

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

Кафедра вычислительной техники
Кафедра автоматизированных систем управления

“УТВЕРЖДАЮ”

ДЕКАН АВТФ

к.т.н. Рева И. Л.

“ ____ ” _____ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Программная инженерия

Образовательная программа: 09.03.01 Информатика и вычислительная техника, профиль: Сетевые информационные технологии

Курс: 4, семестр : 7

Факультет автоматики и вычислительной техники,

		Семестр
№	Вид деятельности	7
1	Всего зачетных единиц (кредитов)	4
2	Всего часов	144
3	Всего занятий в контактной форме, час.	64
4	Лекции, час.	18
5	Практические занятия, час.	0
6	Лабораторные занятия, час.	36
7	из них в активной и интерактивной форме, час.	8
8	Аттестация, час.	2
9	Консультации, час.	8
10	Самостоятельная работа, час.	80
11	Виды самостоятельной работы (курсовой проект, курсовая работа, РГЗ, подготовка к контрольной работе)	РГЗ
12	Вид аттестации	Э

Рабочая программа составлена на основании федерального государственного образовательного стандарта высшего образования по направлению (специальности): 09.03.01 Информатика и вычислительная техника

ФГОС введен в действие приказом №5 от 12.01.2016 г. , дата утверждения: 09.02.2016 г.

Место дисциплины в структуре учебного плана: Блок 1, вариативная

Рабочая программа разработана на основе компетентностной модели выпускника по направлению (специальности): 09.03.01 Информатика и вычислительная техника

Рабочая программа обсуждена на заседании кафедры

ВТ, протокол заседания кафедры №6 от 20.06.2017
АСУ, протокол заседания кафедры №7 от 20.06.2017

Утверждена на совете факультета автоматизации 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.
профессор, д.т.н. Гриф М. Г.

Ответственный за образовательную программу:

заведующий кафедрой Гриф М. Г.

1. Внешние требования

Таблица 1.1

Компетенция ФГОС: ОПК.3 способность разрабатывать бизнес-планы и технические задания на оснащение отделов, лабораторий, офисов компьютерным и сетевым оборудованием; в части следующих результатов обучения:	
з3. знать основы управления проектами создания и внедрения программных продуктов	
Компетенция ФГОС: ПК.2 способность разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования; в части следующих результатов обучения:	
у3. уметь оценивать метрики программных проектов различными методами	
Компетенция НГТУ: ПК.10.В/ПК готовность к разработке моделей компонентов информационных систем, включая модели баз данных и модели интерфейсов "человек - электронно-вычислительная машина"; в части следующих результатов обучения:	
з2. знать основы объектно-ориентированного подхода к программированию	
з4. знать основы системного программирования	
з6. владеть средствами структурного и объектного моделирования программных систем	
з7. знать методологии моделирования, используемые при проектировании информационных систем	
з9. знать стандарты создания пользовательского интерфейса программных продуктов	
у2. уметь строить функциональные модели информационных систем	
у3. уметь строить модели потоков данных информационных систем	
у4. уметь строить объектные модели информационных систем	
Компетенция НГТУ: ПК.11.В/НПд готовность к разработке концептов и проведению занятий по обучению работников применению программно-методических комплексов, используемых на предприятии; в части следующих результатов обучения:	
з1. знать принципы формирования команды ИТ-проекта	
Компетенция НГТУ: ПК.9.В/НИ готовность обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности; в части следующих результатов обучения:	
з4. знать основы методологии создания программного обеспечения	
з6. знать структурный и объектно-ориентированный подходы к проектированию информационных систем	
у4. уметь обосновывать принимаемые проектные решения	

2. Требования НГТУ к результатам освоения дисциплины

Таблица 2.1

Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)	Формы организации занятий
Программная инженерия	
ПК.10.В/ПК.з9 знать стандарты создания пользовательского интерфейса программных продуктов	
1. знать стандарты создания пользовательского интерфейса программных продуктов	Лекции; Лабораторные работы; Самостоятельная работа
ПК.10.В/ПК.з2 знать основы объектно-ориентированного подхода к программированию	
2. знать основы объектно-ориентированного подхода к программированию	Лекции; Самостоятельная работа
ПК.10.В/ПК.з4 знать основы системного программирования	
3. знать основы системного программирования	Лекции; Самостоятельная работа
ПК.10.В/ПК.з6 владеть средствами структурного и объектного моделирования программных систем	

4.владеть средствами структурного и объектного моделирования программных систем	Лабораторные работы; Самостоятельная работа
ПК.10.В/ПК.з7 знать методологии моделирования, используемые при проектировании информационных систем	
5.знать методологии моделирования, используемые при проектировании информационных систем	Лекции; Лабораторные работы; Самостоятельная работа
ПК.10.В/ПК.у2 уметь строить функциональные модели информационных систем	
6.уметь строить функциональные модели информационных систем	Лабораторные работы
ПК.10.В/ПК.у3 уметь строить модели потоков данных информационных систем	
7.уметь строить модели потоков данных информационных систем	Лекции; Лабораторные работы
ПК.10.В/ПК.у4 уметь строить объектные модели информационных систем	
8.уметь строить объектные модели информационных систем	Лекции; Лабораторные работы
ПК.2.у3 уметь оценивать метрики программных проектов различными методами	
9.уметь оценивать метрики программных проектов различными методами	Лекции; Лабораторные работы
ОПК.3.з3 знать основы управления проектами создания и внедрения программных продуктов	
10.знать основы управления проектами создания и внедрения программных продуктов	Лекции; Лабораторные работы; Самостоятельная работа
ПК.9.В/НИ.з4 знать основы методологии создания программного обеспечения	
11.знать основы методологии создания программного обеспечения	Лекции; Самостоятельная работа
ПК.9.В/НИ.з6 знать структурный и объектно-ориентированный подходы к проектированию информационных систем	
12.знать структурный и объектно-ориентированный подходы к проектированию информационных систем	Лекции; Лабораторные работы; Самостоятельная работа
ПК.9.В/НИ.у4 уметь обосновывать принимаемые проектные решения	
13.уметь обосновывать принимаемые проектные решения	Лекции; Самостоятельная работа
ПК.11.В/НПд.з1 знать принципы формирования команды ИТ-проекта	
14.знать принципы формирования команды ИТ-проекта	Лекции; Лабораторные работы; Самостоятельная работа

