

МИНОБРНАУКИ РОССИИ
Ярославский государственный университет им. П.Г. Демидова

Базовая кафедра разработки цифровых платформ
для государственного управления НПО Криста

УТВЕРЖДАЮ

Декан факультета ИВТ
Д.Ю. Чалый


(подпись)

«__18__» _____ мая _____ 2021 г.

Рабочая программа дисциплины
«Моделирование и проектирование государственных информационных систем»

Направление подготовки
02.03.02 Фундаментальная информатика и информационные технологии

Профиль
«Информатика и компьютерные науки»

Квалификация выпускника
Бакалавр

Форма обучения
очная

Программа одобрена
на заседании кафедры

от «15» апреля 2021 года, протокол № 8

Программа одобрена НМК
факультета ИВТ

протокол № 7 от « 17 » мая 2021года

Ярославль
2021

1. Цели освоения дисциплины

Целями освоения дисциплины «Моделирование и проектирование государственных информационных систем» являются изучение основ алгоритмизации и программирования, математических методов в компьютерных технологиях, информационные системы и технологии.

2. Место дисциплины в структуре образовательной программы

Дисциплина «Моделирование и проектирование государственных информационных систем» относится к дисциплинам по выбору образовательной программы и входит в модуль «Часть, формируемая участниками образовательных отношений».

Для освоения данной дисциплиной студенты должны владеть математическими методами в компьютерных технологиях, информационными системами и технологиями, проектированием информационных систем, разработкой программных приложений, программной инженерией, основами алгоритмизации и знанием структур данных.

Полученные в курсе «Моделирование и проектирование государственных информационных систем» знания необходимы для изучения последующих дисциплин модуля «Часть, формируемая участниками образовательных отношений», а также для продолжения обучения в магистратурах по направлениям Прикладная информатика, Фундаментальной информатики и информационных технологий, а также Прикладной математики и информатики.

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ООП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
Общепрофессиональные компетенции		
ПК-3 Способен к разработке и проектированию программного обеспечения, к использованию современных технологий программирования	ПК-3.3 Владеет навыками разработки алгоритмических и программных решений в области системного и прикладного программного обеспечения	Знать: - основные теоремы и уравнения квантовой механики; - приближенные методы вычислений в квантовой механике. Уметь: - воспроизводить ключевые физические принципы и математические приемы, используемые при построении квантовой механики; - определять корректность использования тех или иных физических предположений и математических методов, применяемых при формулировке и решении квантово-механических задач.

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 2 зачетных единиц, 72 акад. часов.

№ п/п	Темы (разделы) дисциплины, их содержание	Се ме стр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)					самос тояте льная работ а	Формы текущего контроля успеваемости Форма промежуточной аттестации (по семестрам) <i>Формы ЭО и ДОТ (при наличии)</i>
			Контактная работа						
			лек ции	пр акт ич еск ие	ла бо рат ор ны е	ко нс ул ьта ци и	атт ест аци онн ые исп ыта ния		
1	Основы Java.	6	2		2			1	Задания для самостоятельной работы. Практическая задача
2	JUnit junit-jupiter-api	6	2		2			2	Задания для самостоятельной работы. Практическая задача
	в том числе с ЭО и ДОТ							2	
3	Версионное хранилище	6	2		2			2	Задания для самостоятельной работы. Практическая задача
4	Code review	6	2		2			5	Задания для самостоятельной работы. Практическая задача
5	Maven и SonarQube	6	2		2			5	Задания для самостоятельной работы. Практическая задача
6	Web практика	6	2		2			5	Задания для самостоятельной работы. Практическая задача

7	Jenkins. Сервер автоматизации	6	2		3			5	Задания для самостоятельной работы. Практическая задача
	<i>в том числе с ЭО и ДОТ</i>							2	
8	Журналирование, отладка, профилирование	6	4		3			4	Задания для самостоятельной работы. Практическая задача
						5	0,3	2,7	Зачет При подготовке к зачету: Контрольная работа по результатам освоения дисциплины
	<i>в том числе с ЭО и ДОТ</i>							1	Контрольная работа по результатам освоения дисциплины ЭУК в LMS Moodle
	Итого за 6 семестр 108 часа		18		18	4	0,3	31,7	
	<i>в том числе с ЭО и ДОТ</i>							5	

Содержание разделов дисциплины:

1. Основы Java.

