

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО

**Директор физтех-школы
радиотехники и компьютерных
технологий**

Д.А. Гаврилов

	Рабочая программа дисциплины (модуля)
по дисциплине:	Принципы разработки современного программного обеспечения
по направлению:	Прикладные математика и физика
профиль подготовки:	Радиотехника и компьютерные технологии Физтех-школа Радиотехники и Компьютерных Технологий кафедра микропроцессорных технологий в интеллектуальных системах управления
курс:	4
квалификация:	бакалавр

Семестры, формы промежуточной аттестации:

7 (осенний) - Дифференцированный зачет

8 (весенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 60 час.

семинары: 0 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 30 час.

Всего часов: 90, всего зач. ед.: 2

Программу составил: М.А. Саламатов, ассистент

Программа обсуждена на заседании кафедры микропроцессорных технологий в интеллектуальных системах управления 30.03.2023

Аннотация

В курсе "Принципы разработки современного программного обеспечения" рассматриваются основные подходы к способам создания современного программного обеспечения. Изучаются основные процессы разработки: каскадная разработка, итерационная, гибкие методологии. Рассматриваются подходы к организации индивидуальной работы. Студенты знакомятся с современными подходами к упаковке и развертыванию приложений: непрерывной интеграции, развертыванию и поставке. Также изучаются подходы к разработке приложений, предназначенных для развертывания в современных облачных средах.

1. Цели и задачи

Цель дисциплины

Целью курса является ознакомление и углубленное изучение процессов и способов организации внутрикомандного взаимодействия, используемых при проектировании, разработке и тестировании современных программных продуктов.

Задачи дисциплины

- Изучение следующих моделей процессов разработки ПО: классическая каскадная, итеративная, гибкие подходы, в т.ч. Scrum и test-driven development;
- рассмотрение подходов к проектному планированию, основ дизайн-мышления, подходов к самостоятельной организации и личной эффективности;
- изучение вопросов современных технологий разработки приложений и их развертывания в облачных средах.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1 Анализирует задачу, выделяя этапы ее решения, действия по решению задачи
	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки
	УК-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности
	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации
ПК-4 Способен критически оценивать применимость используемых методик и методов	ПК-4.1 Знает численные порядки величин, характерных для соответствующей профессиональной области
	ПК-4.3 Способен обосновать причинно-следственные отношения используемых понятий и моделей

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- Основные процессы, используемые при разработке современного программного обеспечения;
- основные техники организации личной эффективности;
- основные технологии, используемые при разработке приложений, разворачиваемых в облачных средах;
- основные концепции программирования на языке Go.

уметь:

- Разбираться в различных процессах, используемых при разработке современного ПО;
- разбираться в технологиях, используемых при создании, тестировании и развертывании современных облачных приложений.

владеть:

- Техниками выбора процессов, подходящих для того или иного вида проекта;
- основными техниками организации личной эффективности;
- инструментами и технологиями, используемыми при создании современных приложений, разворачиваемых в облачных средах;
- языком программирования Go для создания современных приложений, разворачиваемых в облачных средах.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Определение проекта и проектного управления	2			1
2	Основные модели управления проектами и разработки ПО	8			4
3	Фреймворки, используемые в современной разработке ПО	4			2
4	Техники проектного планирования	4			2
5	Методологии дизайн-мышления	2			1
6	Техники личной эффективности	4			2
7	Техники подготовки презентаций	6			3
8	Введение в язык программирования Go: среда разработки, алгоритмы и структуры данных, сторонние библиотеки	10			5
9	Сборка проекта в дистрибутив	2			1
10	Автоматизация тестирования	2			1
11	Многопоточность в языке Go, проектирование REST API на Go	4			2
12	Современные подходы к развертыванию приложений	2			1
13	Виртуализация и ее типы, контейнеры, Docker, Kubernetes	6			3

14	Непрерывная интеграция, поставка, развертывание	4			2
Итого часов		60			30
Подготовка к экзамену		0 час.			
Общая трудоёмкость		90 час., 2 зач.ед.			

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 7 (Осенний)

1. Определение проекта и проектного управления

1.1 Определение проекта.

1.2 Определение “проектного треугольника”.

1.3 Основные задачи, решаемые в рамках проектного управления.

2. Основные модели управления проектами и разработки ПО

2.1 Классическая каскадная модель в определении Уинстона Ройса и ее интерпретации.

2.2 Итерационная модель и предпосылки ее появления.