3. Содержание и структура учебной дисциплины

Таблица 3.1

Темы лекций	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 7				
Дидактическая единица: UML как средство поддержки процесса проектирования				

1. Структура UML, статическая и динамическая составляющие модели	0	1	1, 12, 5	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	1	12, 2, 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	0,5	11, 5	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	0,5	11	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	1	12, 5, 7, 8	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	1	11, 13	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	2	11, 12, 8	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	1	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	1	11, 12, 3, 5	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	11	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
Дидактическая единица: Управление программными проектами				

11. Управление проектом как специфическая деятельность и разновидность менеджмент	0	1	10	Управление проектом как специфическая деятельность и разновидность менеджмента. Функциональная, проектная и матричная структура организации, разрабатывающей ПО. Функциональная классификация участников проекта, возможности совмещения функций. Управление проектом на разных фазах. Фаза инициации (начала). Структура и содержание концепции: финансовая и стратегическая ценность, риски, обоснование выполнимости (прототипирование), цели и результаты, допущения и ограничения, ключевые участники, ресурсы, оценка трудоемкости и сроков. Результаты завершения фазы инициации по УР. Фаза анализа. Иерархическая структура работ (ИСР). Планирование управления содержанием, организационной структуры, управления конфигурацией и управления качеством. Базовое расписание проекта. Диаграммы Ганта, критический путь. Результаты завершения фазы анализа по УР
12. Управление рисками.	0	0,5	10	Управление рисками. Определение и характеристики риска. Шкалы оценивания последствий и вероятности. Способы идентификации. Реакция на риски. Наиболее вероятные риски по Боэму и Архипенкову. Качественные оценки рисков. Количественные оценки: анализ чувствительности, дерево решений, имитационное моделирование. Управление, направленное на снижение рисков.

13. Оценка трудоемкости и сроков проекта	0	1	9	Оценка трудоемкости и сроков. Вероятностный характер оценивания, его природа. По-следствия "агрессивного" планирования. Исходные данные для оценивания, характери-стики проекта, используемые в оценивании. Оценка сроков на основании трудоемкости (по Бозму). Оценка на основе собственного опыта. Метод PERT. Оценка на основе функциональных точек. Оценка по отраслевым данным. Метод СОСОМО II.
14. Формирование команды	0	0,5	14	Подбор команды проекта. Функциональная организация команды.. Мотива-ция. Эффективное взаимодействие. Кодекс этики и профессиональной дея-тельности в ПИ.
Дидактическая единица: Технологические процессы конструирования и тестирования ПО				
15. Шаблоны проектирования и конструирования	0	1	12, 3	Сущность шаблонов (паттернов) проектирования как элементов абстрагирования. Виды шаблонов, их назначение, примеры использования. Базовые шаблоны: делегирование, интерфейс, абстрактный класс, обратный вызов. Производящие шаблоны: builder, factory, prototype, singleton, object pool, разделяющие шаблоны - filter. Структурные шаблоны: adapter, bridge, composite, decorator, facade, flyweight, proxy.Шаблон model-view-controller (MVC).Поведенческие шаблоны: command, interpreter, iterator, mediator, momento, observer, state, starategy, template, visitor.Шаблоны параллелизма: session, worker thread, successive update, router, transaction, single threaded execution, lock object, guarded suspen-sion, balking, scheduler, read/write lock,two-phase termination, double buffering, asynchronous processing

16. Оценка программного кода.	0	1	9	Метрики кода. Метрики количественные, сложности потока управления и потока данных, метрики ООП, прагматические метрики. Средства оценки качества программного кода
17. Тестирование, инспекция, отладка, верификация кода. Виды программных ошибок.	0	1	11, 3	Понятие программной ошибки. Философия ошибок. Классификационные характеристики ошибок. Отладка, инспекция и тестирование как этапы поиска ошибок. Обзор и основная классификация распространенных программных ошибок (дефектов), документирование и анализ ошибок. Ошибки вычислений и преобразований. Ошибки структурирования кода. Ошибки форматов входных данных. Ошибки форматов внутренних данных и соглашений по данным. Ошибки сборки, конфигурирования, размещения. Ошибки использования ресурсов. Ошибки, связанные с ограничениями по ресурсам. Ошибки реактивности и производительности. Ошибки параллелизма и синхронизации. Ошибки распределенных систем и протоколов. Ошибки пользовательского интерфейса. Отладка и инспекция программного кода. Сущность отладки. Приемы отладки. Инспекция как неформальный анализ программного кода. Логический и временной анализ. Методы инспектирования

18. Модульное тестирование	0	1	3	Сущность тестирования. . Экономическая сторона тестирования. Методологическая сторона тестирования. Невозможность исчерпывающего тестирования. Алгоритмически неразрешимые проблемы тестирования и инспекции. Понятие теста, тестового набора. Критерии выбора тестов: структурный (белый ящик), функциональный (черный ящик), стохастический, мутационный. Структурное тестирование. Тестирование операторов, условий (решений) и путей. Комбинационное тестирование. Пример. Функциональное тестирование. Классы эквивалентности по входным данным. Проектирование тестового покрытия. Классы эквивалентности по граничным условиям. Примеры. Модульное и интеграционное тестирование как фазы процесса конструирования. Средства автоматизации тестирования при конструировании (JUnit)
19. Разработка, управляемая тестированием (TDD)	0	1	3	Тестирование, совмещенное с проектированием (TDD-Test Driven Development). Технологический цикл. Паттерны тестирования. Приемы рефакторинга.