Система управления задачами

Соглашение (или описание процесса)

Система учета задач

Redmine. Задачи. Жизненный цикл задачи.

Основы Java

Работа в Redmain

IDEA

Linux

Практическое задание

2. JUnit junit-jupiter-api

ООП

Проектирование калькулятора

Тестирование

JUnit, модульное тестирование

Практическое задание

3. Версионное хранилище

Понятие версионного хранилища

История версионных хранилищ

Распределенные версионные хранилища

Работа с Mercurial

Java. Лямбда функции

Практическое задание

4. Code review

Исходный код

Критерии хорошего кода

Инструменты для review

Задачи, стоящие перед рецензентом

Code review в Kallithea

Практическое задание

5. Maven и SonarQube

Понятие Maven. “Фреймворк” и “сборка”

Возможности Maven

Пример Maven GAV (Group, Artifact, Version)

Инструмент статического анализа кода SonarQube

Понятие SonarQube

Практическое задание

6. Web практика.

Протокол HTTP

Архитектура web-приложения

Разработка web-приложения

Практическое задание

7. Jenkins. Сервер автоматизации

Серверы автоматизации

Общие принципы

Знакомство с Jenkins

Создание своей задачи сборки

Дополнительные требования к проекту

Автоматический анализ в SonarQube

8. Журналирование, отладка, профилирование.

Понятие отладки. Подходы.

Журналирование, отладка, профилирование.

Доработка web-приложения.

5. Образовательные технологии, в том числе технологии электронного обучения и дистанционные образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Академическая лекция с элементами лекции-беседы – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Элементы лекции-беседы обеспечивают контакт преподавателя с аудиторией, что позволяет привлекать внимание студентов к наиболее важным темам дисциплины, активно вовлекать их в учебный процесс, контролировать темп изложения учебного материала в зависимости от уровня его восприятия.

Практическое занятие – занятие, посвященное освоению конкретных умений и навыков по закреплению полученных на лекции знаний.

Консультации – вид учебных занятий, являющийся одной из форм контроля самостоятельной работы студентов. На консультациях по просьбе студентов рассматриваются наиболее сложные моменты при освоении материала дисциплины, преподаватель отвечает на вопросы студентов, которые возникают у них в процессе самостоятельной работы.

В процессе обучения используются следующие технологии электронного обучения и дистанционные образовательные технологии:

Электронный учебный курс «Моделирование и проектирование государственных информационных систем» в LMS Электронный университет Moodle ЯрГУ, в котором:

- представлены задания для самостоятельной работы обучающихся по темам дисциплины;
- осуществляется проведение отдельных мероприятий текущего контроля успеваемости студентов;
- представлены тексты лекций по отдельным темам дисциплины;
- представлены правила прохождения промежуточной аттестации по дисциплине;
- представлен список учебной литературы, рекомендуемой для освоения дисциплины;
- представлена информация о форме и времени проведения консультаций по дисциплине в режиме онлайн;
- посредством форума осуществляется синхронное и (или) асинхронное взаимодействие между обучающимися и преподавателем в рамках изучения дисциплины.

6. Перечень лицензионного и (или) свободно распространяемого программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине

В процессе осуществления образовательного процесса по дисциплине используются: для формирования материалов для текущего контроля успеваемости и проведения промежуточной аттестации, для формирования методических материалов по дисциплине:

- программы Microsoft Office;
- издательская система LaTeX;
- Adobe Acrobat Reader.