2.3 Манифест гибкой разработки программного обеспечения.

3. Фреймворки, используемые в современной разработке ПО

3.1 Фреймворки для реализации манифеста гибкой разработки программного обеспечения: Scrum, test-driven development;

3.2 Введение в понятие непрерывной интеграции, поставки, развертывания.

3.3 Краткое содержание DevOps методологии.

4. Техники проектного планирования

4.1 Техники оценки сроков в проекте.

4.2 Введение в управление проектными рисками.

4.3 Способы оценки бюджета проекта.

5. Методологии дизайн-мышления

5.1 Базовые принципы дизайн-мышления.

5.2 5 стадий дизайн мышления: карта эмпатий, процесс определения требований, формирование идей, прототипирование, тестирование.

5.3 Примеры использования дизайн мышления в индустрии.

6. Техники личной эффективности

6.1 Модели мышления Дэниела Канемана и Тима Урбана.

6.2 Способы формулирования задач.

6.3 Основные методы работы с проектами.

6.4 Способы настройки инструментария для личной эффективности.

7. Техники подготовки презентаций

7.1 Основные составляющие презентаций: история, визуализация, подача.

7.2 Формулировка цели презентации, анализ аудитории.

- 7.3 Способы написания сценария презентации, проблематизация и представление решений.
- 7.4 Базовые подходы к созданию слайдов презентации.
- 7.5 Основные техники подачи.

Семестр: 8 (Весенний)

8. Введение в язык программирования Go: среда разработки, алгоритмы и структуры данных, сторонние библиотеки

- 8.1 Ключевые особенности языка программирования Go.
- 8.2 Области применения Go.
- 8.3 Компиляция Go и зависимость от платформы.
- 8.4 Типичная структура проекта в Go.
- 8.5 Зависимости, тесты, debug, сборка.
- 8.6 Типы данных: строки, константы, массивы, срезы, отображения, структуры.
- 8.7 Работа с основными типами данных: условные конструкции, циклы.
- 8.8 Интерфейсы.
- 8.9 Пакеты: containers и sort.
- 8.10 Сериализация, десериализация.
- 8.11 Переменные стека или кучи.
- 8.12 Трехцветный алгоритм для сборки мусора.

9. Сборка проекта в дистрибутив

- 9.1 Инструментарий для сборки проекта в дистрибутив.
- 9.2 Техники сборки продуктов в дистрибутив.
- 9.3 Виды дистрибутивов.

10. Автоматизация тестирования

- 10.1 Техники автоматизации тестирования.
- 10.2 Инструментарий для автоматизации тестирования.
- 10.3 Способы запуска и исполнения автоматизированных тестов в системах непрерывной интеграции.

11. Многопоточность в языке Go, проектирование REST API на Go

- 11.1 Запуск горутин в Golang.
- 11.2 Создание нескольких горутин.
- 11.3 Каналы Golang в многопоточности.
- 11.4 Использование select в канале горутин.
- 11.5 Каналы nil в Golang ничего не делают.
- 11.6 Блокировка и deadlock в Golang.
- 11.7 Объединение горутин в конвейер.
- 11.8 HTTP методы: GET, POST, PUT, DELETE.

12. Современные подходы к развертыванию приложений

- 12.2 Преимущества и недостатки каждого из методов.

13. Виртуализация и ее типы, контейнеры, Docker, Kubernetes

- 13.1 Определение виртуализации ресурсов.
- 13.2 Преимущества и недостатки использования виртуализации.
- 13.3 Типы виртуализации: виртуальные машины, контейнеры.

13.4 Работа с Docker: Docker file, registry, основные команды docker.

13.5 Оркестраторы контейнеров: Docker swarm, Kubernetes.

13.6 Основные компоненты Kubernetes: Node, pod, stateful set.

13.7 Основные команды Kubernetes.

14. Непрерывная интеграция, поставка, развертывание

14.1 Определение понятий непрерывной интеграции, поставки и развертывания.

14.2 Преимущества и недостатки каждого из методов.

14.3 Обзор инструментов для непрерывной интеграции, поставки и развертывания

14.4 Основные принципы развертывания: Recreate deployment, rolling deployment, ramped deployment, blue/green deployment, A/B testing deployment, shadow deployment, canary deployment.

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

- Необходимое оборудование практических занятий: компьютер, проектор, подключение к сети «Интернет»;
- необходимое программное обеспечение: MS Office Power Point.

