

Документ подписан простой электронной подписью  
 Информация о владельце:  
 ФИО: Мочалин Константин Сергеевич  
 Должность: И.о. ректора  
 Дата подписания: 30.05.2026 15:18:48  
 Уникальный программный ключ:  
 b7695d6b97247fced4385685adb0d9f8e6f2cdf

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА**

**Федеральное государственное бюджетное  
 образовательное учреждение высшего образования  
 "Сибирский государственный университет водного транспорта"**

**Б1.О.10**

**Разработка и эксплуатация сложных информационных систем  
 рабочая программа дисциплины (модуля)**

Закреплена за кафедрой	<b>Информационных систем</b>
Образовательная программа	09.04.02 Направление подготовки "Информационные системы и технологии" Направленность "Проектирование информационных систем и их компонентов" год начала подготовки 2026
Квалификация	<b>магистр</b>
Форма обучения	<b>очная</b>
Общая трудоемкость	<b>5 ЗЕТ</b>

Часов по учебному плану	180	Виды контроля на курсах: экзамен 3
в том числе:		
аудиторные занятия	30	
самостоятельная работа	110	
часов на контроль	36	

**Распределение часов дисциплины по семестрам**

Семестр (<Курс>.<Семестр на курсе>)	3 (2.1)		Итого	
	10 3/6			
Неделя	уп	ип	уп	ип
Лекции	10	10	10	10
Практические	20	20	20	20
Иная контактная работа	4	4	4	4
Итого ауд.	30	30	30	30
Контактная работа	34	34	34	34
Сам. работа	110	110	110	110
Часы на контроль	36	36	36	36
Итого	180	180	180	180

Рабочая программа дисциплины

**разработана в соответствии с ФГОС:**

Федеральный государственный образовательный стандарт высшего образования - магистратура по направлению подготовки 09.04.02 Информационные системы и технологии (приказ Минобрнауки России от 19.09.2017 г. № 917)

**составлена на основании учебного плана образовательной программы:**

09.04.02 Направление подготовки "Информационные системы и технологии"  
Направленность "Проектирование информационных систем и их компонентов"  
год начала подготовки 2026

**Рабочую программу составил(и):**

*ст. преподаватель, Вакуленко А.В.*

Рабочая программа одобрена на заседании кафедры

Заведующий кафедрой Моторин Сергей Викторович

### 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1	Целью дисциплины является обеспечение базового уровня знаний и навыков, необходимых для осуществления научно-технической деятельности в области программной инженерии.
-----	--

### 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Цикл (раздел) ООП:	Б1.О
<b>2.1</b>	<b>Требования к предварительной подготовке обучающегося:</b>
2.1.1	Инженерия информационных систем
2.1.2	Системы поддержки принятия решений
2.1.3	Технологическая (проектно-технологическая) практика
2.1.4	Интеллектуальные системы и технологии
2.1.5	Методы и модели информационных процессов и систем
2.1.6	Технологическая (проектно-технологическая) практика.
2.1.7	Отраслевые информационные технологии
<b>2.2</b>	<b>Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:</b>
2.2.1	Научно-технический семинар
2.2.2	Проектный семинар

### 3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

**ОПК-5: Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем;**

ОПК-5.1: Использует современное программное обеспечение для информационных и автоматизированных систем, включая его модернизацию

ОПК-5.2: Имеет опыт инсталлирования и разработки современного программного и аппаратного обеспечения

**ПК-1: Управление работами по сопровождению и проектами создания (модификации) ИС, автоматизирующих задачи организационного управления и бизнес-процессы**

ПК-1.6: Организационное и технологическое обеспечение проектирования и дизайна ИС

ПК-1.7: Организационное и технологическое обеспечение разработки баз данных ИС

**В результате освоения дисциплины обучающийся должен**

<b>3.1</b>	<b>Знать:</b>
3.1.1	Основные парадигмы и технологии программирования
3.1.2	Предмет, основные понятия и принципы программной инженерии
3.1.3	Методы эффективного управления разработкой программных средств
<b>3.2</b>	<b>Уметь:</b>
3.2.1	Проводить анализ требований к программному обеспечению
3.2.2	Обосновывать выбор модели жизненного цикла программного обеспечения
3.2.3	Применять методы контроля качества и тестирования программного обеспечения
3.2.4	Планировать комплекс работ по разработке программных средств и проектов
<b>3.3</b>	<b>Владеть:</b>
3.3.1	Навыками разработки алгоритмов и программных средств
3.3.2	Навыками сопровождения и реинженерии программного обеспечения
3.3.3	Навыками управления проектами программных средств в команде

