

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

УТВЕРЖДЕНО
Директор

А.В. Малеев

	Рабочая программа дисциплины (модуля)
по дисциплине:	Инструменты разработчика
по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии
курс:	1
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 1 (осенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 48 час.

Всего часов: 108, всего зач. ед.: 3

Программу составил: И.С. Конов, канд. техн. наук, доцент

Программа обсуждена на заседании высшей школы программной инженерии 19.03.2025

Аннотация

Цель дисциплины – сформировать у студентов навыки промышленного программирования, изучение объектно-ориентированных языков программирования с виртуальной машиной и байт-кодом. Дисциплина направлена не только на формирование практических навыков, но и на формирование общей культуры разработки и навыков промышленного программирования.

1. Цели и задачи

Цель дисциплины

- сформировать у студентов навыки проектной деятельности в сфере информационных технологий, современными инструментами командной работы.

Задачи дисциплины

- проведение информационных и обучающих семинаров по тематике студенческих проектов, в том числе по тематике разработке на Java, C++ и др.;
- проведение семинаров для обсуждения постановок задач проектной деятельности, оценки прогресса по выполнению проектов;
- изучение инструментов разработчика для организации командной работы;
- изучение современных систем контроля версий разрабатываемого программного обеспечения (Git).

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.2 Владеет навыками создания программного обеспечения для ЭВМ и систем различной архитектуры
	ОПК-2.1 Способен использовать информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности
ОПК-7 Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой	ОПК-7.3 Умеет выявлять узкие места в процессе разработки и предлагать методы и инструменты для его оптимизации
	ОПК-7.2 Понимает принципы работы баз данных и умеет проектировать структуру данных для эффективного хранения информации
	ОПК-7.1 Обладает навыками создания и выполнения тестовых сценариев для выявления ошибок в программном обеспечении
ПК-3 Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение	ПК-3.4 Знает, как определять оптимальные методы и средства проектирования программного обеспечения и структур данных
	ПК-3.3 Умеет излагать основные принципы построения и виды архитектуры программного обеспечения, методы и средства проектирования программного обеспечения, методологии разработки программного обеспечения и технологии программирования
	ПК-3.2 Умеет выбирать языки программирования для написания программного кода с учетом технического задания
	ПК-3.1 Различает синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования
	ПК-5.4 Имеет практический опыт разработки, внедрения и адаптации прикладного программного обеспечения информационных систем в различных программных средах

ПК-5 Способен проектировать, разрабатывать, внедрять, сопровождать и снимать с эксплуатации информационные системы	ПК-5.3 Умеет разрабатывать прототип информационных систем в соответствии с требованиями и проводить его тестирование для проверки корректности архитектурных решений
	ПК-5.2 Умеет определять оптимальные методы и инструменты разработки, внедрения и адаптации прикладного программного обеспечения информационных систем
	ПК-5.1 Умеет описывать архитектуру, устройство и функционирование информационных систем

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- основы проектной деятельности, принципы командной работы, постановки задач на разработку;
- систему базовых понятий программной инженерии: программное обеспечение (ПО), жизненный цикл ПО; фаза жизненного цикла ПО, программный проект, требование к ПО, ресурсы и роли в программных проектах, проектный план;
- принципы версионирования программного проекта с помощью системы контроля версий Git.

уметь:

- использовать системы контроля версий Git для ведения программных продуктов;
- презентовать предложения по разработке ПО;
- участвовать в обсуждении постановок задач на проектную деятельность.

владеть:

- современными средствами для контроля версий Git;
- средствами для подготовки и демонстрации презентаций по результатам проектной деятельности.

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	Основы проектной деятельности	4	4		8
2	Разработка заданий на проектную деятельность	4	4		8
3	Презентация постановочных докладов	6	6		8
4	Системы контроля версий разрабатываемого программного обеспечения	6	6		8
5	Работа с Git-репозиторием	4	4		8
6	Презентация и защита проектов	6	6		8
Итого часов		30	30		48
Подготовка к экзамену		0 час.			
Общая трудоёмкость		108 час., 3 зач.ед.			

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

Семестр: 1 (Осенний)

1. Основы проектной деятельности

Проект. Типология проектов. Основные характеристики проектной деятельности. Понятие о внеаудиторной самостоятельной работе студента по поиску информации для обеспечения проекта. Понятие об авторском праве. Необходимые условия для организации проектной деятельности. Формы самостоятельной работы. Система регулярного контроля качества самостоятельной части проектной работы.

2. Разработка заданий на проектную деятельность

Постановка задания, выявление требований заказчика, согласование функциональных требований к программному продукту или сервису.

3. Презентация постановочных докладов

Подготовка презентаций с постановками задач на проектирование и разработку, выступление с докладами, обсуждение.

4. Системы контроля версий разрабатываемого программного обеспечения

Git. Введение. Установка и настройка. Git. Внутренняя реализация. Создание изменений: индексация и коммиты. Команды: init, status, add, commit.

5. Работа с Git-репозиторием

Git. Ветвление. Создание и удаление веток, переключение между ними. Команды: branch, checkout, status, log, diff. Перемещение курсора и отмена изменений. Команды git restore, git rm, git reset, git checkout, git commit, git revert.

6. Презентация и защита проектов

Подготовка презентаций с итоговыми докладами по проектам, выступления, обсуждение.

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

компьютер, мультимедийное оборудование (проектор).