Таблица 3.2

Темы лабораторных работ	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 7				
Дидактическая единица: Предмет программной инженерии, жизненный цикл проекта, методологии проектирования и модели процессов ЖЦ				

1. Анализ бизнес-процессов и разработка видения проекта	0	4	12, 7, 8	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	8	Разработка и верификация модели предметной области (диаграммы классов) с учетом существующих бизнес-процессов и представления предметной области в информационной системе
3. Разработка модели прецедентов.	1	4	5	Разработать модель прецедентов, кратко описать роли и содержание прецедентов, расписать сценарии 2-3 наиболее значимых прецедентов, основываясь на модели предметной области
4. Разработка требований.	1	4		Определить полный перечень функциональных и нефункциональных требований к системе. На его основе разработать документ "Спецификация требований к ПО" и древовидную спецификацию требований

5. Разработка прототипа графического интерфейса.	1	4	1	Для всех приложений с учетом их функционала и имеющихся прецедентов разработать систему окон графического интерфейса пользователя (GUI) и диаграммы классов (или граф связей) оконных классов. Обосновать принятые решения требованиями из "Спецификации требований к ПО". Дополнить спецификацию с учетом выполненного проектирования
6. Разработка архитектуры системы	1	4	4, 5, 6, 8	. 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.
7. Реализация прецедента	1	4		Для одного из сценариев, разработанного в л.р.2 проиллюстрировать его реализацию в разработанной архитектуре (диаграмма устойчивости, диаграмма последовательности или коммуникационная диаграмма)
Дидактическая единица: Управление программными проектами				
8. Оценка трудоемкости и стоимости программного проекта	1	4	10, 9	На основе имеющихся данных оценить объем программного кода. Произвести оценку трудоемкости и сроков реализации методами PERT и СОСОМО II.

9. Разработка бизнес-плана программного проекта	1	4	14	. Определить этапы реализации, деятельности, структур команды разработчиков и план исполнения отдельных работ (деятельностей).
---	---	---	----	--

4. Самостоятельная работа обучающегося

№	Виды самостоятельной работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации
Семестр: 7				
1	Шаблон проектирования (конструирования)	13, 2, 4, 5	30	4
Расчетно-графическое задание выполняется в виде разработки одного из шаблонов проектирования. Пояснительная записка должна содержать: - Учебно-методический материал по шаблону (учебники, учебные пособия, лекционный материал - краткое описание) - Материал по практике применения шаблона (форумы, тематические сайты) - Описание разработки - Результаты тестирования Все сторонние материалы должны быть снабжены корректно оформленными библио-графическими ссылками. Варианты заданий 1. Шаблон - прототип. Таблица определяется заголовком, который имеет вектор имен столбцов и вектор объектов-прототипов. Объект - прототип поддерживает интерфейс, позволяющий создать клон объекта, получить имя типа, загрузить значение из строки и вывести значение в строку. При добавлении строки в таблицу с заголовка снимается копия, в том числе и со всех объектов-прототипов. 2. Шаблон - итератор. Операции над итератором: установка на первый, последний, следующий, предыдущий, по логическому номеру, извлечение по ите: 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	1, 10, 11, 12, 13, 2, 3, 4, 5	38	0
Оформление результатов лаб. работ, подготовка к защите, дополнительное изучение лекционного материала по презентациям и метод. материалам, подобранным преподавателем: 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.				
3	Подготовка к аттестации	1, 10, 11, 12, 14, 2, 3, 4, 5	12	4
: 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.				

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 выставять 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: 7		
<i>Лабораторная:</i>	24	48
<i>РГЗ:</i>	6	12
<i>Экзамен:</i>	0	40
Контролирующие материалы приводятся в "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=vds000220170 . - Загл. с экрана."		

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	Экзамен
ОПК.3	з3. знать основы управления проектами создания и внедрения программных продуктов	+		+
ПК.2	у3. уметь оценивать метрики программных проектов различными методами	+		+
	ПК.10.В/ПК з2. знать основы объектно-ориентированного подхода к программированию		+	+
	ПК.10.В/ПК з4. знать основы системного программирования			+
	ПК.10.В/ПК з6. владеть средствами структурного и объектного моделирования программных систем	+	+	+
	ПК.10.В/ПК з7. знать методологии моделирования, используемые при проектировании информационных систем	+		+
	ПК.10.В/ПК з9. знать стандарты создания пользовательского интерфейса программных продуктов	+		+
	ПК.10.В/ПК у2. уметь строить функциональные модели информационных систем	+		+
	ПК.10.В/ПК у3. уметь строить модели потоков данных информационных систем	+		+
	ПК.10.В/ПК у4. уметь строить объектные модели информационных систем	+		+
	ПК.11.В/НПд з1. знать принципы формирования команды ИТ-проекта	+		+
	ПК.9.В/НИ з4. знать основы методологии создания программного обеспечения			+
	ПК.9.В/НИ з6. знать структурный и объектно-ориентированный подходы к проектированию информационных систем	+		+

	ПК.9.В/НИ у4. уметь обосновывать принимаемые проектные решения		+	+
--	--	--	---	---

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. Романов Е. Л. Архитектура и прикладные протоколы клиент-серверных приложений [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=vtls000226073. - Загл. с экрана.
4. Беркун С. Искусство управления IT- проектами / Скотт Беркун ; [пер. с англ. Н. Вильчинский]. - М. [и др.], 2010. - 431 с. : ил.
5. 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 StarUML