4. СТРУКТУРА ДИСЦИПЛИНЫ (МОДУЛЯ)					
Вид занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Литература	ПрПо дгот
Раздел	<b>Раздел 1.</b>				
Лек	Метологии devops и angle /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1	0
Ср	Метологии devops и angle /Ср/	3	18	Л1.1 Л1.2Л2.1 Л2.2Л3.1	0
Лек	Программное обеспечение для создания и конфигурирования виртуальной среды разработки /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1	0
Пр	Создания и конфигурирования виртуальной среды разработки ПО Vagrant /Пр/	3	4	Л1.1 Л1.2Л2.1 Л2.2Л3.1	0
Ср	Создания и конфигурирования виртуальной среды разработки ПО Vagrant /Ср/	3	20	Л1.1 Л1.2Л2.1 Л2.2Л3.1	0
Лек	Программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1	0
Пр	Контейнеризатор приложений в среде Docker /Пр/	3	4	Л1.1 Л1.2Л2.1 Л2.2Л3.1	0
Ср	Контейнеризатор приложений в среде Docker /Ср/	3	20	Л1.1 Л1.2Л2.1 Л2.2Л3.1	0
Лек	Способы автоматизированного создания, тестирования и развертывания программного обеспечения /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1	0
Пр	Jenkins - средство автоматизации /Пр/	3	6	Л1.1 Л1.2Л2.1 Л2.2Л3.1	0
Ср	Jenkins - средство автоматизации /Ср/	3	26	Л1.1 Л1.2Л2.1 Л2.2Л3.1	0
Лек	Программное обеспечение для оркестровки контейнеризированных приложений — автоматизации их развёртывания /Лек/	3	2	Л1.1 Л1.2Л2.1 Л2.2Л3.1	0
Пр	Автоматизации и развёртывания ИС при помощи ПО Kubernetes /Пр/	3	6	Л1.1 Л1.2Л2.1 Л2.2Л3.1	0
Ср	Автоматизации и развёртывания ИС при помощи ПО Kubernetes /Ср/	3	26	Л1.1 Л1.2Л2.1 Л2.2Л3.1	0
ИКР	Экзамен /ИКР/	3	4		0

#### 5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

<p>1 Метологии devops и angle Методология автоматизации технологических процессов сборки, настройки и развёртывания программного обеспечения.</p> <p>2 Программное обеспечение для создания и конфигурирования виртуальной среды разработки Создания и конфигурирования виртуальной среды разработки. Обеспечения виртуализации. Средства управления конфигурациями.</p> <p>3 Программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации Автоматизации развёртывания и управления приложениями. Поддержка контейнеризации. Контейнеризатор приложений</p> <p>4 Способы автоматизированного создания, тестирования и развертывания программного обеспечения Средство автоматизации с открытым исходным кодом. Тестирования и развертывания программного обеспечения.</p> <p>5 Программное обеспечение для оркестровки контейнеризированных приложений — автоматизации их развёртывания Оркестровки контейнеризированных приложений. Масштабирование и координация.</p>
--

## 6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

### 6.1. Перечень видов оценочных средств

Вопросы к практическим работам  
Вопросы к экзамену

### 6.2. Темы письменных работ

Не предусмотрен

### 6.3. Контрольные вопросы и задания

1. Что является основной целью методологии DevOps?
  - A) Увеличение количества промежуточной документации.
  - B) Сокращение времени между внесением изменения в код и его выходом в стабильную эксплуатацию (time-to-market)
  - C) Полное разделение обязанностей разработчиков и системных администраторов.
  - D) Строгое следование водопадной модели разработки.
  
2. Что такое «непрерывная интеграция» (Continuous Integration, CI)?
  - A) Ручное развертывание приложений на продакшн-сервера раз в месяц.
  - B) Автоматизированный процесс развертывания приложения в контейнерах.
  - C) Практика частого слияния кода всех разработчиков в общую ветку с автоматической сборкой и тестированием
  - D) Методология планирования спринтов.
  
3. Какой из принципов Agile НЕ является прямым предшественником практик DevOps?
  - A) Реакция на изменения важнее следования плану.
  - B) Исчерпывающая документация важнее рабочего программного обеспечения.
  - C) Частые поставки рабочего программного обеспечения.
  - D) Непосредственное общение между заказчиком и разработчиком.
  
4. Что такое «канбан» в контексте Agile/DevOps?
  - A) Ежедневная встреча команды.
  - B) Метод оценки сложности задач.
  - C) Система визуализации рабочего потока и ограничения работы в процессе (WIP)
  - D) Фреймворк для масштабирования Agile на большие компании.
  
