты и операторы. Включения, параметры, шлюзы. Продолжения. Их аналоги в программировании. Коммуникационные диаграммы. Диаграммы деятельности. Технология сетей Петри. Параллелизм. Поток управления, узел действия, ребро, узел управления, объектный узел, буферизация и в объектном узле. Объектные узлы - параметры, состояния объектных узлов. Контакты. Прерывающие ребра. Контакты исключений. Потоки объектов. Их аналоги в программировании (фаза реализации). Диаграммы состояний. Конечные автоматы. Состояния. Переходы. События. Сигналы. Составные состояния. Подавтоматы. Предыстория.
Дидактическая единица: Предмет программной инженерии, жизненный цикл проекта, методологии проектирования и модели процессов ЖЦ				
3. Жизненный цикл ПО	0	1	3	Свод знаний по программной инженерии SWEBOOK Жизненный цикл (ЖЦ) программного продукта и проекта. "Легкие" и "тяжелые" модели процессов разработки ПО. Обзор: "как получится", ГОСТы, RUP, Agile, XP. Модели жизненного цикла. Каскадная, итерационная и спиральная модели. Организационная и технологическая составляющие жизненного цикла. Роль UML в формализации технологической составляющей. Моделирование, виды моделей. UML как средство описания различных аспектов "мысленной" модели проектируемой системы.

4. Унифицированный процесс UP	0	1	2, 3	Унифицированный процесс UP. Фазы жизненного цикла: начало, уточнение, построение, внедрение. Содержание и результаты фаз. Итерация и ее рабочие потоки: требования, анализ, проектирование, реализация, тестирование, их содержание. Фаза начала. Видение проекта, глоссарий. Рабочий поток анализа предметной области. Диаграммы классов предметной области.
5. Бизнес-аналитика и системная аналитика	0	2	10, 11, 3, 4, 6	Моделирование предметной области и бизнес-процессов. Бизнес-процессы, диаграммы потоков данных. Язык описания бизнес-процессов BPMN. Различие между бизнес-аналитикой и системной аналитикой. Модель предметной области
6. Управление требованиями	0	2	4, 5, 6	Требования. Место рабочего потока определения требований. Функциональные и не-функциональные, перечень. Атрибуты требований: приоритеты, состояние, трудоемкость, стабильность, риск, полезность. Способы извлечения и фильтрации требований. Актеры и роли. Прецеденты. Выявление актеров и прецедентов: бизнес-модель предметной области, модель требований, глоссарий проекта. Диаграммы прецедентов, детализация прецедентов, спецификации прецедентов. Отображение требований на прецеденты.
7. Архитектура и проектирование ПС	0	3	14, 16, 7	Архитектура как система представлений: представление классов, процессов, реализации, развертывания и прецедентов (требований). Классы анализа. Виды классов: граница, управление, сущность. Выявление классов анализа. Диаграммы устойчивости. Архитектурные аспекты технологического процесса проектирования (по SWEBOOK)

8. Проектирование графического интерфейса	0	1	13	Проектирование графического интерфейса (GUI). Основные аспекты. Архитектурное проектирование, основанное на GUI. Факторы, характеризующие GUI: производительность, человеческие ошибки, обучение, субъективное восприятие, запоминание, поиск, визуализация, навигация
9. Примеры проектирования ПС	0	2	14, 15, 2, 3, 6, 7, 8, 9	Примеры проектирования. Многослойная структура клиент-серверных приложений. Слои, интерфейсы, протоколы. Тонкие и толстые клиенты. Примеры проектирования и основного набора моделей и документов: система выдачи ключей аудиторий. . система подсчета рейтинга успеваемости. Распределение компонентов. Клиент-серверные приложения: тонкий и толстый клиент. Обработка ошибок и исключительных ситуаций, управление отказоустойчивостью и надежностью. Разделение представления и сущности: концепция модель-вид-контроллер.
10. Экстремальное и гибкое программирование/проектирование	0	1	1, 3	Экстремальное и гибкое программирование. Манифест экстремального программирования (XP). Гибкие (agile) технологии. SCRUM. Agile UP, ICONIX. SCRUM как технологический фреймворк.. Терминология. Спринт. Митинг. Собственник проекта. Команда. SCRUM-мастер. Беклог проекта и спринта. Планирование спринта. Диаграмма сгорания. Оценка трудоемкости. Покер-планирование
16. Оценка программного кода.	0	1	18	Метрики кода. Метрики количественные, сложности потока управления и потока данных, метрики ООП, прагматические метрики. Средства оценки качества программного кода