2 COCOMO® П.2000.4

9. Материально-техническое обеспечение

Презентационное оборудование

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

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

1. Обобщенная структура фонда оценочных средств учебной дисциплины

Обобщенная структура фонда оценочных средств по дисциплине Программная инженерия 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 способность разрабатывать бизнес-планы и технические задания на оснащение отделов, лабораторий, офисов компьютерным и сетевым оборудованием	з3. знать основы управления проектами создания и внедрения программных продуктов	Оценка трудоемкости и стоимости программного проекта Управление проектом как специфическая деятельность и разновидность менеджмент Управление рисками.	Отчет по лабораторной работе, разделы...	Экзамен, вопросы 18-20
ПК.10.В/ПК готовность к разработке моделей компонентов информационных систем, включая модели баз данных и модели интерфейсов "человек - электронно-вычислительная машина"	з2. знать основы объектно-ориентированного подхода к программированию	Диаграммы UML. Структуры и поведение Шаблон проектирования (конструирования)	РГЗ	Экзамен, вопросы 10-14
ПК.10.В/ПК	з4. знать основы системного программирования	Модульное тестирование Примеры проектирования ПС Разработка, управляемая тестированием (TDD) Тестирование, инспекция, отладка, верификация кода. Виды программных ошибок. Шаблоны проектирования и конструирования		Экзамен, вопросы 21-26
ПК.10.В/ПК	з6. владеть средствами структурного и объектного моделирования программных систем	Разработка архитектуры системы Шаблон проектирования (конструирования)	Отчет по лабораторной работе, РГЗ	Экзамен, вопросы 7-9
ПК.10.В/ПК	з7. знать методологии моделирования, используемые при проектировании информационных систем	Бизнес-аналитика и системная аналитика Жизненный цикл ПО Примеры проектирования ПС Разработка архитектуры системы Разработка модели прецедентов. Структура UML, статическая и динамическая составляющие модели	Отчет по лабораторной работе	Экзамен, вопросы 4-5,10-14

ПК.10.В/ПК	з9. знать стандарты создания пользовательского интерфейса программных продуктов	Проектирование графического интерфейса Разработка прототипа графического интерфейса. Структура UML, статическая и динамическая составляющие модели	Отчет по лабораторной работе	Экзамен, вопрос 9
ПК.10.В/ПК	у2. уметь строить функциональные модели информационных систем	Разработка архитектуры системы	Отчет по лабораторной работе	Экзамен, вопросы 6-8
ПК.10.В/ПК	у3. уметь строить модели потоков данных информационных систем	Анализ бизнес-процессов и разработка видения проекта Бизнес-аналитика и системная аналитика	Отчет по лабораторной работе	Экзамен, вопрос 4
ПК.10.В/ПК	у4. уметь строить объектные модели информационных систем	Анализ бизнес-процессов и разработка видения проекта Архитектура и проектирование ПС Бизнес-аналитика и системная аналитика Диаграммы UML. Структуры и поведение Моделирование предметной области Разработка архитектуры системы	Отчет по лабораторной работе	Экзамен, вопрос 4
ПК.11.В/НПД готовность к разработке конспектов и проведению занятий по обучению работников применению программно-методических комплексов, используемых на предприятии	з1. знать принципы формирования команды ИТ-проекта	Разработка бизнес-плана программного проекта Формирование команды	Отчет по лабораторной работе	Экзамен, вопрос 19
ПК.2/ПТ готовность к разработке компонентов аппаратно-программных комплексов и баз данных с использованием современных инструментальных средств и технологий программирования	у3. уметь оценивать метрики программных проектов различными методами	Оценка программного кода. Оценка трудоемкости и сроков проекта Оценка трудоемкости и стоимости программного проекта	Отчет по лабораторной работе	Экзамен, вопрос 18
ПК.9.В/НИ готовность обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их	з4. знать основы методологии создания программного обеспечения	Архитектура и проектирование ПС Жизненный цикл ПО Примеры проектирования ПС Тестирование, инспекция, отладка, верификация кода. Виды программных ошибок. Унифицированный процесс UP Управление требованиями Экстремальное и гибкое программирование/проектиро		Экзамен, вопросы 2,3,15

корректности и эффективности		вание		
ПК.9.В/НИ	з6. знать структурный и объектно-ориентированный подходы к проектированию информационных систем	Анализ бизнес-процессов и разработка видения проекта Архитектура и проектирование ПС Бизнес-аналитика и системная аналитика Диаграммы UML. Структуры и поведение Примеры проектирования ПС Структура UML, статическая и динамическая составляющие модели Шаблоны проектирования и конструирования	Отчет по лабораторной работе	Экзамен, практический вопрос
ПК.9.В/НИ	у4. уметь обосновывать принимаемые проектные решения	Управление требованиями Шаблон проектирования (конструирования)	РГЗ	Экзамен, практический вопрос

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по дисциплине проводится в 7 семестре - в форме экзамена, который направлен на оценку сформированности компетенций ОПК.3, ПК.10.В/ПК, ПК.11.В/НПд, ПК.2/ПТ, ПК.9.В/НИ.

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

В 7 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.10.В/ПК, ПК.11.В/НПд, ПК.2/ПТ, ПК.9.В/НИ, 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 экзамена

по дисциплине «Программная инженерия», 7 семестр

1. Методика оценки

Экзамен проводится в устной форме по билетам. Билет формируется по следующему правилу: первый (теоретический) вопрос выбирается из диапазона, второй вопрос – практическая задача на проектирование. Выбирается вариант предметной области и одна из компонент проектирования. В ходе экзамена преподаватель вправе задавать студенту дополнительные вопросы по другим компонентам проектирования выбранной предметной области.

Форма экзаменационного билета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № _____

к экзамену по дисциплине «Программная инженерия»

Вопрос 1. Оценка программного кода. Метрики кода. Метрики количественные, сложности потока управления и потока данных, метрики ООП, прагматические метрики. Средства оценки качества программного кода.

Вопрос 2. Система мониторинга междугородных автобусных перевозок. Карта автодорог, маршрут, график движения по маршруту, планирование рейсов, GPS-навигация транспортных средств, отслеживание графика движения по маршруту, обработка аварийных ситуаций. Модель классов предметной области

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

2. Критерии оценки

- Ответ на экзаменационный билет считается **неудовлетворительным**, если студент не ответил на теоретический или практический вопрос, оценка составляет <30% базовой.
- Ответ на экзаменационный билет (тест) засчитывается на **пороговом** уровне, если студент при ответе на теоретический вопрос дает определение основных понятий, может показать причинно-следственные связи явлений, при решении практической задачи допускает непринципиальные ошибки, например, структурные,

оценка составляет 30-60% от базовой.

