



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение
высшего образования
**«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ИГУ»)**

Институт математики и информационных технологий
Кафедра алгебраических и информационных систем



Рабочая программа дисциплины (модуля)

Б1.В.05 Проектирование информационных систем

Направление подготовки информационные технологии	02.03.02	Фундаментальная	информатика	и
Направленность (профиль) подготовки информационные технологии		Фундаментальная	информатика и	
Квалификация выпускника	бакалавр			
Форма обучения	очная			

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

Цель

Ознакомление студентов с различными приемами проектирования информационных систем, проектными решениями, формирование практических умений и навыков, необходимых для приобретения квалификации бакалавра.

Задачи:

- ознакомление с особенностями организации и проведения всех фаз жизненного цикла разработки информационных систем;
- формирование практических умений и навыков реализации информационных систем на языке Java;
- формирование навыков работы на различных этапах жизненного цикла разработки информационных систем;
- создание базы для подготовки выпускной квалификационной работы.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

2.1. Учебная дисциплина (модуль) относится к части программы, формируемой участниками образовательных отношений, и изучается на третьем курсе.

2.2. Для изучения данной учебной дисциплины (модуля) необходимы знания, умения и навыки, сформированные дисциплинами Программирование, Базы данных.

2.3. Перечень последующих учебных дисциплин, для которых необходимы знания, умения и навыки, формируемые данной учебной дисциплиной: Архитектура программного обеспечения.

3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс освоения дисциплины направлен на формирование компетенций (элементов следующих компетенций) в соответствии с ФГОС ВО по соответствующему направлению подготовки.

Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Компетенция	Индикаторы компетенций	Результаты обучения
ПК-6 Способность к анализу требований и разработке вариантов реализации информационной системы; способность к оценке качества, надежности и эффективности информационной системы в конкретной профессиональной сфере.	ИДК ПК6.1 Способен выполнять анализ требований и разрабатывать варианты реализации информационной системе в рамках существующих методологий	Знает приемы объектно-ориентированного проектирования Умеет применять методы объектно-ориентированного проектирования на примере прикладных объектов конфигурации. Владеет приемами предметно-ориентированного проектирования для различных областей предметной области
	ИДК ПК6.2 Способен составлять спецификации для разрабатываемых информационных систем	Знает принципы и виды документирования приложений Умеет составлять базовые спецификации для определения требований приложений

		Владеет навыками описания спецификаций на основе прецедентов использования информационной системы
	ИДК ПК6.3 Способность к оценке качества, надежности и эффективности информационной системы в конкретной профессиональной сфере	Знает виды и приемы тестирования программного обеспечения Умеет проводить модульное и сценарное тестирование программного обеспечения Владеет навыками оценки эффективности оценки реализованных проектных решений
ПК-1 Способность к установке, администрированию программных систем; к реализации технического сопровождения информационных систем; к интеграции информационных систем с используемыми аппаратно-программными комплексами.	ИДК ПК1.1 Способен устанавливать и администрировать программные системы	Знает принципы внедрения и сопровождения информационных систем Умеет адаптировать прикладные решения под нужды конкретного пользователя Владеет навыками внедрения прикладного решения под нужды конкретного пользователя
ПК-5 Способность собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности; способность к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности, в том числе с применением технологий искусственного интеллекта	ИДК ПК5.1 Способен собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности	Знает методы выявления требований к информационной системе Умеет документировать процесс выявления требований, анализа и проектирования программных систем выполнять работы на стадии проектирования проекта Владеет навыками работы с инструментальными средствами проектирования программного обеспечения
	ИДК ПК5.2 Способен к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности	Знает приемы разработки новых технологических решений в сфере профессиональной деятельности Умеет выбирать приемы и технологии для разработки новых технологических решений Владеет навыками использования технологий для разработки новых решений в сфере профессиональной деятельности

4. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

Объем дисциплины составляет 9 зачетных единиц, 324 часа, в том числе 35 часов на контроль, практическая подготовка 66.
Форма промежуточной аттестации: 5 семестр - зачет, 6 семестр - экзамен.

4.1. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ, СТРУКТУРИРОВАННОЕ ПО ТЕМАМ, С УКАЗАНИЕМ ВИДОВ УЧЕБНЫХ ЗАНЯТИЙ И ОТВЕДЕННОГО НА НИХ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ

№ п/п	Раздел дисциплины/темы	Семестр	Виды учебной работы, включая самостоятельную работу обучающихся и трудоемкость (в часах)					Формы текущего контроля успеваемости
			Контактная работа преподавателя с обучающимися			Самостоятельная работа + контроль		
			Лекции	Семинарские (практические занятия)	Контроль обучения			
1	Тема 1. Диаграммы определения требований	5	2	2	0,5	4		лаб.
2	Тема 2. Диаграммы классов	5	2	4	0,5	8		тест
3	Тема 3 Порядок использования шаблонов проектирования	5	1	2	1	4		лаб.
4	Тема 4. Примеры шаблонов проектирования	5	5	8	2	14		лаб.
5	Тема 5. Принципы проектирования	5	3	8	1	8		тест
6	Тема 6. Основы тестирования программного обеспечения	5	1	2	1	4		лаб.
7	Тема 7. Виды тестирования программного обеспечения	5	1	4	1	4		лаб.
8	Тема 8. Место тестирования в цикле разработки	5	1	4	1	4		тест
9	Тема 9. Библиотека объектно-реляционного отображения Hibernate	6	6	6	2	18	5	контр. раб.
10	Тема 10 Фреймворк Spring	6	6	6	1	18	6	лаб.