4. Самостоятельная работа обучающегося

№	Виды самостоятельной работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации
Семестр: 5				
1	РГЗ	1, 3	22	4
РГЗ выполняется в форме реферата - описания одной из методологий проектирования (сопровождения) и поддерживающего его фреймворка. Реферат должен содержать: аннотированный обзор источников, на основании которого дается собственное оригинальное описание методологии и фреймворка с перечнем свойств, характеристик, описания поддерживаемых процессов проектирования, особенностей установки и развертывания. Также создается аннотированный список отзывов о применении методологии и фреймворка. , подробная информация приведена в приложении №3 : Романов Е. Л. Программная инженерия [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000220170 . - Загл. с экрана.				
2	Подготовка к занятиям	11, 12, 5, 6, 7, 9	31	0
Оформление результатов лаб. работ, подготовка к защите. Подготовка к практическим занятиям по заранее объявленной теме: Романов Е. Л. Программная инженерия [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000220170 . - Загл. с экрана. Пустовалова Н. В. Программная инженерия (метрическая теория программ) [Электронный ресурс] : электронный учебно-методический комплекс [для студентов направления 080800 Прикладная информатика] / Н. В. Пустовалова, Г. И. Кайгородцев ; Новосиб. гос. техн. ун-т. - Новосибирск, [2014]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000208496 . - Загл. с экрана.				
3	Подготовка к аттестации	1, 10, 13, 14, 3, 8	10	3
: Романов Е. Л. Программная инженерия [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000220170 . - Загл. с экрана. Пустовалова Н. В. Программная инженерия (метрическая теория программ) [Электронный ресурс] : электронный учебно-методический комплекс [для студентов направления 080800 Прикладная информатика] / Н. В. Пустовалова, Г. И. Кайгородцев ; Новосиб. гос. техн. ун-т. - Новосибирск, [2014]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000208496 . - Загл. с экрана.				

5. Технология обучения

Для организации и контроля самостоятельной работы обучающихся, а также проведения консультаций применяются информационно-коммуникационные технологии (табл. 5.1).

Таблица 5.1

Деятельность	Информационно-коммуникационные технологии
Информирование	Социальные сети: YandexDisk vk.com/cprog_cs Skype ИДО - дисц."Программная инженерия"; Чат: Skype
Консультирование	e-mail: romanov@corp.nstu.ru; Чат: Skype
Размещение учебных материалов	Социальные сети: YandexDisk vk.com/cprog_cs Skype ИДО - дисц."Программная инженерия"

6. Правила аттестации обучающихся по учебной дисциплине

Для аттестации обучающихся по дисциплине используется балльно-рейтинговая система (БРС), позволяющая выставлять оценки по традиционной шкале и 15-уровневой ECTS. Краткая информация о БРС приведена в табл. 6.1.

Таблица 6.1

Оцениваемые виды деятельности обучающихся	Мин. балл	Максимальный балл
Семестр: 5		
<i>Лабораторная:</i>	25	50
<i>РГЗ:</i>	15	30
<i>Зачет:</i>	0	20

В таблице 6.2 представлено соответствие форм контроля заявляемым требованиям к результатам освоения дисциплины.