- Ответ на экзаменационный билет засчитывается на **базовом** уровне, если студент при дает развернутый ответ на теоретический вопрос с несущественными ошибками и решает практическую задачу с несущественными ошибками, оценка составляет 60-80% от базовой.
- Ответ на экзаменационный билет (тест) билет засчитывается на **продвинутом** уровне, если студент дает развернутый ответ на теоретический вопрос и дополнительные вопросы, решает практическую задачу и вносит изменения в решение по дополнительным требованиям, оценка составляет 80-100% от базовой.

3. Шкала оценки

В общей оценке по дисциплине экзаменационные баллы учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины, т.е. 40 баллов за экзамен.

4. Вопросы к экзамену по дисциплине «Программная инженерия»

Теоретический вопрос

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.
18. Метрика проекта. Оценка сроков на основании трудоемкости (по Боэму). Оценка на основе собственного опыта. Метод PERT. Оценка на основе функциональных точек. Оценка по отраслевым данным. Метод COSOMO II

19. Управление проектами как инженерная дисциплина. Особенности управления программными проектами. Роль и место УПП в программной инженерии (ПИ). Компоненты организационного (менеджмент) и технологического (исполнение) планирования в УПП.
20. Определение и характеристики риска. Шкалы оценивания последствий и вероятности. Способы идентификации. Реакция на риски. Наиболее вероятные риски по Бозму и Архипенкову. Качественные оценки рисков. Количественные оценки: анализ чувствительности, дерево решений, имитационное моделирование. Управление, направленное на снижение рисков. Вероятностный характер оценивания, его природа. Последствия «агрессивного» планирования. Исходные данные для оценивания, характеристики проекта, используемые в оценивании. Оценка сроков на основании трудоемкости (по Бозму).
21. Понятие программной ошибки. Философия ошибок. Классификационные характеристики ошибок. Отладка, инспекция и тестирование как этапы поиска ошибок. Ошибки вычислений и преобразований. Ошибки структурирования кода. Ошибки форматов входных данных. Ошибки форматов внутренних данных и соглашений по данным. Ошибки сборки, конфигурирования, размещения. Ошибки использования ресурсов. Ошибки, связанные с ограничениями по ресурсам. Ошибки реактивности и производительности. Ошибки параллелизма и синхронизации. Ошибки распределенных систем и протоколов. Ошибки пользовательского интерфейса.
22. Структурное тестирование. Тестирование операторов, условий (решений) и путей. Комбинационное тестирование. Пример. Функциональное тестирование. Классы эквивалентности по входным данным. Проектирование тестового покрытия. Классы эквивалентности по граничным условиям. Примеры.
23. Отладка и инспекция программного кода. Сущность отладки. Приемы отладки. Инспекция как неформальный анализ программного кода. Логический и временной анализ. Методы инспектирования. TDD – разработка, управляемая тестированием.
24. Шаблоны проектирования. Производящие шаблоны builder, factory, prototype, singleton. Шаблоны проектирования. Структурные шаблоны adapter, bridge, composite, decorator, facade, flyweight, proxy
25. Шаблоны проектирования. Поведенческие шаблоны command, iterator, mediator, snapshot, observer, state, strategy, method template, visitor.
26. Шаблоны проектирования. Системные шаблоны model-view-controller (MVC), session, transaction
27. Шаблоны проектирования. Шаблоны параллелизма Single Threaded Execution, Two-phase Termination, Asynchronous Task, Lock Object, Read/Write Lock, Scheduler, Double Buffering, Producer-consumer

Практический вопрос

Для произвольно выбранного варианта лабораторных работ выполнить черновую разработку одной из компонент:

1. Модель классов предметной области

2. Общая архитектура системы
3. Диаграмма состояний одной из сущностей
4. Диаграмма деятельности, потоков данных для бизнес-процесса
5. Перечень требований (функциональные, бизнес-требования, системные требования, бизнес-правила, атрибуты качества) в виде дерева требований

Варианты заданий к практическому вопросу – предметная область проектирования.

1. Система продажи билетов в кинотеатре. Клиент, кассир, смена, билетер, администратор, планирование сеансов, продажа билетов кассиром, бронирование и продажа через Интернет, финансовые отчеты, план зала.
2. Система продажи театральных билетов. Приложение кассира - множество точек продажи, приложение распространителя, бронирование через Интернет, связь с платежными системами, план зала (мета-уровень описания), спектакли, репертуар.
3. Система автоматизации диспетчерской службы такси. Диспетчер, водитель, клиент, директор, прием заказов, ведение очередей, ручное распределение заказов, приложение водителя, мониторинг прохождения заказа.
4. Система автоматизированного заказа такси через Интернет. Серверное приложение для автоматического распределения заказов с учетом нагрузки, web-приложение и мобильное приложение для клиентов, приложение водителя, приложение администратора для форсмажорных и конфликтных ситуаций. Автоматическое распределение заказов на основе расстояния для клиента и других критериев, предложение свободных заказов водителю, голосование за заказ, мониторинг прохождения заказа. Виды адресов с привязкой к GPS-координатам: почтовый, место, корпоративный (фирма, организация).
5. Система ведения корпоративной адресной базы для мобильных клиентов. Типы адресов: служебный - подразделение, корпус, кабинет, домашний - почтовый. Типы контактов: электронная почта, телефон, социальная сеть, адрес. Административная структура организации. Хранение списка контактов, обмен контактами, иерархическая многомерная адресная книга с каталогами (тегами), общая и личная адресные книги.
6. Мессенджер мобильных клиентов аналогичный WhatsUp. Регистрация по номеру мобильного телефона. Передача сообщений, файлов, синхронизация адресных книг, иерархическая многомерная адресная книга с каталогами (тегами), поиск по общей адресной книге, личная адресная книга с собственной системой каталогов (тегов)
7. Мессенджер с прямой связью клиентов. Сервер контактов, регистрация, сохранение IP-адресов клиента и статистики пребывания, авторизация с сообщением текущего IP-адреса, дозвон по списку IP-адресов, прямая связь с передачей сообщений и файлов, локальные адресные книги, обмен адресами.
8. Система заказных грузоперевозок по городу. Клиент, диспетчер, магазин, водитель. Прием и оформление заказов, отчеты и сопроводительные документы, распределение заказов диспетчером. Два вида заказов: точка-точка и развоз товаров со склада по клиентам. Приложение водителя: просмотр заказов, мониторинг проведения заказа, планирование последовательности

исполнения для развоза, времени доставки. Приложение диспетчера: прием и оформление заказа, распределение, планирование доставки. Параметры заказа – вес и габариты грузов. Транспортные средства и водители. Оплата доставки авансом и при выполнении заказа.