11	Тема 11. Библиотека Spring MVC	6	8	8	2	30	6	лаб.
12	Тема 12. Библиотеки фреймворка Spring	6	6	6	2	23	6	контр. раб.
13	Тема 13. Модели и процессы жизненного цикла ПО	6	2	2	1	6	4	лаб.
14	Тема 14. Метрики процессов программного обеспечения	6	2	2	1	6	4	лаб.
15	Тема 15. Документирование программного обеспечения	6	2	2	1	6	4	лаб.
Итого часов			48	66	18	157/35		

4.2. ПЛАН ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Семестр	Название раздела, темы	Самостоятельная работа обучающихся			Оценочное средство	Учебно-методическое обеспечение самостоятельной работы
		Вид самостоятельной работы	Сроки выполнения	Затраты времени (час.)		
5	Тема 1. Диаграммы определения требований	Выполнение домашней работы	5-я неделя семестра	4	Домашняя работа	Лекции и материалы курса на платформе ИОС Domic
5	Тема 2. Диаграммы классов	Выполнение домашней работы	5-я неделя семестра	8	Домашняя работа	Лекции и материалы курса на платформе ИОС Domic
5	Тема 3 Порядок использования шаблонов проектирования	Выполнение домашней работы	7-я неделя семестра	4	Домашняя работа	Лекции и материалы курса на платформе ИОС Domic
5	Тема 4. Примеры шаблонов проектирования	Выполнение домашней работы	12-неделя семестра	14	Домашняя работа	Лекции и материалы курса на платформе ИОС Domic
5	Тема 5. Принципы проектирования	Выполнение домашней работы	13-неделя семестра	8	Домашняя работа	Лекции и материалы курса на платформе ИОС Domic
5	Тема 6. Основы тестирования программного обеспечения	Выполнение домашней работы	17-неделя семестра	4	Домашняя работа	Лекции и материалы курса на платформе ИОС Domic
5	Тема 7. Виды тестирования программного обеспечения	Выполнение домашней работы	17-неделя семестра	4	Домашняя работа	Лекции и материалы курса на платформе ИОС Domic

Семестр	Название раздела, темы	Самостоятельная работа обучающихся			Оценочное средство	Учебно-методическое обеспечение самостоятельной работы
		Вид самостоятельно работы	Сроки выполнения	Затраты времени (час.)		
5	Тема 8. Место тестирования в цикле разработки	Выполнение домашней работы	17-неделя семестра	4	Домашняя работа	Лекции и материалы курса на платформе ИОС Domic
6	Тема 9. Библиотека объектно-реляционного отображения Hibernate	Выполнение домашней работы	5-я неделя семестра	23	Домашняя работа	Лекции и материалы курса на платформе ИОС Domic
6	Тема 10 Фреймворк Spring	Выполнение домашней работы	8-я неделя семестра	24	Домашняя работа	Лекции и материалы курса на платформе ИОС Domic
6	Тема 11. Библиотека Spring MVC	Выполнение домашней работы, выполнение этапов курсовой работы	15-я неделя семестра	36	Домашняя работа, выполнение этапов курсовой работы	Лекции и материалы курса на платформе ИОС Domic
6	Тема 12. Библиотеки фреймворка Spring	Выполнение домашней работы, выполнение этапов курсовой работы	16-я неделя семестра	29	Домашняя работа, выполнение этапов курсовой работы	Лекции и материалы курса на платформе ИОС Domic
6	Тема 13. Модели и процессы жизненного цикла ПО	Выполнение домашней работы	17-я неделя семестра	10	Домашняя работа	Лекции и материалы курса на платформе ИОС Domic
6	Тема 14. Метрики процессов программного обеспечения	Выполнение домашней работы	17-я неделя семестра	10	Домашняя работа	Лекции и материалы курса на платформе ИОС Domic
6	Тема 15. Документирование программного обеспечения	Выполнение домашней работы	17-я неделя семестра	10	Домашняя работа	Лекции и материалы курса на платформе ИОС Domic
Общая трудоемкость самостоятельной работы по дисциплине (час)				192		
Из них объем самостоятельной работы с использованием электронного обучения и дистанционных образовательных технологий (час)				192		

4.3. СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. Унифицированный язык моделирования информационных систем

Тема 1. Диаграммы определения требований

Язык UML. Его назначение и сферы применения. Унифицированность языка и его преимущества. Особенности графического изображения вариантов использования и актеров. Основные отношения между графическими элементами, их стереотипы. Особенности концептуального моделирования бизнес-систем в форме диаграмм вариантов использования. Рассмотрение требований в форме диаграмм вариантов использования. Сценарии вариантов использования, их графическая интерпретация. Диаграмма деятельности и особенности ее построения. Концепция маркеров и передача управления от одного действия к другому. Действия и переходы на диаграмме деятельности. Ветвление и распараллеливание процессов на диаграмме деятельности. Особенности изображения объектов на диаграмме деятельности.

Тема 2. Диаграммы классов

Графическое изображение класса, его атрибутов и операций. Конкретные и абстрактные классы. Видимость и кратность атрибутов и операций классов. Отношение ассоциации, варианты его графического изображения. Отношение обобщения классов. Наследование атрибутов и операций классов. Отношения агрегации и композиции, их семантические особенности.

Раздел 2. Шаблоны проектирования

Тема 3 Порядок использования шаблонов проектирования

Назначение паттернов. Их структура. Необходимость их использования. Примеры реализации паттернов. Поиск объектов реализации. Спецификация интерфейсов объекта. Спецификация реализации объектов. Механизмы повторного использования программного кода.

Тема 4. Примеры шаблонов проектирования

Порождающие шаблоны `abstract factory`, `builder.factory method`. Структура, описание и область применения паттернов. Структурные шаблоны `adapter`, `composite`. Паттерн `decorator`. Структура, описание и область применения паттернов. Поведенческие шаблоны: `chain of responsibility`, `command`, `strategy`. Структура, описание и область применения паттернов.

Тема 5. Принципы проектирования

Выбор и использование шаблонов проектирования. Примеры и рекомендации использования комбинации шаблонов для реализации гибкой архитектуры информационной системы. Принципы проектирования SOLID. Принципы построения гибкой архитектуры программных модулей.

Раздел 3. Тестирование программного обеспечения

Тема 6. Основы тестирования программного обеспечения

Понятие тестирования и отладки. Трудности тестирования ПО. Статическое и динамическое тестирование. Фазы тестирования. Понятие набора тестов и пакета тестов.

Тема 7. Виды тестирования программного обеспечения

Тестирование «белого ящика». Особенности тестирования «черного ящика». Способ разбиения по эквивалентности. Способ анализа граничных значений. Методика тестирования программных систем. Тестирование элементов. Нисходящее и восходящее тестирование интеграции. Тестирование правильности. Системное тестирование.

Тема 8. Место тестирования в цикле разработки

Организация процесса тестирования в рамках жизненного цикла разработки информационной системы. Деятельность тестировщика во время фазы анализа и фазы проектирования. Подготовка и выполнение тестов. Организация альфа и бета тестирования.