6.Перечень рекомендуемой литературы

Основная литература

1. «A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Seventh Edition and The Standard for Project Management », Project Management Institute, 2021
2. Джефф Сазерленд, «Scrum. Революционный метод управления проектами», Манн, Иванов и Фербер, 2022
3. Бек Кент, “Экстремальное программирование. Разработка через тестирование”, Питер, 2022

Дополнительная литература

1. Лейфер Ларри, Леврик Михаэль, Линк Патрик, “Дизайн-мышление. От инсайта к новым продуктам и рынкам”, Питер, 2020
2. Максим Дорофеев, “Путь джедая. Поиск собственной методики продуктивности”, Манн, Иванов и Фербер, 2023
3. Алексей Каптерев, “Хорошая, плохая, продающая. Мастерство презентации 2.0”, Манн, Иванов и Фербер, 2020
4. Алексей Каптерев, “Мастерство презентации”, Манн, Иванов и Фербер, 2013
5. Erik St. Martin, William Kennedy, Brian Ketelsen, “Go in Action”, Manning, 2015
- [10] Henry van Merode, “Continuous Integration (CI) and Continuous Delivery (CD): A Practical Guide to Designing and Developing Pipelines”, Apress, 2023

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

1. Agile-манифест разработки программного обеспечения, интернет-ресурс,
<http://agilemanifesto.org/iso/ru/manifesto.html>
2. Поисковые системы.

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

Необходимое программное обеспечение: MS Office Power Point

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Студент, изучающий дисциплину, должен с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике. Успешное освоение курса требует напряжённой самостоятельной работы студента. В программе курса приведено минимально необходимое время для работы студента над темой. Самостоятельная работа включает в себя:

- чтение и конспектирование рекомендованной литературы;
- проработку учебного материала (по конспектам, по учебной и научной литературе);
- подготовку к дифференцированным зачетам.

Руководство и контроль за самостоятельной работой студента осуществляется в форме индивидуальных консультаций.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов, следует обращаться за консультациями к преподавателю.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Прикладные математика и физика
профиль подготовки:	Радиотехника и компьютерные технологии Физтех-школа Радиотехники и Компьютерных Технологий кафедра микропроцессорных технологий в интеллектуальных системах управления
курс:	4
квалификация:	бакалавр
Семестры, формы промежуточной аттестации:	
7 (осенний) - Дифференцированный зачет	
8 (весенний) - Дифференцированный зачет	
Разработчик:	М.А. Саламатов, ассистент

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1 Анализирует задачу, выделяя этапы ее решения, действия по решению задачи
	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки
	УК-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности
	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации
ПК-4 Способен критически оценивать применимость используемых методик и методов	ПК-4.1 Знает численные порядки величин, характерных для соответствующей профессиональной области
	ПК-4.3 Способен обосновать причинно-следственные отношения используемых понятий и моделей

2. Показатели оценивания компетенций

В результате изучения дисциплины «Принципы разработки современного программного обеспечения» обучающийся должен:

знать:

- Основные процессы, используемые при разработке современного программного обеспечения;
- основные техники организации личной эффективности;
- основные технологии, используемые при разработке приложений, разворачиваемых в облачных средах;
- основные концепции программирования на языке Go.

уметь:

- Разбираться в различных процессах, используемых при разработке современного ПО;
- разбираться в технологиях, используемых при создании, тестировании и развертывании современных облачных приложений.

владеть:

- Техниками выбора процессов, подходящих для того или иного вида проекта;
- основными техниками организации личной эффективности;
- инструментами и технологиями, используемыми при создании современных приложений, разворачиваемых в облачных средах;
- языком программирования Go для создания современных приложений, разворачиваемых в облачных средах.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

Контрольные вопросы для текущего контроля:

1. Перечислите основные компоненты “проектного треугольника”
2. Опишите основные компоненты и техники проектного планирования
3. Опишите основные компоненты классической проектной модели Уинстона Ройса
4. Расскажите основные этапы разработки ПО по фреймворку Scrum
5. Перечислите известные вам составляющие методологии DevOps
6. Перечислите основные типы рисков в проектном управлении и методы работы с рисками
7. Опишите работу систем принятия решений по Даниэлу Канеману
8. Перечислите основные инструменты дизайн-мышления
9. Приведите примеры использования инструментов дизайн-мышления
10. Расскажите об основных методах создания сценариев публичного выступления
11. Опишите ключевые особенности языка программирования Go
12. Какие компоненты Kubernetes вам известны?
13. Какие инструменты для реализации CI/CD подхода вам известны?
14. Опишите принципы работы Go-рутин
15. Перечислите известные вам HTTP методы и способы работы с ними в языке программирования Go