9. Система продажи билетов на междугородные автобусы. Планирование рейсов, расписание, чартерные рейсы, типы автобусов, планы рассадки, водители, кассиры, смены, визуализация рассадки, приобретение билетов в кассе и в кассовых терминалах, бронирование через Интернет, сводные отчеты по маршруту и дате.
10. Система мониторинга междугородных автобусных перевозок. Карта автодорог, маршрут, график движения по маршруту, планирование рейсов, GPS-навигация транспортных средств, отслеживание графика движения по маршруту, обработка аварийных ситуаций.
11. Система планирования междугородных транспортных перевозок. Транспортные средства - тип, тоннаж, вместимость. Населенные пункты, дорожная сеть, прием заявок, поиск подходящего транспорта с учетом его местонахождения и порожнего прогона, составление маршрута, оформление и проводка заявок, отчеты по периоду времени, транспортному средству, водителю.
12. Система мониторинга междугородных транспортных перевозок. Населенные пункты, дорожная сеть, маршрут, планирование движения по маршруту, GPS-навигация транспортных средств, отслеживание графика движения по маршруту, обработка аварийных ситуаций, отслеживание заправки, отчеты по расходу топлива и рабочему времени водителей.
13. Справочная система наличия товаров. Многоуровневая система категорий и марок товара. Мета-система классификационных признаков и их значений, например, вес, цвет, производитель, объем памяти, наличие GPS и т.п.. Торговые сети, торговые точки с привязкой к GPS-координатам. Ассортимент в торговой точке, количество товара. Приложение пользователя: поиск по местоположению, по условиям, сформированным для признаков. Приложение торговой точки – редактирование ассортимента. Приложение администратора – редактирование категорий и марок, классификационных признаков.
14. Логистическая система интернет-магазина с пунктами выдачи и доставкой по городу. Прием заказов, оформление заявок поставщикам, уведомление клиентов, отслеживание работы курьеров, отчеты по пунктам выдачи. Многоуровневая система категорий и марок товара. Мета-система классификационных признаков и их значений, например, вес, цвет, производитель, объем памяти, наличие GPS и т.п.. Наличие товара на складе. Отслеживание балансов по каждому виду товара: заказано, наличие на складе, в заявках к поставщикам. Принятие товара на складе, формирование комплектов заказов для пунктов выдачи и курьеров.
15. Система электронного документооборота учебного процесса в ВУЗе. Структура учебного заведения, факультеты, кафедры, группы, студенты, преподаватели. Сторонние организации. Приказы о зачислении/отчислении, назначении тем выпускных и курсовых работ, распределение на практику, распоряжения по подразделениям. Прохождение приказа: создание, визирование, утверждение, нумерация, рассылка.

16. Система книговыдачи школьной библиотеки с использованием QR-кодов. Систематический и алфавитный каталог книг с учетом экземпляров, рекомендованные учебники по предметам, структура учебного процесса: класс, ученик, предметы, преподаватели, сторонние лица. Формирование комплектов учебников, выдача литературы на абонемент. QR-коды читательских билетов, экземпляров книг, выдача и прием, ведение формуляра с историей, уведомление о просрочках.
17. Система книговыдачи библиотеки с использованием QR-кодов. Алфавитный каталог книг с учетом экземпляров, многомерный иерархический тематический каталог или система тегов. Выдача литературы на абонемент. QR-коды читательских билетов, экземпляров книг, ведение формуляра с историей, уведомление о просрочках. Ведение очередей на дефицитные книги - артефакты, уведомления об очередности.
18. Система управления кафе/баром. План зала, закрепление официантов за столиками, планирование смен. Меню – категории, позиции, описание, привязка к кухне или бару. Проведение заказа: закрепление столика за официантом, выбор по меню, частичный заказ, заявки в бар и на кухню, повторение заказа, итоговый расчет. Приложение посетителя, официанта, бармена – планшет, администратора – desktop.
19. Система бронирования мест для клубных мероприятий. План концертов. Анонсы. Стоимость столиков. План зала - мета-уровень описания, настройка под конкретный клуб. Билеты – столики, танцпол. Электронная предоплата. Бронирование. Приложение кассира. Мобильное или web-приложение клиента.
20. Система бронирования мест в гостинице. Мета-уровень описания конкретной гостиницы – расположение и типы номеров, поэтажные планы, список услуг, фото общие и отдельных номеров, расценки. Бронирование через интернет, визуализация свободных/занятых, расчет стоимости, квитанции, отчеты по периодам. Бронирование индивидуальное и групповое. Балансы по занятым, свободным и забронированным номерам по датам и категориям. Заселение, продление проживания, дополнительные услуги, частичный и итоговый расчет.
21. Система планирования взаимосвязанных работ. Учет сотрудников и их занятости запланированными работами. Определение работ в виде цепочек заданий с параллельными ветвями. Распределение заданий по работникам с учетом их занятости. Сдвиг сроков зависимых работ при задержке выполнения одной из них. Приложения сотрудника, руководителя подразделения, генерация отчетов.
22. Система мониторинга обслуживания по заявкам. Система с предварительным сбором заявок и обслуживанием. Категории и виды работ, исполнители, возможность выполнения ими работ по категориям и видам (квалификация). Прием заявок, планирование исполнения, распределение по исполнителям. Оперативное планирование времени исполнения заявок, отслеживание времени исполнения, коррекция времени при задержках с уведомлением клиентов, отказы. Мобильный клиент сотрудника, приложение диспетчера.
23. Система поддержки технологии SCRUM для удаленной работы. Поддержка основных элементов технологии SCRUM - исполнители (команда), лидер, собственник проекта, истории пользователей (задачи), спринты, беклоги,

мониторинг исполнения задач, диаграммы сгорания. Митинги, покер-планирование в режиме чата.