Раздел 4. Технологии java для разработки информационных систем

Тема 9. Библиотека объектно-реляционного отображения Hibernate

Принципы функционирования библиотек ORM. Назначение и преимущества Hibernate. Конфигурационные файлы. Маппинг классов. Встроенные типы данных. Формирование отношений между объектами. Отношения по значению. Отношения сущностей. Виды настроек отображения для описания отношений. Виды запросов в Hibernate. Внешние запросы. Запросы HQL. Запросы как критерии отбора. Дополнительные настройки Hibernate.

Тема 10 Фреймворк Spring

Возможности и назначение Spring. Способы связывания компонентов. Автоматическое связывание и автоматическое определение компонентов. Контекст приложения. Аспектно-ориентированное программирование. Встраивание логики в существующее приложение. Понятие среза, точки подключения, аспекта, совета.

Тема 11. Библиотека Spring MVC

Создание веб-приложений с помощью Spring MVC. Концепция главного контроллера. Концепция контроллеров, сервисов и представлений. Настройка контроллеров обработки данных. Обработка форм. Валидация данных. Генерирование вывода различного формата. Динамическое создание элементов html-страниц. Шаблонизатор Thymeleaf.

Тема 12. Библиотеки фреймворка Spring

Подключение источника данных с помощью Spring Data. Репозитории. Декларативное описание запросов. Аутентификация с помощью Spring Security. Реализация регистрации, аутентификации, парольной защиты веб-приложения.

Раздел 5. Элементы жизненного цикла программного обеспечения

Тема 13. Модели и процессы жизненного цикла программного обеспечения

Классический жизненный цикл ПО. Стратегии конструирования ПО. Инкрементная модель. Спиральная модель. Быстрая разработка приложений. Тяжеловесные и облегченные процессы. XP-процесс. Практики экстремального программирования. Сложности внедрения XP-процесса. Подход TDD

Тема 14. Метрики процессов программного обеспечения

Планирование проектных задач. Параметры оценки проекта. Факторы, влияющие на стоимость программного продукта. Методы оценки себестоимости ПО. Конструктивная модель стоимости. Факторы затрат. Управление процессом разработки программного обеспечения через изменение значений параметров конструктивной модели стоимости.

Тема 15. Документирование программного обеспечения

Роль документирования, документирование требований, документация архитектуры, техническая документация, пользовательская документация, документация по ГОСТу. Ведение документации в формате Wiki.

4.3.1. Перечень семинарских, практических занятий и лабораторных работ

№ п/н	№ раздела и темы	Наименование семинаров, практических и лабораторных работ	Трудоемкость (час.)		Оценочные средства	Формируемые компетенции (индикаторы)*
			Всего часов	Из них практическая подготовка		
1	2	3	4	5	6	7
1	1	Тема 1. Диаграммы определения требований	2	2	Лабораторная работа	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3)
2	1	Тема 2. Диаграммы классов	4	4	Лабораторная работа	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3)
3	2	Тема 3 Порядок использования шаблонов проектирования	2	2	Лабораторная работа	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3), ПК-5 (ПК-5.1, ПК-5.2)
4	2	Тема 4. Примеры шаблонов проектирования	8	8	Лабораторная работа	ПК-5 (ПК-5.1, ПК-5.2)
5	2	Тема 5. Принципы проектирования	8	8	Лабораторная работа	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3), ПК-5 (ПК-5.1, ПК-5.2)
6	3	Тема 6. Основы тестирования программного обеспечения	2	2	Лабораторная работа	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3)
7	3	Тема 7. Виды тестирования программного обеспечения	4	4	Лабораторная работа	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3)
8	3	Тема 8. Место тестирования в цикле разработки	4	4	Лабораторная работа	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3), ПК-1 (ПК-1.1)
9	4	Тема 9. Библиотека объектно-реляционного отображения Hibernate	6	6	Лабораторная работа, контрольная работа	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3), ПК-5 (ПК-5.1, ПК-5.2), , ПК-1 (ПК-1.1)
10	4	Тема 10 Фреймворк Spring	6	6	Лабораторная работа	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3), ПК-5 (ПК-5.1, ПК-5.2), , ПК-1 (ПК-1.1)
11	4	Тема 11. Библиотека Spring MVC	8	8	Лабораторная работа	ПК-6 (ПК-6.1, ПК-

						6.2, ПК6.3), ПК-5 (ПК-5.1, ПК-5.2), , ПК-1 (ПК-1.1)
12	4	Тема 12. Библиотеки фреймворка Spring	6	6	Лабораторная работа, контрольная работа	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3), ПК-5 (ПК-5.1, ПК-5.2), , ПК-1 (ПК-1.1)
13	5	Тема 13. Модели и процессы жизненного цикла ПО	2	2	Лабораторная работа	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3), ПК-1 (ПК-1.1)
14	5	Тема 14. Метрики процессов программного обеспечения	2	2	Лабораторная работа	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3), ПК-1 (ПК-1.1)
15	5	Тема 15. Документирование программного обеспечения	2	2	Лабораторная работа	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3), ПК-1 (ПК-1.1)
		Всего	66	66		

4.3.2. Перечень тем (вопросов), выносимых на самостоятельное изучение студентами в рамках самостоятельной работы (СР)

Не предусмотрено.

4.4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Самостоятельная работа студентов всех форм и видов обучения является одним из обязательных видов образовательной деятельности, обеспечивающей реализацию требований Федеральных государственных стандартов высшего образования. Согласно требованиям нормативных документов самостоятельная работа студентов является обязательным компонентом образовательного процесса, так как она обеспечивает закрепление получаемых на лекционных занятиях знаний путем приобретения навыков осмысления и расширения их содержания, навыков решения актуальных проблем формирования общекультурных и профессиональных компетенций, научно-исследовательской деятельности, подготовки к семинарам, лабораторным работам, сдаче зачетов и экзаменов. Самостоятельная работа студентов представляет собой совокупность аудиторных и внеаудиторных занятий и работ. Самостоятельная работа в рамках образовательного процесса в вузе решает следующие задачи:

- закрепление и расширение знаний, умений, полученных студентами во время аудиторных и внеаудиторных занятий, превращение их в стереотипы умственной и физической деятельности;
- приобретение дополнительных знаний и навыков по дисциплинам учебного плана;
- формирование и развитие знаний и навыков, связанных с научно-исследовательской деятельностью;
- развитие ориентации и установки на качественное освоение образовательной программы;
- развитие навыков самоорганизации;
- формирование самостоятельности мышления, способности к саморазвитию, самосовершенствованию и самореализации;

– выработка навыков эффективной самостоятельной профессиональной теоретической, практической и учебно-исследовательской деятельности.