Таблица 6.2

Коды компетенций ФГОС	Результаты обучения	Формы контроля		
		Защита ЛР	Защита РГЗ	Зачет
ОПК.3	у1. владеть объектно-ориентированным подходом к программированию			+
ПК.1	з1. знать методы и средства проектирования программного обеспечения			+
	з2. знать основные методы измерения и оценки характеристик программного обеспечения			+
	з3. знать принципы построения архитектуры программного обеспечения и виды архитектур программного обеспечения		+	+
	з5. знать типовые метрики программного обеспечения			+
	з6. знать возможности существующей программно-технической архитектуры			+
	у5. уметь использовать возможности имеющейся технической и/или программной архитектуры	+		+
ПК.14	у7. уметь проводить оценку и обоснование рекомендуемых решений			+
ПК.16	з1. знать языки формализации функциональных спецификаций			+
	у1. уметь разрабатывать технические спецификации на программные компоненты и их взаимодействие			+
ПК.19	з5. знать нотации моделирования			+
	у5. уметь моделировать бизнес-процессы	+		+
	у6. уметь применять соглашение о моделировании			+
ПК.22	з6. знать методы и средства проектирования программных интерфейсов			+
ПК.5	з1. знать методологии разработки программного обеспечения		+	+
ПК.7	з1. знать методы выявления требований			+
	у1. уметь выбирать средства реализации требований к программному обеспечению	+		+
	у4. уметь вырабатывать варианты реализации требований			+

Фонд оценочных средств по дисциплине представлен в приложении № 1 к рабочей программе.

7. Литература

Основная литература

1. Липаев В. В. Программная инженерия. Методологические основы : [учебник для вузов по направлению "Бизнес-информатика" (080700)] / В. В. Липаев ; Гос. ун-т - высш. шк. экономики. - М., 2006. - 605, [1] с.
2. Кайгородцев Г. И. Введение в курс метрической теории и метрологии программ : [учебник для вузов по специальности 080801 "Прикладная информатика (по областям)" и другим экономическим специальностям] / Г. И. Кайгородцев. - Новосибирск, 2011. - 191 с. : ил., табл. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000155463
3. Беркун С. Искусство управления IT- проектами / Скотт Беркун ; [пер. с англ. Н. Вильчинский]. - М. [и др.], 2010. - 431 с. : ил.
4. Буч Г. Язык UML : руководство пользователя / Г. Буч, Д. Рамбо, И. Якобсон ; [пер. с англ. Н. Мухина]. - М., 2007. - 493 с. : ил.

Дополнительная литература

1. Рейнвотер Д. Х. Как пасти котов : наставление для программистов, руководящих другими программистами / Дж. Ханк Рейнвотер ; [пер. с англ. Ю. Гороховский]. - СПб., 2008. - 255 с. : ил.
2. Якобсон А. Унифицированный процесс разработки программного обеспечения / А. Якобсон, Г. Буч, Дж. Рамбо ; [пер. с англ. В. Горбункова]. - СПб. [и др.], 2002. - 492 с. : ил.
3. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ / Гради Буч ; пер. с англ. под ред. И. Романовского и Ф. Андреева. - М., 1998. - 558 с. : ил. - Тит. л. парал. рус., англ..
4. Орлов С. А. Технологии разработки программного обеспечения. Разработка сложных программных систем : Учебное пособие для вузов / С. А. Орлов. - СПб., 2003. - 473 с. : ил.
5. Вязовик Н. А. Программирование на Java. Курс лекций / Н. А. Вязовик ; Интернет ун-т информ. технологий. - М., 2003. - 586 с. : ил., схемы

Интернет-ресурсы

1. Архипенков С. Лекции по управлению программными проектами [Электронный ресурс] / С. Архипенков. - Москва, 2009. - 128 с. - Режим доступа : http://www.arkhipenkov.ru/resources/sw_project_management.pdf. - Загл. с экрана.
2. ЭБС НГТУ : <http://elibrary.nstu.ru/>
3. ЭБС «Издательство Лань» : <https://e.lanbook.com/>
4. ЭБС IPRbooks : <http://www.iprbookshop.ru/>
5. ЭБС "Znaniy.com" : <http://znaniy.com/>
6. :

8. Методическое и программное обеспечение

8.1 Методическое обеспечение

1. Пустовалова Н. В. Программная инженерия (метрическая теория программ) [Электронный ресурс] : электронный учебно-методический комплекс [для студентов направления 080800 Прикладная информатика] / Н. В. Пустовалова, Г. И. Кайгородцев ; Новосиб. гос. техн. ун-т. - Новосибирск, [2014]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000208496. - Загл. с экрана.
2. Романов Е. Л. Программная инженерия [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000220170. - Загл. с экрана.