5. Что из перечисленного лучше всего описывает практику «Непрерывное развертывание» (Continuous Deployment)?
  - A) Любое изменение, прошедшее все стадии CI-пайплайна, автоматически выкатывается на продакшен.
  - B) Развертывание на продакшен происходит только после одобрения менеджера.
  - C) Любое изменение, прошедшее все стадии CI-пайплайна, автоматически выкатывается на продакшен.
  - D) Развертывание происходит ежеквартально большими релизами.
  
6. Что такое «Инфраструктура как код» (Infrastructure as Code, IaC)?
  - A) Ручная настройка серверов по документации.
  - B) Управление и provision инфраструктуры с помощью конфигурационных файлов, которые можно версионировать и тестировать
  - C) Использование только физических серверов.
  - D) Написание кода приложения, который напрямую управляет сетевыми коммутаторами.
  
7. Какое ПО является отраслевым стандартом для создания локальных виртуальных машин?
  - A) Docker
  - B) Kubernetes
  - C) Oracle VirtualBox или VMware Workstation
  - D) Jenkins
  
8. Какое средство позволяет описывать конфигурацию виртуальной среды в коде (например, для создания одинаковых VM для всех разработчиков)?
  - A) Git
  - B) Vagrant
  - C) Ansible
  - D) Docker Compose
  
9. Какой инструмент НЕ является системой управления конфигурациями (Configuration Management)?
  - A) Vagrant (это инструмент provisioningu, а не полноценная CM-система)
  - B) Ansible
  - C) Chef
  - D) Puppet
  
10. Что из перечисленного обеспечивает аппаратную виртуализацию?
  - A) Docker
  - B) Hyper-V

- C) Vagrant
- D) Terraform

11. Какой инструмент использует декларативный язык для описания инфраструктуры и является независимым от облачного провайдера?

- A) VMware vSphere
- B) Terraform
- C) VirtualBox
- D) Packer

12. Для чего в первую очередь используется инструмент Packer?

- A) Для оркестровки контейнеров.
- B) Для управления виртуальными машинами в runtime.
- C) Для создания «образов» машин (VM images, AMI) из единого источника конфигурации
- D) Для автоматического развертывания приложений.

13. Что такое Docker?

- A) Система управления конфигурациями.
- B) Платформа для разработки, доставки и запуска приложений в контейнерах
- C) Средство виртуализации на уровне операционной системы.
- D) Инструмент для оркестровки виртуальных машин.

14. Какой файл используется для определения состава и конфигурации многоконтейнерного приложения Docker?

- A) Dockerfile
- B) Jenkinsfile
- C) docker-compose.yml
- D) Vagrantfile

15. Что делает инструкция FROM в Dockerfile?

- A) Копирует файлы из хостовой системы в контейнер.
- B) Определяет базовый образ, от которого строится новый образ
- C) Задает команду, которая будет выполнена при запуске контейнера.
- D) Объявляет переменные среды.

16. Какая команда Docker используется для сборки образа из Dockerfile?

- A) docker run
- B) docker create
- C) docker start
- D) docker build

17. Какой из перечисленных НЕ является аналогом/альтернативой Docker?

- A) Podman
- B) Containerd
- C) Kubernetes (это оркестратор, а не только runtime)
- D) rkt (Rocket)

18. Что из себя представляет Docker-образ?

- A) Запущенный экземпляр приложения.
- B) Неизменяемый шаблон с инструкциями для создания контейнера
- C) Файл для описания сети между контейнерами.
- D) Инструмент для мониторинга контейнеров.

19. Какое ПО является широко используемым open-source сервером для автоматизации CI/CD?

- A) GitLab
- B) Jenkins
- C) TeamCity
- D) Bamboo

20. Что такое «пайплайн» (pipeline) в CI/CD?

- A) Гибкая методология разработки.
- B) Сетевое соединение между серверами.
- C) Автоматизированный последовательный процесс сборки, тестирования и развертывания кода
- D) Система контроля версий.

21. Где обычно описывается конфигурация пайплайна в современных CI-системах (при подходе «Pipeline as Code»)?

- A) В интерфейсе администратора Jenkins.
- B) В отдельной базе данных.
- C) В файле (например, Jenkinsfile, .gitlab-ci.yml) в корне репозитория с кодом приложения

D) В логах выполнения сборки.

22. Какой этап CI/CD пайплайна обычно следует сразу после сборки (build)?

- A) Развертывание на продакшен.
- B) Запуск автоматических тестов (юнит-тестов, интеграционных)
- C) Ручное ревью кода.
- D) Создание документации.

23. Что из перечисленного является облачной SaaS-платформой для CI/CD, тесно интегрированной с Git?

- A) Jenkins
- B) GitHub Actions
- C) Apache Maven
- D) Nagios

24. Какой инструмент часто используется как зависимость в CI-пайплайне для управления зависимостями и сборки Java-проектов?