Подготовка к лекции. Качество освоения содержания конкретной дисциплины прямо зависит от того, насколько студент сам, без внешнего принуждения формирует у себя установку на получение на лекциях новых знаний, дополняющих уже имеющиеся по данной дисциплине. Время на подготовку студентов к двухчасовой лекции по нормативам составляет не менее 0,2 часа.

Подготовка к практическому занятию. Подготовка к практическому занятию включает следующие элементы самостоятельной деятельности: четкое представление цели и задач его проведения; выделение навыков умственной, аналитической, научной деятельности, которые станут результатом предстоящей работы. Выработка навыков осуществляется с помощью получения новой информации об изучаемых процессах и с помощью знания о том, в какой степени в данное время студент владеет методами исследовательской деятельности, которыми он станет пользоваться на практическом занятии. Подготовка к практическому занятию нередко требует подбора материала, данных и специальных источников, с которыми предстоит учебная работа. Студенты должны дома подготовить к занятию 3–4 примера формулировки темы исследования, представленного в монографиях, научных статьях, отчетах. Затем они самостоятельно осуществляют поиск соответствующих источников, определяют актуальность конкретного исследования процессов и явлений, выделяют основные способы доказательства авторами научных работ ценности того, чем они занимаются. В ходе самого практического занятия студенты сначала представляют найденные ими варианты формулировки актуальности исследования, обсуждают их и обосновывают свое мнение о наилучшем варианте. Время на подготовку к практическому занятию по нормативам составляет не менее 0,2 часа.

Подготовка к контрольной работе. Контрольная работа назначается после изучения определенного раздела (разделов) дисциплины и представляет собой совокупность развернутых письменных ответов студентов на вопросы, которые они заранее получают от преподавателя. Самостоятельная подготовка к контрольной работе включает в себя: — изучение конспектов лекций, раскрывающих материал, знание которого проверяется контрольной работой; повторение учебного материала, полученного при подготовке к семинарским, практическим занятиям и во время их проведения; изучение дополнительной литературы, в которой конкретизируется содержание проверяемых знаний; составление в мысленной форме ответов на поставленные в контрольной работе вопросы; формирование психологической установки на успешное выполнение всех заданий. Время на подготовку к контрольной работе по нормативам составляет 2 часа.

Подготовка к зачету. Самостоятельная подготовка к зачету должна осуществляться в течение всего семестра. Подготовка включает следующие действия: перечитать все лекции, а также материалы, которые готовились к семинарским и практическим занятиям в течение семестра, соотнести эту информацию с вопросами, которые даны к зачету, если информации недостаточно, ответы находят в предложенной преподавателем литературе. Рекомендуется делать краткие записи. Время на подготовку к зачету по нормативам составляет не менее 4 часов.

Подготовка к экзамену. Самостоятельная подготовка к экзамену схожа с подготовкой к зачету, особенно если он дифференцированный. Но объем учебного материала, который нужно восстановить в памяти к экзамену, вновь осмыслить и понять, значительно больше, поэтому требуется больше времени и умственных усилий. Важно сформировать целостное представление о содержании ответа на каждый вопрос, что предполагает знание разных научных трактовок сущности того или иного явления, процесса, умение раскрывать факторы, определяющие их противоречивость, знание имен ученых, изучавших обсуждаемую проблему. Необходимо также привести информацию о матери-

алах эмпирических исследований, что указывает на всестороннюю подготовку студента к экзамену. Время на подготовку к экзамену по нормативам составляет 36 часов для бакалавров.

В ФБГОУ ВО «ИГУ» организация самостоятельной работы студентов регламентируется Положением о самостоятельной работе студентов, принятым Ученым советом ИГУ 22 июня 2012 г.

4.5. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ РАБОТ (ПРОЕКТОВ)

1. Приложение для организации собеседований.
2. Ведение личных финансов с графическим представлением данных.
3. Разработка сайта кафедры КАиИС.
4. Учет локального запаса продуктов.
5. Приложение для кураторов с графическим представлением данных.
6. Система для онлайн тестирования.
7. Приют для животных/собак.
8. Обработка списков вещей в походы.
9. Формирование раскладки и рациона туриста.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

а) перечень литературы

1. Орлов С.А. Программная инженерия. Технологии разработки программного обеспечения : учеб. для студ. вузов / С. А. Орлов. - 5-е изд., обновл. и доп. - СПб. : Питер, 2018. - 640 с. - ISBN 978-5-496-01917-0 (10 экз.)
2. Гаврилова, И. В. Разработка приложений : учебное пособие : И. В. Гаврилова. - 3-е изд., стер. - Электрон. текстовые дан. - Москва : ФЛИНТА, 2017. - 243 с. ; есть. - Режим доступа: <https://lib.rucont.ru/efd/246530>. - ЭБС "Руконт". - неогранич. доступ. - ISBN 978-5-9765-1482-9
3. Хеффельфингер, Д. Разработка приложений Java EE 6 в NetBeans 7 : руководство / Д. Хеффельфингер ; перевод с английского Е. Н. Карышева. — М. : ДМК Пресс, 2013. — 330 с. — ISBN 978-5-94074-914-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/58693> — Режим доступа: для авториз. пользователей.
4. Николаев, Е. И. Объектно-ориентированное программирование [Электронный ресурс] : учебное пособие. Е. И. Николаев. - Электрон. текстовые дан. - Ставрополь : СКФУ, 2015. - 225 с. - Режим доступа: <https://e.lanbook.com/book/155240>, <https://e.lanbook.com/img/cover/book/155240.jpg>. - ЭБС "Лань". - Неогранич. доступ.
5. Грекул В.И. Проектирование информационных систем: курс лекций: Учеб. Пособие для студ. вузов / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. – М.: Интернет Ун-т информ. технологий, 2005. – 299 с. – ISBN 5-9556-0033-7.
6. Spring 4 для профессионалов : научное издание / К. Шефер, К. Хо, Р. Харроп. - 4-е изд. - М. : И. Д. Вильямс, 2017. - 749 с. - ISBN 978-5-8459-1992-2
7. Котляров В.П. Основы тестирования программного обеспечения: учеб. пособие / В. П. Котляров, Т. В. Коликова. – М.: Интернет-Ун-т информ. технологий: Бином. Лаб. знаний, 2006. – 286 с. – ISBN 5-94774-406-4