7. Перечень современных профессиональных баз данных и информационных справочных систем, используемых при осуществлении образовательного процесса по дисциплине (при необходимости)

В процессе осуществления образовательного процесса по дисциплине используются:

Автоматизированная библиотечно-информационная система «БУКИ-NEXT»
http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php

8. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет» (при необходимости), рекомендуемых для освоения дисциплины

а) основная литература

1. Моделирование процессов и систем [Электронный ресурс] : учебник и практикум для вузов / Е. В. Стельмашонок, В. Л. Стельмашонок, Л. А. Еникеева, С. А. Соколовская ; под ред. Е. В. Стельмашонок, М., Юрайт, 2021, 289с
2. Гвоздева, Т. В., Проектирование информационных систем. Стандартизация [Электронный ресурс] : учебное пособие для вузов / Т. В. Гвоздева, Б. А. Баллод. — 2-е изд., стереотип., СПб., Лань, 2021, 252с
2. <https://www.redmine.org/guide>
3. <https://git-scm.com/doc>

б) дополнительная литература

1. <https://junit.org/junit5/docs/current/user-guide/>
2. <https://www.jetbrains.com/help/idea/discover-intellij-idea.html>
3. <https://docs.gitlab.com/>
4. <https://maven.apache.org/guides/getting-started/>
5. <https://docs.sonarqube.org/latest/>
6. <https://undertow.io/documentation.html>
7. <https://developer.mozilla.org/ru/docs/Web/HTML>

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

- учебные аудитории для проведения занятий лекционного типа;
- учебные аудитории для проведения практических занятий (семинаров);
- учебные аудитории для проведения групповых и индивидуальных консультаций;
- учебные аудитории для проведения текущего контроля и промежуточной аттестации;
- помещения для самостоятельной работы;
- помещения для хранения и профилактического обслуживания технических средств обучения.

Специальные помещения укомплектованы средствами обучения, служащими для представления учебной информации большой аудитории. Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к электронной информационно-образовательной среде ЯрГУ.

Автор:

Заведующая кафедрой, к.э.н., О.В. Коновал

**Приложение № 1 к рабочей программе дисциплины
«Моделирование и проектирование
государственных информационных систем»**

**Фонд оценочных средств
для проведения текущего контроля успеваемости
и промежуточной аттестации студентов
по дисциплине**

**1. Типовые контрольные задания и иные материалы,
используемые в процессе текущего контроля успеваемости**

Задания для самостоятельной работы

Задания по теме № 1 «Основы Java»:

Практическая задача

1. Зайти в студенческий Redmine, проект “2021 Введение в промышленную разработку ПО”, Wiki.
2. Создать самим себе задачу на основе текста задачи 1.
3. Перевести задачу в “В работу”, выполнить: создать проект через IDEA, ...

IDEA: создание проекта

File > New > Project... > Maven.

Заполнить groupId.

Например: ru.ac.uniyar.ivanov

ru.practice.summer.ivanov

Заполнить artifactId: calculator, Next.

Выбрать расположение проекта. Например: /home/user/projects/calculator

IDEA: создание класса консольного приложения

1. src/main/java, ПКМ, New Java class, ConsoleApp, ok.
public class ConsoleApp {
public static void main(String[] argv) {
System.out.println("Привет!")
}

Практическая задача

Написать калькулятор для целых чисел.

Ввод/вывод - через консоль.

На вход: первое число, операция(+, -, *, /), второе число.

между числами и знаком пробел

Пример ввода: 1 + 3

На выходе: результат выполнения операции.

Пример: 4

Добавить поддержку обыкновенных дробей

Пример ввода: $1/2 * 2/3$

Пример вывода: $2/6$

Добавить поддержку аргументов командной строки

Если числа и операция переданы через аргументы командной строки, то не запрашивать ничего у пользователя и сразу выдавать результат.

Задания по теме № 2 «JUnit junit-jupiter-api»:

План работ на лабораторную

-
-
- работу”
-
- бумаге
- проектирования
-
-
- преподавателю
-
- “Дерева”
-
- завершении

- Создать задачу
- Перевести свою задачу “В
- Спроектировать “Дерево” на
- Отписаться в задаче об окончании
- Написать тесты для всех функций
- Показать набор тестов
- Приступить к реализации
- Отписаться в задаче, о

Описание задачи

-
- хранящую в себе дерево
-
- идентификатор
-
-
-
-
- имени и идентификатору
-
-
-
-
- библиотека