8.2 Специализированное программное обеспечение

1 COSOMO® П.2000.4

9. Материально-техническое обеспечение

Презентационное оборудование

№	Наименование	Назначение
1	Презентационное оборудование (мультимедиа-проектор, экран, компьютер для управления)	Проведение практических занятий

Компьютерный класс

№	Наименование	Назначение
1	Компьютерный класс (Компьютеры объединены в локальную сеть с выходом в Internet)	Компьютеры объединены в локальную сеть с выходом в Internet

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Новосибирский государственный технический университет»

Кафедра вычислительной техники

“УТВЕРЖДАЮ”
ДЕКАН АВТФ
к.т.н., доцент И.Л. Рева
“__” ____ Г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

дисциплины

Проектирование программных систем

Образовательная программа: 09.03.04 Программная инженерия, профиль: Технологии
разработки программного обеспечения

1. Обобщенная структура фонда оценочных средств учебной дисциплины

Обобщенная структура фонда оценочных средств по дисциплиныПроектирование программных систем приведена в Таблице.

Таблица

Формируемые компетенции	Показатели сформированности компетенций (знания, умения, навыки)	Темы	Этапы оценки компетенций	
			Мероприятия текущего контроля (курсовой проект, РГЗ(Р) и др.)	Промежуточная аттестация (экзамен, зачет)
ОПК.3 готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	у1. владеть объектно-ориентированным подходом к программированию	Примеры проектирования ПС Унифицированный процесс UP		Зачет, вопрос 3
ПК.1/ПТ готовность применять основные методы и инструменты разработки программного обеспечения	з1. знать методы и средства проектирования программного обеспечения	Экстремальное и гибкое программирование/проектирование		Зачет, вопросы 15-16
ПК.1/ПТ	з2. знать основные методы измерения и оценки характеристик программного обеспечения	Анализ бизнес-процессов и разработка видения проекта		Зачет, вопросы 4
ПК.1/ПТ	з3. знать принципы построения архитектуры программного обеспечения и виды архитектур программного обеспечения	Архитектура и проектирование ПС Примеры проектирования ПС	РГЗ	Зачет, вопросы 6-7
ПК.1/ПТ	з5. знать типовые метрики программного обеспечения	Оценка программного кода.		Зачет, вопросы 17
ПК.1/ПТ	з6. знать возможности существующей программно-технической архитектуры	Примеры проектирования ПС		Зачет, вопрос 8
ПК.1/ПТ	у5. уметь использовать возможности имеющейся технической и/или программной архитектуры	Анализ бизнес-процессов и разработка видения проекта Архитектура и проектирование ПС Разработка архитектуры системы	Отчет по лабораторной работе	Зачет, вопросы...