б) периодические издания

в) список авторских методических разработок:

лекции по курсу, видео-лекции, дампы данных, расположенные в ИОС Domic

г) базы данных, информационно-справочные и поисковые системы

1. Spring Framework Reference Documentation. URL: <http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/>
2. Hibernate Documentation. URL: <https://hibernate.org/orm/documentation/6.1/>
3. Thymeleaf tutorial. URL: <https://www.thymeleaf.org/documentation.html>

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. УЧЕБНО-ЛАБОРАТОРНОЕ ОБОРУДОВАНИЕ:

Для проведения лекционных занятий необходима аудитория с презентационным оборудованием, для проведения практических занятий необходим компьютерный класс на 25-30 рабочих мест (в зависимости от численности учебной группы), оборудованная доской, презентационной техникой.

6.2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ:

1. Комплект разработчика приложений Java Platform (JDK) 14, Standard Edition (распространяется бесплатно);
2. Интегрированная среда разработки NetBeans IDE 12 (распространяется бесплатно, LGPLv2.1, GPLv2 with Classpatch exception);
3. LibreOffice Impress (распространяется бесплатно, GNU LGPL v3+, MPL 2.0);
4. Сборка веб-сервера XAMPP URL: (<http://www.apachefriends.org/en/xampp.html>)
5. UML-редактор StarUML (распространяется бесплатно с ограничением функциональности)

6.3. ТЕХНИЧЕСКИЕ И ЭЛЕКТРОННЫЕ СРЕДСТВА:

ИОС EDUCA, DOMIC, презентационное оборудование, персональный компьютер с возможностью демонстрации презентаций в формате pdf.

7. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При реализации данного курса используются следующие образовательные технологии: технологии традиционного обучения, игровые технологии, технологии проблемного обучения, технологии обучения в сотрудничестве, технологии контекстного обучения, интерактивные технологии, технологии дистанционного обучения, активные педагогические технологии.

8. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

8.1. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ВХОДНОГО КОНТРОЛЯ

Не предусмотрено

8.2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ТЕКУЩЕГО КОНТРОЛЯ

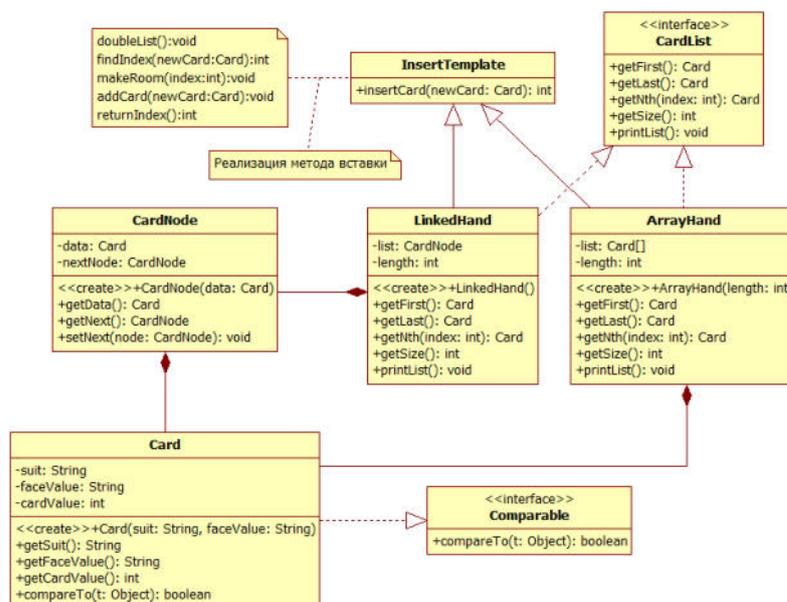
Лабораторные работы, тесты, контрольные работы представлены на странице курса в ИОС Domic в соответствии с п. 4.1

Лабораторная Паттерн Шаблонный метод

Требуется разработать программу, которая умеет правильно добавлять новую карту в колоду игровых карт. Особенностью программы является то, что колода может быть основана на двух коллекциях (связный список и массив), которые требуется реализовать самостоятельно.

Класс Card

Класс содержит следующий набор атрибутов: suit, faceValue, cardValue. Атрибут suit представляет масть карты. Масти упорядочены следующим образом: "spades", "clubs", "dimonds", "hearts". Атрибут faceValue представляет значение карты: от карты "2" до карты "ace". Атрибут cardValue предназначен для организации сортировки колоды и содержит вес карты с рамках одной масти (от 2 до 14).



При добавлении карты в колоду она должна найти свое место. Место карты определяется ее мастью и ее значением. Для реализации этого поведения карты необходимо уметь сравнивать. Сравнение должно быть реализовано в методе `compareTo` интерфейса `Comparable`. Например, следующий порядок карт будет правильным:

ace of spades, 2 of clubs, 4 of clubs, 7 of diamonds, jack of hearts, ace of hearts

Метод `toString` должен выводить значение карты в указанном выше формате

Интерфейс `CardList`

Интерфейс должен включать следующий набор методов:

`getFirst` — возвращает первый элемент в списке

`getLast` — возвращает последний элемент в списке

`getNth` — возвращает элемент с указанным номером

`getSize` — возвращает размер списка

`printList` — выводит на печать содержимое всего списка

Класс `ArrayHand`

Особенностью этого класса является то, что он оперирует со структурой фиксированной длины — массивом. Начальный размер массива задается в конструкторе. Но массив должен содержать неограниченное количество элементов. Поэтому при добавлении новой карты в уже заполненный массив сначала создается новый массив, размер которого в два раза больше предыдущего, затем элементы старого массива копируются в новый, после этого новый массив становится текущим носителем колоды и новая карта добавляется уже в него.

Обратите внимание, что при добавлении в середину массива новой карты, элементы стоящие справа сдвигаются на 1 позицию.

Класс `ListHand`

Поскольку связный список является динамической структурой с односторонним обходом, то для его реализации потребуется дополнительный класс `CardNode`, который представляет собой узел списка и содержит информацию о карте текущего узла и ссылке на следующий узел.