- Реализовать структуру данных
- Использовать ООП
- Каждый узел хранит строку и
- Реализовать операции
- создание дерева
- добавление узла в дерево
- удаление дочернего узла по его
- удаление всех дочерних узлов
- поиск дочернего узла по имени
- изменение узла
- Main функция не нужна. Это будет

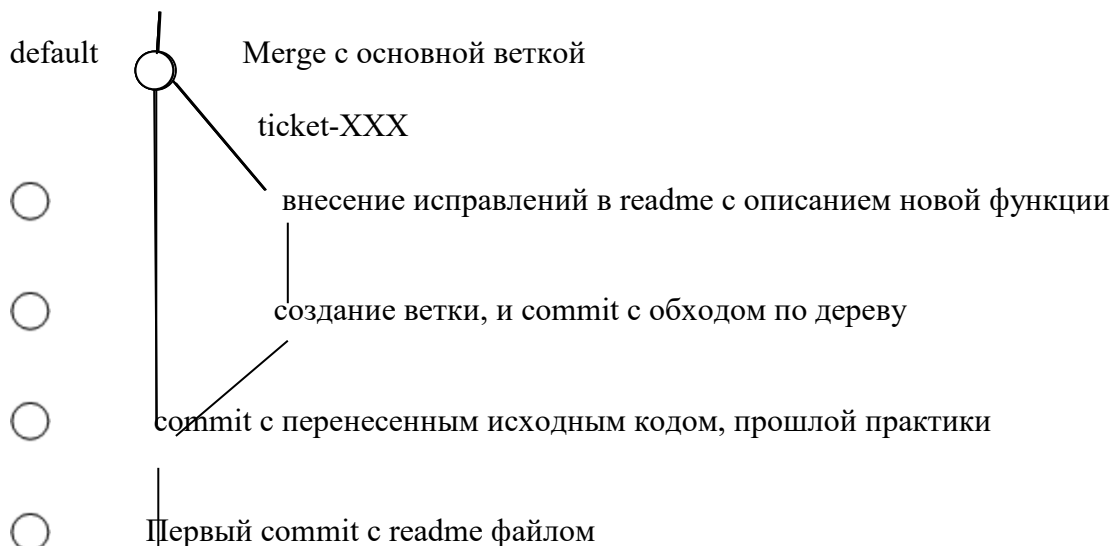
Задания по теме № 3 «Версионное хранилище»:

Практическая задача

-
-
- Kalithea (<имя_tree>)
- занятия (/src, pom.xml)
- консоль
-
-
- номер задачи)
-
- default веткой

- Создать задачу
- Создать новый репозиторий
- Перенести проект с прошлого
- Метод обхода дерева и печати в
- Индикация уровня узла пробелами
- Создать ветку ticket-XXX (XXX-
- Коммит в ветку
- В конце занятия сделать merge с

Что должно получиться



Задания по теме № 4 «Code review»:

1. ух проектов. Провести code review не менее 2-
2. ух рецензентов по своим проектам. Получить одобрение не менее 2-

Задания по теме № 5 «Maven и SonarQube»:

- Выполнить команду
mvn sonar:sonar -Dsonar.host.url=студенческий сервер SonarQube
- Зайти на студенческий сервер SonarQube
- Посмотреть на список ошибок
-

Задания по теме № 6 «Web практика»:

Практическая задача

Разработать web-приложение с функциями:

- просмотра дерева,
- добавления элемента,
- удаления элемента,
- модификации элемента.

Подробнее - см. задачу в студенческий Redmine

Подготовить проект

1. Открыть свой проект с деревом в IDEA.
 2. Добавить новые зависимости в pom.xml.
 3. Добавить классы-примеры.
 4. Запустить приложение, убедиться что по ссылке выводится тестовая страница.
- Код для добавления - см. Wiki в студенческий Redmine

Задания по теме № 7 «Jenkins Сервер автоматизации»:

Требования к проекту: pom.xml

pom.xml должен быть расположен в корне репозитория

Требования к проекту: зависимости

```
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-api</artifactId>
  <version>5.3.1</version>
</dependency>
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-engine</artifactId>
  <version>5.3.1</version>
  <scope>test</scope>
</dependency>

<dependency>
  <groupId>org.junit.vintage</groupId>
  <artifactId>junit-vintage-engine</artifactId>
  <version>5.3.1</version>
  <scope>test</scope>
</dependency>
```