24. Система многомерной организации документов. В качестве документов могут выступать короткие заметки (записи в БД), файлы документов, изображений, звуковых. Поддерживается дерево редактирования версий файла. Создается произвольное количество систем классификаций (каталогов ссылок на файлы) по набору ключевых параметров, например, изображения могут классифицироваться по размеру, цветовой палитре, содержанию, наличию определенных предметов, звуковые файлы по исполнителю, стилю, наличию музыкальных инструментов.

Паспорт расчетно-графического задания (работы)

по дисциплине «Программная инженерия», 7 семестр

1. Методика оценки

Расчетно-графическое задание выполняется в виде разработки одного из шаблонов проектирования. Пояснительная записка должна содержать:

- Учебно-методический материал по шаблону (учебники, учебные пособия, лекционный материал – краткое описание)
 - Материал по практике применения шаблона (форумы, тематические сайты)
 - Описание разработки
 - Результаты тестирования
- Все сторонние материалы должны быть снабжены корректно оформленными библиографическими ссылками.

При защите расчетно-графического задания (работы) оцениваются следующие виды работ:

- Постановка и анализ задачи;
- Разработка структур данных для представления предметной области и алгоритмов;
- Объем и качество программного кода;
- Тестирование программного кода;
- Оформление текста программы и результатов работы

2. Критерии оценки

Согласно положению о балльно-рейтинговой системе НГТУ, базовый балл рейтинга за РГР определен в рабочей программе (12 баллов). Соответственно, критерий оценки определяется в процентах к этому баллу:

- РГР считается **невыполненной**, если отсутствуют необходимые разделы описания программы, либо программа не работает, либо студент не в состоянии объяснить принципов ее работы (алгоритма) – оценка составляет <25% базовой;
- РГР засчитывается на **пороговом** уровне, разработка не полностью соответствует заданию, имеются ошибки при тестировании, пояснительная записка оформлена со значительными структурными, стилистическими и грамматическими ошибками - оценка составляет не более 50% базовой
- РГР засчитывается на **базовом** уровне, если функционал разработки соответствует заданию, отсутствуют ошибки тестирования, пояснительная записка оформлена в целом грамотно - оценка составляет 50-80% базовой
- РГР засчитывается на **продвинутом** уровне, если функционал реализован полностью и эффективно, в пояснительной записке отражены все аспекты

структурного, функционального проектирования и тестирования - 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 (12 баллов).

2. Примерный перечень тем РГЗ(Р)

1. *Шаблон MVC. Сетевая (локальная) игра типа «Морской бой», «Домино».* Модель хранит структуру данных игры - расстановку кораблей, и над ней выполняются методы по управлению игрой. Контроллер определяет порядок ходов, проверяет возможность выполнения хода. Внешнее представление (view) связано с контроллером двунаправленным интерфейсом: контроллер управляет отображением элементов игрового поля, выводит текстовые сообщения, получает от представления события – клик по элементу поля, начало новой игры, завершение и т.п.. Варианты игры:
 - локальная: одна модель, один контроллер, два представления для игроков;
 - сетевая: одна модель, один контроллер, два представления для игроков. Представления связываются с приложением, содержащим контроллер и модель. Контроллер обеспечивает множественные соединения и синхронизацию;
 - сетевая: две модели, два контроллера, два представления. Контроллеры поддерживают соединение и передают команды: сделан ход, синхронизация моделей, сброс и начальная установка игры.
2. *Шаблон MVC (классический).* Представление имеет ряд текстовых полей, отображающих параметры модели физического объекта. На каждый параметр представление отдельно подписывается к модели на событие, связанное с его изменением. Интерфейс подписки идентичен для всех параметров. Для каждого параметра создается оригинальный контроллер, обрабатывающий команду изменения значения параметра со стороны представления. Модель описывается набором зависимостей, параметры которой могут быть *входными, выходными (результатами) и выходными с возможностью задания начальных значений (инициализацией).*
3. *Шаблон – прототип. Класс таблицы с произвольной структурой столбцов.* Хранимые данные разных типов - целые, вещественные, дата, время, GPS-координаты создаются на основе абстракции данных с функционалом: имя класса, парсинг из строки и вывод в строку, клонирование, сравнение и сложение с объектом того же типа. Строка таблицы определяется набором объектов-прототипов для столбцов. Сама строка также клонируется. Таблица состоит из вектора строк-имен, строки-прототипа и строк самой таблицы. Функционал: создание таблицы, добавление столбцов, добавление строк, сортировка по столбцу с заданным номером, сложение строк, сохранение и