ПК.14/НИ готовность обосновать принимаемые проектные решения, осуществлять постановку и выполнение экспериментов по проверке их корректности и эффективности	у7. уметь проводить оценку и обоснование рекомендуемых решений	Анализ бизнес-процессов и разработка видения проекта Архитектура и проектирование ПС Моделирование предметной области Примеры проектирования ПС		Зачет, вопросы 4-8
ПК.16/А способность формализовать предметную область программного проекта и разработать спецификации для компонентов программного продукта	з1. знать языки формализации функциональных спецификаций	Диаграммы UML. Структуры и поведение Примеры проектирования ПС		Зачет, вопросы 10-14
ПК.16/А	у1. уметь разрабатывать технические спецификации на программные компоненты и их взаимодействие	Примеры проектирования ПС		Зачет, вопросы 8,10-14
ПК.19/П владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения	з5. знать нотации моделирования	Бизнес-аналитика и системная аналитика Диаграммы UML. Структуры и поведение Структура UML, статическая и динамическая составляющие модели		Зачет, вопросы 10-14
ПК.19/П	у5. уметь моделировать бизнес-процессы	Анализ бизнес-процессов и разработка видения проекта Бизнес-аналитика и системная аналитика Моделирование предметной области	Отчет по лабораторной работе	Зачет, вопросы 4
ПК.19/П	у6. уметь применять соглашение о моделировании	Диаграммы UML. Структуры и поведение Моделирование предметной области Разработка архитектуры системы Разработка модели прецедентов. Разработка требований.		Зачет, вопросы 10-14
ПК.22/П способность создавать программные интерфейсы	з6. знать методы и средства проектирования программных интерфейсов	Проектирование графического интерфейса Разработка модели прецедентов.		Зачет, вопрос 9
ПК.5/ПТ владение стандартами и моделями жизненного цикла	з1. знать методологии разработки программного обеспечения	Бизнес-аналитика и системная аналитика Жизненный цикл ПО Моделирование предметной области Примеры	РГЗ	Зачет, вопросы 1-3,15-16

		проектирования ПС Разработка модели прецедентов. Унифицированный процесс УР Экстремальное и гибкое программирование/проектирование		
ПК.7/ОУ владение методами управления процессами разработки требований, оценки рисков, приобретения, проектирования, конструирования, тестирования, эволюции и сопровождения	з1. знать методы выявления требований	Бизнес-аналитика и системная аналитика Управление требованиями		Зачет, вопрос 5
ПК.7/ОУ	у1. уметь выбирать средства реализации требований к программному обеспечению	Разработка модели прецедентов. Управление требованиями	Отчет по лабораторной работе	Зачет, вопрос 5
ПК.7/ОУ	у4. уметь вырабатывать варианты реализации требований	Бизнес-аналитика и системная аналитика Примеры проектирования ПС Разработка модели прецедентов. Управление требованиями		Зачет, вопрос 5

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по дисциплине проводится в 5 семестре - в форме дифференцированного зачета, который направлен на оценку сформированности компетенций ОПК.3, ПК.1/ПТ, ПК.14/НИ, ПК.16/А, ПК.19/П, ПК.22/П, ПК.5/ПТ, ПК.7/ОУ.

Зачет проводится в устной форме по билетам.

Кроме того, сформированность компетенций проверяется при проведении мероприятий текущего контроля, указанных в таблице раздела 1.

В 5 семестре обязательным этапом текущей аттестации является расчетно-графическое задание (работа) (РГЗ(Р)). Требования к выполнению РГЗ(Р), состав и правила оценки сформулированы в паспорте РГЗ(Р).

Общие правила выставления оценки по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОПК.3, ПК.1/ПТ, ПК.14/НИ, ПК.16/А, ПК.19/П, ПК.22/П, ПК.5/ПТ, ПК.7/ОУ, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

Ниже порогового. Уровень выполнения работ не отвечает большинству основных требований, теоретическое содержание курса освоено частично, пробелы могут носить существенный характер, необходимые практические навыки работы с освоенным материалом сформированы не достаточно, большинство предусмотренных программой обучения учебных заданий не выполнены или выполнены с существенными ошибками.

Пороговый. Уровень выполнения работ отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера,

необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками.

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

Продвинутый. Уровень выполнения работ отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

Паспорт зачета

по дисциплине «Проектирование программных систем», 5 семестр

1. Методика оценки

Зачет проводится в устной (письменной) форме, по билетам (тестам). Билет формируется по следующему правилу: оба вопроса выбираются из всего диапазона. В ходе зачета преподаватель вправе задавать студенту дополнительные вопросы из общего перечня (п. 4).

Форма билета для зачета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № _____

к зачету по дисциплине «Проектирование программных систем»

Вопрос 1. Жизненный цикл (ЖЦ) программного продукта и проекта. «Легкие» и «тяжелые» модели процессов разработки ПО. Этапы и технологические процессы (дисциплины) ЖЦ. Результаты этапов и основные документы. Каскадная, итеративная и спиральная модели

Вопрос 2. UML. Диаграммы состояний. Конечные автоматы.