Требования к проекту: плагины

```
<plugin>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>2.22.0</version>
</plugin>
```

ние

Практическая задача

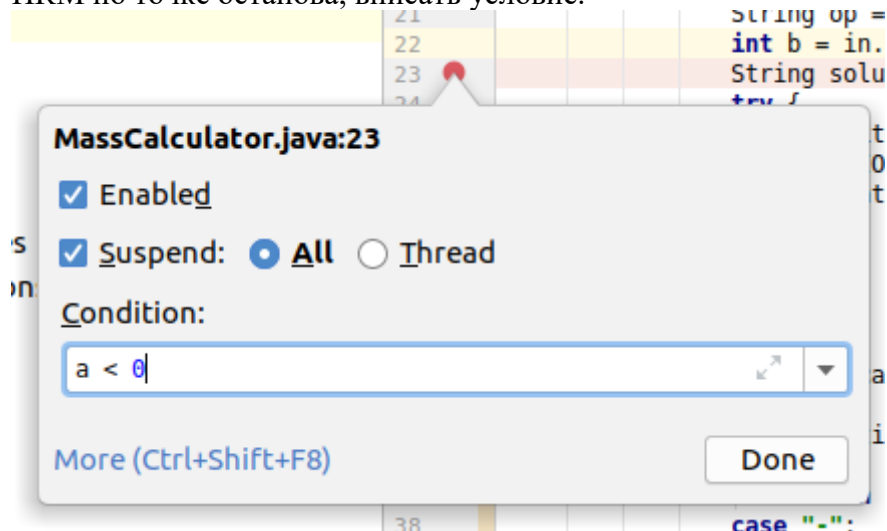
1. Скачать проект - файловое хранилище
2. Открыть в IDEA.
3. Запустить, разобраться как работает.

Практическая задача

1. Обновиться на ветку big_data
hg up big_data -C
2. Локализовать ошибку во входных данных с помощью логирования.
logger.info

Практическая задача

ПКМ по точке останова, вписать условие.



Локализовать вторую ошибку.

С помощью:

- изучения исходного кода,
- условных точек останова,
- логирования.

Практическая задача

1. Запустить jvisualvm.
2. Изучить процесс IDEA или процесс калькулятора.

Список вопросов к зачету:

1. Способность работать с системой учета задач Redmine. Умеет создавать задачи, умеет информировать о завершении задачи или возникновении проблем с решением через систему учета задач.
2. Способность разрабатывать консольные приложения уровня продвинутый калькулятор на Java в IDE IntelliJ IDEA. Может разработать консольные приложения уровня продвинутый калькулятор на Java в IDE IntelliJ IDEA.

3. Способность работать с системой контроля версий Git. Может добавлять файлы и изменения в локальный и удаленный репозиторий в основную рабочую ветку и в дополнительные ветки. Умеет забирать чужие изменения и обновлять свою рабочую копию.
4. Способность разрабатывать юнит-тесты с использованием библиотеки JUnit5 и подхода TDD. Умеет разрабатывать приложения написав сначала интерфейсы и тесты на большую часть требуемого функционала, и только затем реализацию. Умеет разрабатывать тесты на JUnit5.
5. Способность вносить код в хранилище через Pull request'ы и проводить рецензирование чужих Pull request'ов. Умеет вносить код в хранилище через Pull request'ы и проводить рецензирование чужих Pull request'ов на примере GitLab.
6. Способность работать с Java проектом с помощью Maven: создавать, добавлять зависимости, собирать. Умеет работать с Java проектом с помощью Maven: создавать, добавлять зависимости, собирать.
7. Способность проверять качество кода в SonarQube. Умеет проверить качество кода проекта и исправить замечания анализаторов при предоставленном сервере SonarQube.
8. Способность разработать Web-приложение с CRUD операциями для списка или дерева в т.ч. с минимальным пользовательским интерфейсом на HTML. Умеет разрабатывать Web-приложение на Java выдающее динамические HTML страницы. Умеет обрабатывать POST-запросы для форм и реализовывать на таких запросах и чистом HTML действия по редактированию элементов дерева или списка. Умеет проектировать и реализовывать адресацию и маршрутизацию для HTTP ресурсов - элементов дерева или списка.
9. Способность работать в DevTools браузера во вкладке Network в объеме необходимом для отладки HTTP GET и HTTP POST запросов, в т.ч. отладке отправки формы. Умеет переходить от запроса в DevTools браузера к методу в коде сервера, в т.ч. в случае возникновения ошибок при разработке.
10. Способность с помощью логирования и продвинутой отладки находить ошибки в большом ПО или ПО работающем с гигантскими объемами данных. Умеет с помощью логирования и продвинутой отладки находить ошибки в большом ПО или ПО работающем с гигантскими объемами данных.