- A) Apache Maven или Gradle
- B) Docker Compose
- C) Kubernetes
- D) Terraform

25. Что является самой популярной open-source системой оркестровки контейнеров?

- A) Docker Swarm
- B) Kubernetes (K8s)
- C) Apache Mesos
- D) Nomad

26. Как называется минимальная и неделимая единица развертывания в Kubernetes?

- A) Докер (Docker)
- B) Образ (Image)
- C) Под (Pod)
- D) Сервис (Service)

27. Какой объект Kubernetes обеспечивает постоянное хранилище данных, независимое от жизненного цикла Pod?

- A) ConfigMap
- B) Secret
- C) PersistentVolume (PV)
- D) Deployment

28. Что такое «Deployment» в Kubernetes?

- A) Физический сервер в кластере.
- B) Объект, который описывает желаемое состояние группы идентичных Pod'ов и управляет их обновлением и масштабированием
- C) Внутренняя DNS-запись для доступа к Pod'ам.
- D) Способ изолировать ресурсы CPU и памяти.

29. Какой объект Kubernetes предоставляет стабильную конечную точку (endpoint) для доступа к группе динамически меняющихся Pod?

- A) Ingress
- B) Service
- C) Node
- D) ReplicaSet

30. Какой инструмент CLI является основным для взаимодействия с кластером Kubernetes?

- A) docker-cli
- B) kubectl
- C) helm
- D) kubeadm

#### 6.4. Методические материалы, определяющие процедуры оценивания

Итоговая оценка является арифметической суммой всех баллов полученных студентом в процессе изучения дисциплины. В учет итоговой оценки по данной методике принимается шкала оценивания каждого вида занятий по данной дисциплине: лекции, практики, лабораторные работы, семинары и т.д. Преподавателем на первом занятии озвучивается максимальное количество баллов которое можно получить за данный вид занятий. Вес каждого вида занятий в баллах зависит от объема этих занятий и утверждается на первом заседании кафедры в текущем учебном году.

Методика получения итоговой оценки по 4-х балльной шкале

- 5 (отлично)  $\geq 85$
- 4 (хорошо)  $75 \div 84$

3 (удовлетворительно)	51÷74
2 (неудовлетворительно)	≤50

## 7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

### 7.1 Рекомендуемая литература

#### 7.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год
Л1.1	Олейник Павел Петрович	Корпоративные информационные системы: для бакалавров и специалистов	Санкт-Петербург: Питер, 2012
Л1.2	Орлов Сергей Александрович, Цилькер Борис Яковлевич	Технологии разработки программного обеспечения: Современный курс по программной инженерии	Санкт-Петербург: Питер, 2012

#### 7.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год
Л2.1	Петров Владимир Николаевич	Информационные системы: учеб. пособие для студентов высш. учеб. заведений, обучающихся по направлению "Информатика и вычисл. техника"	Санкт-Петербург: Питер, 2003
Л2.2	Советов Борис Яковлевич, Водяхо А. И., Дубенецкий В. А.	Архитектура информационных систем: учебник	Москва: Изд. центр "Академия", 2012

#### 7.1.3. Методические разработки

	Авторы, составители	Заглавие	Издательство, год
Л3.1	Тушков Константин Юрьевич	Проектирование информационных систем: метод. указ. к курсовому проекту для студентов 4 курса электромех. фак.	Новосибирск: НГАВТ, 2008

## 8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Назначение	Оборудование
Помещение для самостоятельной работы обучающихся	Комплект учебной мебели; ПК – 6 шт., подключенных к сети "Интернет" и обеспечивающих доступ в электронную информационно-образовательную среду Университета.
Учебная аудитория для проведения занятий лекционного типа	Аудиторная доска; Комплект учебной мебели; Мультимедийное оборудование: проектор (стационарный), экран (стационарный), ПК (стационарный)
Учебная аудитория для проведения занятий лекционного типа	Аудиторная доска; Комплект учебной мебели; Мультимедийное оборудование: проектор (стационарный), Экран (стационарный), ПК (стационарный)
Компьютерный класс - Лаборатория информационных систем - учебная аудитория для проведения лабораторных занятий	Аудиторная доска; Комплект учебной мебели; ПК-9 шт. (в т.ч. преподавательский); Мультимедийное оборудование: проектор, экран, ПК (переносной)
Компьютерный класс - Лаборатория информационных систем - учебная аудитория для проведения лабораторных занятий	Аудиторная доска; Комплект учебной мебели; ПК-9 шт. (в т.ч. преподавательский); Мультимедийное оборудование: проектор, экран, ПК (переносной)