Алгоритм добавления карты в колоду

Согласно паттерну Шаблонный метод, вам предстоит реализовать единый алгоритм добавления новой карты в колоду:

Если список полон, увеличить его размер

Найти позицию для вставки карты

Создать ячейку для новой карты

Добавить новую карту в список

Вернуть номер карты в списке

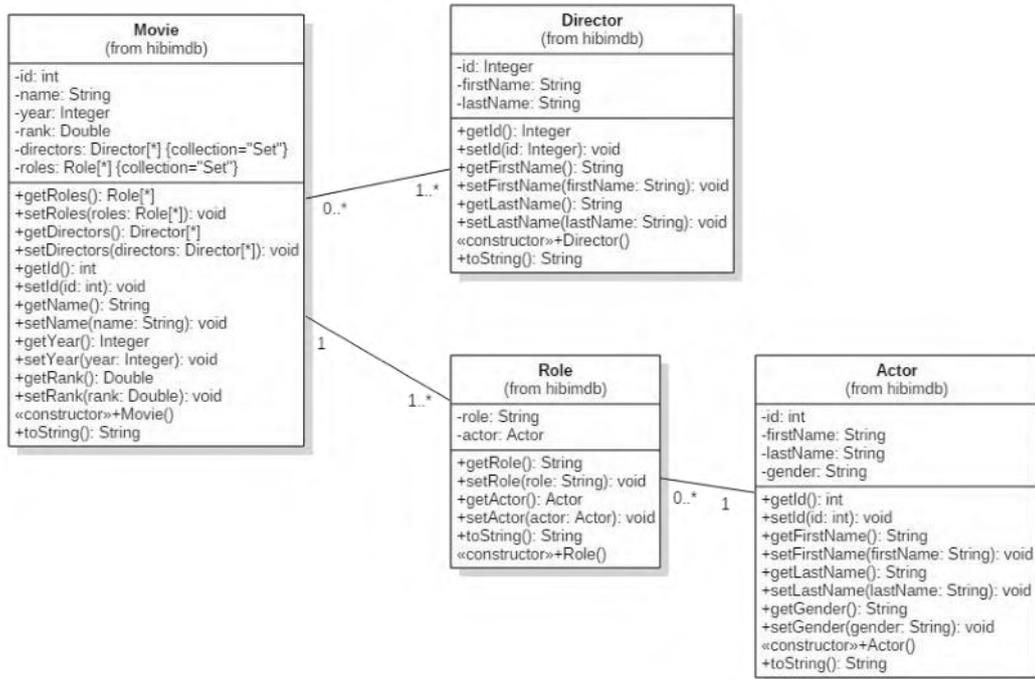
Каждый из представленных выше шагов должен быть реализован в виде метода. Обратите внимание, что первый шаг не имеет смысла для связного списка, поэтому данный метод следует реализовывать в виде перехватчика. Метод по вставке должен быть реализован в классе `InsertTemplate`, который является родительским по отношению к классам со списками.

Задание

Для реализации программы требуется принять решение, какие из шагов алгоритма вставки будут представлять собой абстрактные методы (реализуются в каждом классе-наследнике отдельно), а какие должны быть реализованы в классе-родителе как общие на два наследника.

Лабораторная ORM Hibernate. База данных фильмов и актеров

Требуется реализовать преобразование Hibernate для загрузки имеющихся данных по фильмам и актерам в программу. Перед началом работы загрузите на локальный сервер MySQL данные из предлагаемого дампа imdb_small.sql.



Создайте структуру классов, представленных на диаграмме. Напишите mapping-файлы для загрузки объектов из базы данных в систему.

Особенностью задания является то, что для роли нет идентификатора. Это означает, класс фильма и класс роли должны быть связаны отношением по значению. Описание связи между ролью и актером происходит в mapping-файле для фильма.

В основной программе: Выведите всех актеров с именем Julia, Выведите всех актеров, принявших участие в фильме Fight Club, Выведите всех актеров фильма Pi, а также их роли, Выведите информацию о фильмах, которые вышли с 1995 по 2000 год, Выведите всех актеров, которые снимались в 1999 году.

Для получения соответствующей информации следует воспользоваться конструкцией Criteria.

Лабораторная Книжная полка

Постановка задачи

Создайте программу для хранения и поиска книг. Книги описываются следующим набором параметров: заголовок; автор (или авторы); год издания; количество страниц; ISBN; ключевые слова; фото обложки*.

Книги, авторы и ключевые слова должны представлять собой отдельные объекты.

Дополнительно требуется реализовать функциональность по поиску книг по трем параметрам: по заголовку; по автору; по ключевым словам.

Основная форма приложения может состоять из трех вкладок: Общий список книг, Добавление новых книг (авторов, ключевых слов), Поиск книг по параметрам.

Содержимое списков должно обновляться только при переходе на соответствующую вкладку (страницу). Поиск книги должен быть реализован с возможностью обработки нескольких критериев одновременно (см. флажок «Использовать»)

Дополнительная функциональность*

Добавьте новую форму для вывода детальной информации о книге. Форма открывается при выборе книги в общем списке или в списке найденных книг. На форме разместите информацию о книге и изображение обложки. Для этого нужно добавить возможность хранения в базе данных двоичных данных и возможность отображения растровых рисунков на форме.

Лабораторная работа Spring. Работа со списками данных

Требуется написать программу для получения данных по странам. На начальной странице пользователь выбирает один из континентов. Ему показывается список стран (в виде таблицы), расположенных на указанном континенте. При выборе конкретной страны выводится страница с детальной информацией.

Реализация программы представляется в несколько этапов:

Создание maven-проекта и настройка pom-файла, файла web.xml и диспетчера

Создание и обработка списка выбора с континентами

Формирование списка стран и вывод в виде таблицы (каждая из строк представляет ссылку на соответствующую страну)

Вывод информации о стране: код, название, континент, население, площадь

Заголовок класса Country

```
public class Country {
    private String code;
    private String name;
    private String continent;
    private Double surfaceArea;
    private Integer population;

    //constructors and methods here
}
```

В класса должен присутствовать полный конструктор

Получение информации

Получение списка континентов, стран и информации о странах осуществляется с помощью класса CountryService. В нем содержатся данные по странам и набор методов для запросов. Данный класс должен быть объявлен как сервис и соответствующий bean объявлен в файле applicationContext

Формирование списка стран

Для отображения списка стран можно воспользоваться циклом на jsp-странице

```
<c:forEach items="${countries}" var="country">
  <tr>
    <td>${country.name}</td>
    <td>${country.continent}</td>
  </tr>
</c:forEach>
```

Обработка параметров

Переход к выбранной осуществляется через передачу ее кода либо в качестве параметра get-запроса, либо в url-адресе. Например

```
http://localhost:8080/world/country?code=PRY
```

или

```
http://localhost:8080/world/country/PRY
```

В первом случае для получения данных потребуется следующий заголовок метода в контроллере:

```
@RequestMapping(value="/country", params="code")
public String getCountry(Model model, @RequestParam("code") String code){
    //code here
}
```

Во втором:

```
@RequestMapping(value="/country/{code}")
public String getCountry(Model model, @PathVariable("code") String code){
    //code here
}
```

Проверка ошибок

Добавьте класс для перехвата всех исключений, возникающих в проекте. Должны перехватываться следующие ошибки: получение страны по несуществующему коду, обращение по несуществующему адресу.