Утверждаю: зав. кафедрой _____ должность, ФИО
(подпись)

(дата)

Согласно положению о балльно-рейтинговой системе НГТУ, базовый балл рейтинга за зачет составляет 20 баллов в целом и по 10 баллов за вопрос. Соответственно, критерий оценки определяется в процентах к этому баллу:

- Ответ на билет для зачета считается **неудовлетворительным**, если студент при ответе на вопросы не дает определений основных понятий, не способен показать причинно-следственные связи явлений, допускает принципиальные ошибки, оценка составляет менее 50% базовой
- Ответ засчитывается на **пороговом** уровне, если в теоретических вопросах даны только основные определения - оценка составляет не более 50% базовой
- Ответ засчитывается на **базовом** уровне, если в теоретических вопросах отражена структура вопроса (определения, свойства, правила) - оценка составляет 50-80% базовой

- Ответ засчитывается на **продвинутом** уровне, если дан развернутый ответ на теоретический вопрос и доп. вопросы - оценка составляет 80-100% базовой

2. Шкала оценки

В общей оценке по дисциплине баллы за зачет учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

3. Вопросы к зачету по дисциплине «Проектирование программных систем»

1. Сущность программной инженерии (ПИ). Связь с computer science. Особенности в сравнении и другими инженерными дисциплинами. Свод знаний и ПИ SWEBOK
2. Жизненный цикл (ЖЦ) программного продукта и проекта. «Легкие» и «тяжелые» модели процессов разработки ПО. Этапы и технологические процессы (дисциплины) ЖЦ. Результаты этапов и основные документы. Каскадная, итеративная и спиральная модели.
3. Унифицированный процесс UP. Фазы жизненного цикла: исследование, анализ, реализация, внедрение. Содержание и результаты фаз. Итерация и ее рабочие потоки: требования, анализ, проектирование, реализация, тестирование, их содержание.
4. Фаза исследования. Основные дисциплины и артефакты. Дисциплина «анализ предметной области», бизнес-анализ. Диаграммы потоков данных, деятельности. Моделирование предметной области.
5. Фаза анализа и проектирования. Дисциплина «анализ требований». Способы извлечения и фильтрации требований. Бизнес-требования, бизнес-требования, системные требования, функциональные требования. Разработка и управление требованиями. Документ «спецификация требований к ПО ». Диаграммы прецедентов.
6. Фаза анализа и проектирования. Понятие архитектуры, ее многомерность. Основные методы проектирования и их особенности: структурное, функциональное, объектно-ориентированное, компонентное, проектирование на основе структур данных. Классы анализа. Виды классов: граница, управление, сущность. Диаграммы устойчивости. Архитектурные аспекты технологического процесса проектирования (по SWEBOK)
7. Фаза анализа и проектирования. Дисциплина проектирование (design). Ключевые моменты проектирования по SWEBOK: параллелизм, контроль и обработка событий, распределение компонентов, обработка ошибок и исключительных ситуаций и обеспечение отказоустойчивости, взаимодействие и представление (MVC), сохраняемость данных (доступность «долгоживущих» данных).
8. Фаза анализа и проектирования. Многоуровневая архитектура клиент-серверных приложений. Тонкие и толстые клиенты. Локальное и сетевое взаимодействие слоев через интерфейсы и протоколы. Совместное использование кода различными типами клиентов.
9. Фаза анализа и проектирования Проектирование графического интерфейса (GUI). Основные аспекты. Архитектурное проектирование, основанное на GUI. Факторы, характеризующие GUI: производительность, человеческие ошибки, обучение, субъективное восприятие, запоминание, поиск, визуализация, навигация