Правила выставления оценки на зачете.

По итогам зачета выставляется одна из оценок: «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

Оценка «Отлично» выставляется студенту, который демонстрирует глубокое и полное владение содержанием материала; осуществляет межпредметные связи; умеет связывать теорию с практикой. Студент дает развернутые, полные и четкие ответы на вопросы, соблюдает логическую последовательность при изложении материала.

Оценка «Хорошо» выставляется студенту, ответ которого на зачете в целом соответствуют указанным выше критериям, но отличается меньшей обстоятельностью, глубиной, обоснованностью и полнотой. В ответе имеют место отдельные неточности (несущественные ошибки).

Оценка «Удовлетворительно» выставляется студенту, который дает недостаточно полные и последовательные ответы на вопросы, но при этом демонстрирует умение выделить существенные и несущественные признаки и установить причинно-следственные связи. Ответы излагаются с использованием терминологии, но при этом допускаются ошибки в определении и раскрытии некоторых основных понятий, формулировке положений, которые студент затрудняется исправить самостоятельно. При аргументации ответа студент не обосновывает свои суждения. На часть дополнительных вопросов студент затрудняется дать ответ или дает неверные ответы.

Оценка «Неудовлетворительно» выставляется студенту, который демонстрирует разрозненные, бессистемные знания; беспорядочно и неуверенно излагает материал; не умеет выделять главное и второстепенное, не умеет соединять теоретические положения с практикой, не устанавливает межпредметные связи; допускает грубые ошибки при определении сущности раскрываемых понятий, явлений, вследствие непонимания их существенных и несущественных признаков и связей; дает неполные ответы, логика и последовательность изложения которых имеют существенные и принципиальные нарушения, в ответах отсутствуют выводы. Дополнительные и уточняющие вопросы преподавателя не приводят к коррекции ответов студента. На основную часть дополнительных вопросов студент затрудняется дать ответ или дает неверные ответы.

**Приложение № 2 к рабочей программе дисциплины
«Моделирование и проектирование
государственных информационных систем»**

Методические указания для студентов по освоению дисциплины

Основной формой изложения учебного материала по дисциплине «Моделирование и проектирование государственных информационных систем» являются лекции, причем в достаточно большом объеме. По большинству тем предусмотрены практические занятия. Для успешного освоения дисциплины очень важно решение достаточно большого количества задач, как в аудитории, так и самостоятельно в качестве домашних заданий. Примеры решения задач разбираются на лекциях и практических занятиях, при необходимости по наиболее трудным темам проводятся дополнительные консультации. Основная цель решения задач – помочь усвоить фундаментальные понятия и основы промышленной разработки.

Задания для самостоятельного решения формулируются на лекциях и практических занятиях. В качестве заданий для самостоятельной работы дома студентам предлагаются задачи, аналогичные разобранным на лекциях и практических занятиях или немного более сложные, которые являются результатом объединения нескольких базовых задач. Полный список заданий для самостоятельной работы по темам (разделам) дисциплины приведен в ЭУК в LMS Moodle «Моделирование и проектирование государственных информационных систем». Вопросы, возникающие в процессе или по итогам решения этих задач, можно задать на консультациях или в форуме (чате) в ЭУК в LMS Moodle.