Вариант описания задания на курсовой проект. Сервис проката автомобилей

Требуется разработать информационную систему для отслеживания финансовых показателей работы пункта проката автомобилей. В автопарк компании входит некоторое количество автомобилей различных марок, стоимостей и типов. Каждый автомобиль имеет свою цену для организации проката. В пункт проката обращаются клиенты. Все клиенты проходят обязательную регистрацию, при которой о них собирается стандартная информация (ФИО, адрес, телефон). Каждый клиент может обращаться в пункт проката несколько раз. Все обращения клиентов фиксируются, при этом по каждой сделке запоминаются дата выдачи и ожидаемая дата возврата. Перед получением автомобиля клиент оставляет некоторую залоговую сумму, которая ему полностью возвращается после успешной сдачи автомобиля. Стоимость проката автомобиля зависит не только от самого автомобиля, но и от срока его проката, а также от года выпуска. Также предусмотрена система штрафов за возвращение автомобиля в ненадлежащем виде.

Для пользователя нужно разработать следующие отчеты: информация по имеющимся автомобилям в разрезе их занятости и в разрезе их финансовых показателей

8.3. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПРОМЕЖУТОЧНОГО КОНТРОЛЯ

Материалы для проведения текущего и промежуточного контроля знаний студентов:

№	Вид контроля	Контролируемые темы (разделы)	Контролируемые компетенции/ индикаторы
1	2	3	4
1	Контроль в виде тестов Разноуровневые задания к лабораторному практикуму	Раздел 1. Унифицированный Язык моделирования информационных систем	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3)

2	Контроль в виде тестов Разноуровневые задания к лабораторному практикуму	Раздел 2. Шаблоны проектирования	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3), ПК-5 (ПК-5.1, ПК-5.2)
3	Контроль в виде тестов Разноуровневые задания к лабораторному практикуму	Раздел 3. Тестирование программного обеспечения	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3), ПК-5 (ПК-5.1, ПК-5.2), ПК-1 (ПК-1.1)
	Зачет	Промежуточная аттестация	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3), ПК-5 (ПК-5.1, ПК-5.2), ПК-1 (ПК-1.1)
4	Контроль в виде тестов Контрольные работы Разноуровневые задания к лабораторному практикуму	Раздел 4. Технологии java для разработки информационных систем	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3), ПК-5 (ПК-5.1, ПК-5.2), ПК-1 (ПК-1.1)
5	Разноуровневые задания к лабораторному практикуму	Раздел 5. Элементы жизненного цикла программного обеспечения	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3), ПК-1 (ПК-1.1)
	Экзамен, курсовой проект	Промежуточная аттестация	ПК-6 (ПК-6.1, ПК-6.2, ПК6.3), ПК-5 (ПК-5.1, ПК-5.2), ПК-1 (ПК-1.1)

Оценочные средства текущего контроля формируются в соответствии с Положением о балльно-рейтинговой системе.

Процедура сдачи зачета

Выполнение тестов, лабораторных и домашних заданий в информационно-образовательной среде дает 50 баллов.

При проведении зачета студенту может быть задан вопрос по одной из представленных ниже тем для собеседования, оценивающийся в 20 баллов.

Темы для собеседования на зачете

1. Основы UML. Диаграммы классов.
2. Основы UML. Диаграммы вариантов использования.
3. Основы UML. Диаграммы деятельности.
4. Назначение паттернов проектирования. Их структура. Примеры реализации паттернов.
5. Паттерн Abstract Factory. Структура, описание и область применения.
6. Паттерн Singleton. Структура, описание и область применения.
7. Паттерн Factory Method. Структура, описание и область применения.
8. Паттерн Adapter. Структура, описание и область применения.
9. Паттерн Composite. Структура, описание и область применения.
10. Паттерн Decorator. Структура, описание и область применения.
11. Паттерн Facade. Структура, описание и область применения.
12. Паттерн Chain of Responsibility. Структура, описание и область применения.
13. Паттерн Command. Структура, описание и область применения.
14. Паттерн Iterator. Структура, описание и область применения.
15. Паттерн Observer. Структура, описание и область применения.
16. Понятие тестирования и отладки. Трудности тестирования ПО. Статическое и динамическое тестирование. Фазы тестирования. Понятие набора тестов и пакета тестов.

17. Организация процесса тестирования в рамках жизненного цикла разработки информационной системы. Деятельность тестировщика во время фазы анализа и фазы проектирования. Подготовка и выполнение тестов. Организация альфа и бета тестирования.
18. Виды тестирования. Классификация тестирования.
19. Способы тестирования условий. Тестирование ветвей и операторов отношений. Организация и проведение модульного тестирования.
20. Особенности тестирования «черного ящика». Способ разбиения по эквивалентности. Способ анализа граничных значений. Методика тестирования программных систем.
21. Нисходящее и восходящее тестирование интеграции. Тестирование правильности. Системное тестирование.
22. Автоматизация модульного тестирования. JUnit.

Процедура сдачи экзамена

Выполнение лабораторных, домашних заданий в информационно-образовательной среде, а также контрольных работ и тестов дает 60 баллов в семестре. Распределение весов учебных единиц представлено в ИОС Domic. В оценке экзамен составляет 30 баллов. Дополнительные 10 баллов выставляется за посещаемость.