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

Перечень контрольных вопросов для сдачи дифференцированного зачёта в осеннем семестре:

1. Перечислите известные вам техники управления проектами: в чем заключается отличие между ними?
2. Перечислите основные известные вам сценарии управления позитивными и негативными рисками на проекте
3. Перечислите известные вам техники оценки сроков и бюджета проекта
4. Каким образом фреймворк Scrum реализует Agile-манифест разработки ПО?
5. Какие роли, церемонии и артефакты в Scrum вам известны?
6. В чем заключается роль DevOps инженера?
7. Приведите примеры результатов работы команды на каждой из пяти стадий дизайн-мышления
8. Напишите сценарий сессии дизайн-мышления для команды произвольного проекта
9. Опишите работу систем мышления по Дэниэлу Канеману и Тиму Урбану? В чем заключаются их сходства и отличия?
10. Перечислите основные критерии для написания персонального списка задач
11. Расскажите об известных вам способах работы с инструментарием для ведения списков задач
12. Перечислите основные компоненты публичного выступления
13. Перечислите известные вам способы анализа аудитории для публичного выступления
14. Перечислите элементы, рекомендованные к изображению на слайдах публичного выступления
15. Какие базовые техники подачи в рамках публичного выступления вам известны?

Перечень контрольных вопросов для сдачи дифференцированного зачёта в весеннем семестре:

1. Перечислите основные этапы компиляции проекта на языке программирования Go
2. Перечислите компоненты типичного проекта на языке программирования Go
3. Перечислите методы работы с основными типами данных: строки, константы, массивы, срезы, отображения, структуры

4. Приведите пример взаимодействия переменных стека и кучи в языке программирования Go
5. Приведите примеры использования пакетов `containers` и `sort` в языке программирования Go
6. Опишите принцип работы трехцветного алгоритма для сборки мусора
7. Перечислите известные вам виды дистрибутивов для проектов на языке программирования Go
8. Перечислите основные известные вам техники автоматизации тестирования
9. Для произвольного проекта на языке программирования Go приведите примеры различных видов тестов, а также опишите сценарии их автоматизации
10. Приведите примеры использования горутин в языке программирования Go
11. Опишите типичный сценарий объединения горутин в конвейер
12. Приведите примеры GET, POST, PUT, DELETE запросов на языке программирования Go
13. Для произвольного проекта на языке программирования Go выберите сценарий развертывания. Опишите детали реализации, а также преимущества выбранного сценария
14. Дайте определения понятию виртуализации ресурсов. Перечислите известные вам типы виртуализации, включая контейнерную виртуализацию
15. Перечислите основные этапы создания Docker контейнеров. Опишите функциональность Docker –file и Docker registry
16. Перечислите примеры и основные функциональности оркестраторов Docker контейнеров. Опишите стадии развертывания Kubernetes кластера.
17. Для произвольного проекта опишите набор контейнеров и их распределение в рамках Kubernetes кластера. Опишите основные сущности и команды, с которыми вы будете работать
18. Дайте определение понятиям непрерывной интеграции, поставки и развертывания. В чем заключаются отличия между ними?
19. Какие инструменты организации непрерывной интеграции вам известны? Для произвольного проекта опишите этапы настройки выбранного инструмента непрерывной интеграции.
20. Для произвольного проекта на языке программирования Go опишите процесс непрерывного развертывания с публикацией конечного решения в виде Docker-контейнеров в Kubernetes

Критерии оценивания

Оценка отлично (10) выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (9) выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (8) выставляется студенту, показавшему систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.

Оценка хорошо (7) выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты.

Оценка хорошо (6) выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Оценка хорошо (5) выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.

Оценка удовлетворительно (4) выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.

Оценка удовлетворительно (3) выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.

Оценка неудовлетворительно (2) выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных принципов и не умеет использовать полученные знания при решении типовых задач.

Оценка неудовлетворительно (1) выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубейшие ошибки в формулировках базовых понятий дисциплины и вообще не имеет навыков решения типовых практических задач.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Дифференцированный зачет проводится в устной форме.

При проведении дифференцированного зачета обучающемуся предоставляется 20 минут на подготовку. Опрос обучающегося проводится в течение 30 минут.

Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины, а также справочной литературой, вычислительной техникой, конспектами лекций.