- загрузка из файла. Программный и интерфейс и оконное приложение. Тестирование на больших данных - импорт из Excel, генерация тестовых таблиц.
4. *Шаблон – приспособленец. Работа с деревом версий текстового файла.* Файл редактируется по словам. Класс словаря содержит хэш-таблицу адаптеров со ссылками на оригинальные слова в виде *ключ – само слово*. Адаптер содержит количество ссылок на слово из всех версий текста. Каждая версия текста – вектор ссылок на адаптеры. При добавлении или вставке слова в версию текста оно ищется в фабрике, при нахождении счетчик в адаптере увеличивается на 1. Иначе добавляется в словарь с новым адаптером. При удалении счетчик ссылок уменьшается. Изменение слова рассматривается как последовательность операций удаления и вставки. Вся структура данных сериализуется в файл. Протестировать шаблон на сказке *Репка*.
 5. *Шаблон – композиция для древовидной системы. Графический редактор с группировкой/разгруппировкой элементов, изменением размеров, переносом на передний/задний план, перемещением объектов, сохранением картинки в файл.* Абстрактный класс элемента, группы элементов и конкретных графических объектов - окружность, полигон, строка текста. Ограничивающий прямоугольник, селекция объектов по точке и прямоугольнику.
 6. *Шаблон Command.* Группа команд обработки объекта с общим интерфейсом *Do/ReDo/Undo*. Производный класс запоминает параметры, необходимые для выполнения прямой и обратной команды. Класс – менеджер команд поддерживает очередь команд ограниченной длины, текущий обрабатываемый объект, методы *Do/ReDo/Undo*, выбирая их из очереди и вызывая соответствующие методы в объектах-командах. **Графический редактор** набором команд редактирования графических объектов - создать объект, переместить, изменить размер, удалить, переместить на передний и задний план.
 7. *Шаблон – пул потоков.* Класс потока представляет собой поток, запускаемый при создании объекта. Содержит ссылку на исполняемый код через *Runnable*, код завершения и ссылку на менеджер пула. Код потока содержит цикл, в котором засыпает, после пробуждения исполняемый назначенный код, уведомляет менеджер о своем завершении и засыпает. Менеджер потоков содержит вектор потоков-исполнителей. При обращении к менеджеру с запросом, содержащим код исполнения и код завершения, выбирается поток из пула, заполняется данными и пробуждается. Если свободного потока нет, то запрос ставится в очередь. Провести сравнительное тестирование для потока запросов с пулом и при запуске потоков обычным образом.
 8. *Шаблон – прокси (фильтр).* Класс с интерфейсом текстового потока при конструировании делегируется к однотипному объекту источнику и **отфильтровывает набор слов**, передаваемый при конструировании, используется внутренняя очередь символов для отложенного распознавания. Протестировать на цепочке фильтров для разных наборов слов.
 9. *Шаблон – сессия* для соединения на сокетах в виде библиотеки для клиента/сервера. Клиент и сервер используют синхронный обмен *запрос-ответ*. При первоначальном установлении соединения клиент получает уникальный идентификатор, сервер создает дескриптор соединения. Клиент нумерует передаваемые сообщения, сервер сохраняет номер и ответ на последнее

переданное сообщение. Сервер периодически или после каждого изменения сохраняет дескрипторы в файл. Клиент также запоминает в файле идентификатор сессии, номер и последнее переданное сообщение. Обеспечить сохранность последовательности (отсутствие пропадания и дублирования) сообщений при перезагрузке клиента и сервера. Реализовать модель банкомата и платежной системы.

10. *Модель параллельных запросов к серверу от группы потоков.* Множество потоков может посылать независимые запросы к серверу через единственное соединение. Запрос ставится в очередь, нумеруется, клиентский процесс засыпает. Поток передачи от клиента выбирает сообщения из очереди и передает в соединение. Поток приема на сервере, приняв очередное сообщение, запускает поток исполнения запроса, по завершении которого в очередь ответов ставится ответ, в котором сохраняется порядковый номер запроса. Поток передачи ответов на сервере выбирает сообщения из очереди и передает в соединение. Поток приема ответов определяет по номеру в сообщении поток, передавший запрос и пробуждает его. Ответное сообщение находится в запросе, переданном потоком, и выводится им. Промоделировать *группу потоков, посылающих случайные слова, которые сервер переворачивает со случайной задержкой.*
11. *Шаблон – кэш объектов.* Разработать класс - кэш объектов с возможностью изменения размера кэша, сбора статистики и применения различных стратегий вытеснения (FIFO, LRU, RAND). Использовать для кэширования слов текстового файла при последовательном чтении. Кэш – хеш-таблица с ключом-словом и необходимыми параметрами для моделирования, например, номер последнего обращения для LRU. При чтении очередного слова проверяется его наличие в кэше. При отсутствии производится замещение. Для выбранного файла строится зависимость доли попадания в кэш как функция от размера кэша и способа вытеснения. Сравнить результаты для файлов с разным содержимым - художественное произведение, технический текст.
12. *Шаблон – итератор.* Операции над итератором: установка на первый, последний, следующий, предыдущий, по логическому номеру, извлечение через итератор, вставка на позицию итератора, замещение, удаление. Структура данных: двухуровневый массив ссылок (двумерный массив), двоичное дерево, дерево с данными в конечных вершинах, список с массивом ссылок [9]. Использование нескольких итераторов. *Замечание по теме:* для корректной реализации операции удаления в основном классе использовать вектор созданных итераторов. При удалении одним из них элемента остальные, которые на него ссылаются, генерируют исключение при очередном обращении. Рассмотреть другие варианты корректного разделения.

Разработка классов, использующих внутренние потоки для промежуточной буферизации данных. Для моделирования прикладных процессов использовать потоки, которые засыпают на случайный момент времени, после чего читают/записывают очередную порцию данных постоянного или случайного размера. Предусмотреть сбор статистики в классах буферизации – средний объем данных в буфере.

13. Класс буферизованного ввода в реальном времени. При конструировании получает параметр – физический поток данных с интерфейсом *InputStream*.

Использует внутренний циклический буфер или односвязный список блоков, содержащих массив байтов фиксированной размерности (буферный пул). Создает поток, который читает байты из входного потока и записывает в циклический буфер. При заполнении циклического буфера засыпает. Метод чтения извлекает из циклического буфера очередной байт, возвращает -1 при окончании данных в потоке-источнике, блокируется при отсутствии данных в циклическом буфере.

14. Класс – *PipedStream* с циклическим буфером данных. Класс *PipedOutputStream* имеет циклический буфер, в который пишет поток байтов, блокируя текущий поток при заполнении буфера. Класс *PipeInputStream* получает ссылку на *PipedOutputStream* и при чтении данных либо блокируется при их отсутствии, либо извлекает данные, деблокируя поток записи.
15. Класс отложенной записи. При открытии файла классом с присоединенным интерфейсом *OutputStream* создается *ByteOutputStream*, в который пишутся данные потока. При закрытии объект, содержащий имя файл и байтный массив ставится в очередь, из которой фоновый поток их извлекает и пишет файлы. Сравнить среднее время записи в обычный и *отложенный файл*.