Для успешной сдачи экзамена требуется осветить два теоретических вопроса, каждый из которых дает 15 баллов. После процедуры выбора, экзаменуемому предоставляется 60 минут для подготовки ответа на выбранные вопросы. После ответа на основные вопросы экзаменуемому может быть задан дополнительный вопрос, оценивающийся в 10 баллов.

Критерии оценивания теоретических вопросов:

15 баллов: экзаменуемый самостоятельно и без наводящих вопросов осветил все основные разделы вопроса, ввел необходимые определения, описал необходимые алгоритмы; привел примеры.

10 баллов: экзаменуемый, после задания уточняющих и наводящих вопросов, в достаточной степени осветил содержание вопроса, ввел определения, возможно не осветил некоторые шаги алгоритмов, привел примеры.

5 баллов: экзаменуемый не осветил вопрос, но ответил на наводящие, уточняющие и дополнительные вопросы, привел основные определения, но не привел необходимые примеры.

0 баллов: не ответил на выбранные им вопросы экзамена и уточняющие вопросы экзаменатора.

Вопросы к экзамену

1. Назначение паттернов проектирования. Их структура. Примеры реализации паттернов.
2. Паттерн Abstract Factory. Структура, описание и область применения.
3. Паттерн Composite. Структура, описание и область применения.
4. Паттерн Decorator. Структура, описание и область применения.
5. Паттерн Chain of Responsibility. Структура, описание и область применения.
6. Паттерн Command. Структура, описание и область применения.
7. Паттерн Iterator. Структура, описание и область применения.

8. Паттерн Observer. Структура, описание и область применения.
9. Понятие тестирования и отладки. Трудности тестирования ПО. Статическое и динамическое тестирование. Фазы тестирования. Понятие набора тестов и пакета тестов.
10. Виды тестирования. Различные классификации тестирования.
11. Особенности тестирования «черного ящика». Способ разбиения по эквивалентности. Способ анализа граничных значений. Методика тестирования программных систем.
12. Автоматизация модульного тестирования. JUnit.
13. Принципы функционирования библиотек ORM. Назначение и преимущества Hibernate. Конфигурационные файлы. Маппинг классов.
14. Формирование отношений между объектами. Отношения по значению. Отношения сущностей. Виды настроек отображения для описания отношений.
15. Виды запросов в Hibernate. Внешние запросы. Запросы JPQL. Запросы как критерии отбора. Дополнительные настройки Hibernate.
16. Возможности и назначение Spring. Способы связывания компонентов. Автоматическое связывание и автоматическое определение компонентов.
17. Создание веб-приложений с помощью Spring MVC. Настройка контроллеров обработки данных. Обработка форм.
18. Создание веб-приложений с помощью Spring MVC. Настройка контроллеров обработки данных. Валидация данных. Работа с сервисами.
19. Обеспечение доступа к данным веб-приложений. Spring Data.
20. Реализация аутентификации веб-приложений. Spring Security.
21. Классический жизненный цикл ПО. Стратегии конструирования ПО. Инкрементная модель.
22. Быстрая разработка приложений. Тяжеловесные и облегченные процессы. XP-процесс. Практики экстремального программирования. Сложности внедрения XP-процесса.
23. Факторы, влияющие на стоимость программного продукта. Конструктивная модель стоимости. Управление процессом разработки программного обеспечения через изменение значений параметров конструктивной модели стоимости.
24. Роль документирования, документирование требований, документация архитектуры, техническая документация, пользовательская документация.

Оценивание курсовой работы

Курсовая работа включает реализацию программного проекта, написание текста курсовой и защиту (выступление) проекта.

Программный проект оценивается по следующим критериям: сложность проекта, использование в проекте БД, реализация сценариев, требующих авторизованных пользователей, использование в реализации служебных механизмов программных библиотек, используемые проектные решения, аккуратность (общее впечатление) проекта.

Текст курсовой работы оценивается согласно методическим рекомендациям по структуре и содержанию работы, представленным на странице курса в ИОС Domic. Правила оформления курсовой и шаблон в издательской системе Latex также представлены в методических рекомендациях.

Выступление (защита курсового проекта) проводится публично в присутствии студенческой группы и заинтересованных преподавателей кафедры. Для выступления студент

готовит слайды презентации и видео-демонстрацию работы основных сценариев работы приложения.

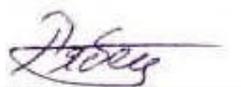
«Отлично» — выставляется в случае, если реализованное приложение полностью соответствует указанным выше критериям. Работа имеет четкую внутреннюю логическую структуру, соответствующую методическим рекомендациям. Процесс защиты продемонстрировал полную разработанность избранной предметной области и компетентность студента.

«Хорошо» — если реализованное приложение вполне соответствует большинству указанных выше критериев. Работа имеет четкую внутреннюю логическую структуру, в целом соответствующую методическим рекомендациям. Процесс защиты продемонстрировал необходимую разработанность предметной области. Вместе с тем, реализованное приложение содержит ряд недостатков, недоработок реализованных сценариев, не имеющих принципиального характера.

«Удовлетворительно» — если реализованное приложение соответствует отдельным (некоторым) из указанных выше критериев. Работа не полностью соответствует всем формальным требованиям, предъявляемым в методических рекомендациях. Проработаны только базовые сценарии предметной области.

«Неудовлетворительно» — выставляется в случае, если в процессе защиты выявились факты плагиата основных результатов работы, несоответствие заявленных результатов имеющейся реализации, несоответствие реализованного приложения критериям, тривиальность приложения. Также оценка выставляется в случае непредставления текста курсовой работы.

Разработчик:

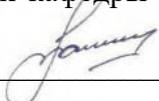


Рябец Леонид Владимирович, к.ф.-м.н., доцент, доцент кафедры алгебраических и информационных систем

Программа составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 02.03.02 «Фундаментальная информатика и информационные технологии» (уровень бакалавриата), утвержденный приказом Министерства образования и науки Российской Федерации от 23 августа 2017 г. N 808, зарегистрированный в Минюсте России «14» сентября 2017 г. № 48185 с изменениями и дополнениями с изменениями и дополнениями от: 26 ноября 2020 г., 8 февраля 2021 г.

Программа рассмотрена на заседании кафедры Алгебраических и информационных систем ИМИТ ИГУ «04» апреля 2023 г.

Протокол № 9 Зав. кафедрой _____ Пантелеев В.И.



Настоящая программа, не может быть воспроизведена ни в какой форме без предварительного письменного разрешения кафедры-разработчика программы.