10. Виды моделей. Сущность UML как средства моделирования. Структура UML, статическая и динамическая составляющие модели. Составные элементы: сущности, отношения, диаграммы. Виды сущностей: структурные сущности – класс, интерфейс, кооперация, прецедент, активный класс, компонент, узел; поведенческие сущности – взаимодействия, деятельности, автоматы; группирующая сущность – пакет, аннотационная сущность – примечание.
11. UML. Виды отношений: зависимость, ассоциация, агрегация, композиция, включение, обобщение, реализация. Отношения. Связи – отношения между объектами. Направленность связи, Сообщения. Диаграммы объектов. Ассоциации – отношения между классами. Свойства ассоциации: имя, кратность, навигация, атрибуты. Рефлексивные ассоциации, деревья и сети. Классы атрибутов ассоциаций (классы-ассоциации). Зависимости. Зависимости использования «use», «call», «parameter», «send» и «instantiate». Зависимости абстракции. Зависимости доступа.
12. UML. Принятые деления: классификатор-экземпляр, интерфейс-реализация. Расширения: ограничения, стереотипы. Классификация диаграмм. Диаграммы классов (объектов). Диаграммы взаимодействий, коммуникационные диаграммы.
13. UML. Диаграммы деятельности. Технология сетей Петри. Параллелизм. Поток управления, узел действия, ребро, узел управления, объектный узел, буферизация и в объектном узле. Объектные узлы – параметры, состояния объектных узлов. Контакты. Прерывающие ребра. Контакты исключений. Потоки объектов. Их аналоги в программировании
14. UML. Диаграммы состояний. Конечные автоматы.
15. Экстремальное и гибкое программирование. Манифест экстремального программирования (XP). Гибкие (agile) технологии. SCRUM. Agile UP, ICONIX.
16. SCRUM как технологический фреймворк.. Терминология. Спринт. Митинг. Собственник проекта. Команда. SCRUM-мастер. Беклог проекта и спринта. Планирование спринта. Диаграмма сгорания. Оценка трудоемкости. Покер-планирование.
17. Оценка программного кода. Метрики кода. Метрики количественные, сложности потока управления и потока данных, метрики ООП, прагматические метрики. Средства оценки качества программного кода.

Паспорт расчетно-графического задания (работы)

по дисциплине «Проектирование программных систем», 5 семестр

1. Методика оценки

РГЗ выполняется в форме реферата - описания одной из методологий проектирования (сопровождения) и поддерживающего его фреймворка. Реферат должен содержать:

- аннотированный обзор источников;
- собственное оригинальное описание методологии и фреймворка;
- перечень свойств, характеристик, описания поддерживаемых процессов проектирования, особенностей установки и развертывания;
- аннотированный список отзывов о применении методологии и фреймворка.
- выводы

1. Критерии оценки

Согласно положению о балльно-рейтинговой системе НГТУ, базовый балл рейтинга за РГР определен в рабочей программе. Соответственно, критерий оценки определяется в процентах к этому баллу:

- РГР считается **не выполненной**, если выполнены не все части РГЗ(Р), оценка составляет менее 30% базовой.
- РГР засчитывается на **пороговом** уровне, содержание не полностью раскрывает задание, список источников обзора ограничен, пояснительная записка оформлена со значительными структурными, стилистическими и грамматическими ошибками - оценка составляет 30-50% базовой
- РГР засчитывается на **базовом** уровне, если содержание работы соответствует заданию, пояснительная записка оформлена в целом грамотно - оценка составляет 50-80% базовой
- РГР засчитывается на **продвинутом** уровне, если содержание раскрыто полностью и разнообразно, количество источников обзора более 30, в пояснительной записке отражены все аспекты тематики, имеется аналитическая часть - оценка составляет 80-100% базовой

3. Шкала оценки

В общей оценке по дисциплине баллы за РГЗ(Р) учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Примерный перечень тем РГЗ(Р)

1. Обзор средств поддержки гибкого проектирования (agile)
2. TDD – разработка, направляемая тестированием
3. AgileUP – совмещение методологии гибкой разработки в рамках унифицированного процесса
4. SCRUM – фреймворк гибкого проектирования. Практика применения.

5. Обзор проектов, выполненных в методологии UP