6. Перечень рекомендуемой литературы

Основная литература

1. Системная и программная инженерия. Словарь-справочник, учебное пособие / В. К. Батоврин. — Москва, ДМК Пресс, 2010. — URL: <https://e.lanbook.com/book/1097> (дата обращения: 26.01.2021). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

Дополнительная литература

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

не требуется

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

веб-репозитории Git (Github, Gitlab).

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

Успешное освоение курса требует самостоятельной работы студента. В программе курса приведено минимально необходимое время для работы студента над темой.

Самостоятельная работа включает в себя:

- проработку учебного материала;
- подготовку к семинарам, разработку презентаций по проектам.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии
курс:	<u>1</u>
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 1 (осенний) - Дифференцированный зачет

Разработчик: И.С. Конов, канд. техн. наук, доцент

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.2 Владеет навыками создания программного обеспечения для ЭВМ и систем различной архитектуры
	ОПК-2.1 Способен использовать информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности
ОПК-7 Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой	ОПК-7.3 Умеет выявлять узкие места в процессе разработки и предлагать методы и инструменты для его оптимизации
	ОПК-7.2 Понимает принципы работы баз данных и умеет проектировать структуру данных для эффективного хранения информации
	ОПК-7.1 Обладает навыками создания и выполнения тестовых сценариев для выявления ошибок в программном обеспечении
ПК-3 Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение	ПК-3.4 Знает, как определять оптимальные методы и средства проектирования программного обеспечения и структур данных
	ПК-3.3 Умеет излагать основные принципы построения и виды архитектуры программного обеспечения, методы и средства проектирования программного обеспечения, методологии разработки программного обеспечения и технологии программирования
	ПК-3.2 Умеет выбирать языки программирования для написания программного кода с учетом технического задания
	ПК-3.1 Различает синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования
ПК-5 Способен проектировать, разрабатывать, внедрять, сопровождать и снимать с эксплуатации информационные системы	ПК-5.4 Имеет практический опыт разработки, внедрения и адаптации прикладного программного обеспечения информационных систем в различных программных средах
	ПК-5.3 Умеет разрабатывать прототип информационных систем в соответствии с требованиями и проводить его тестирование для проверки корректности архитектурных решений
	ПК-5.2 Умеет определять оптимальные методы и инструменты разработки, внедрения и адаптации прикладного программного обеспечения информационных систем
	ПК-5.1 Умеет описывать архитектуру, устройство и функционирование информационных систем

2. Показатели оценивания компетенций

В результате изучения дисциплины «Инструменты разработчика» обучающийся должен:

знать:

- основы проектной деятельности, принципы командной работы, постановки задач на разработку;
- систему базовых понятий программной инженерии: программное обеспечение (ПО), жизненный цикл ПО; фаза жизненного цикла ПО, программный проект, требование к ПО, ресурсы и роли в программных проектах, проектный план;
- принципы версионирования программного проекта с помощью системы контроля версий Git.

уметь:

- использовать системы контроля версий Git для ведения программных продуктов;
- презентовать предложения по разработке ПО;
- участвовать в обсуждении постановок задач на проектную деятельность.

владеть:

- современными средствами для контроля версий Git;
- средствами для подготовки и демонстрации презентаций по результатам проектной деятельности.

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

При подготовке презентации на постановочные и итоговые доклады следует ориентироваться на следующие общие рекомендации:

Время:

- Количество слайдов примерно соответствует длине доклада в минутах. Если у вас слайдов намного больше, чем времени, то вы просто не успеете показать все слайды, либо будете показывать их слишком быстро и аудитория не поймет доклада. Если у вас слайдов слишком мало, то это означает, что вы не эффективно их используете. Чаще всего времени не хватает. Лучше рассказать полностью об одной теме, чем не успеть рассказать о двух. Вне зависимости от степени важности второй темы.

Доклад:

- Повествование должно быть последовательным и логичным.
- Возврат к старому слайду — зачастую не удачное решение.
- Доклад следует делить на разделы.
- Если доклад длится более 15-20 минут, следует перед каждым разделом давать его краткий план.
- В каждом разделе не должно быть более 4-5 параграфов (иначе к тому времени, как вы дойдете до последнего параграфа, аудитория успеет забыть план этого раздела).-
- Названия разделов и параграфов должны быть краткими и ёмкими.
- Начинайте доклад с пояснения, о чем вы будете рассказывать.
- Завершайте свой доклад обобщением уже сказанных основных тезисов в более короткой и понятной форме. Люди наиболее внимательны в начале и конце доклада. Итоги — это ваш второй шанс донести главную мысль до слушателя.

Слайды:

- Каждый слайд должен иметь заголовок.
- Не полностью заполненный слайд лучше, чем переполненный. Обычно, в слайде должно быть от 20 до 40 слов. Разумный максимум — 80 слов.
- Делайте слайд проще. У аудитории всего около 50 секунд на его восприятие.
- Не стоит переоценивать аудиторию. Делайте доклад более доступным.
- Не показывайте в слайдах то, о чем не будете рассказывать.
- Избегайте сплошной текст. Лучше используйте нумерованные и маркированные списки.
- Не используйте уровень вложения в списках глубже двух. Лучше используйте схемы и диаграммы.
- Используйте краткие предложения или фразы.
- Не переносите слова.

Шрифты:

- Используйте не более двух шрифтов (один для заголовков, один для текста).
- Не используйте для заголовков и текста похожие шрифты.
- Не используйте для основного текста и заголовков декоративные, рукописные, готические, моноширинные шрифты.
- Шрифт в схемах и диаграммах должен совпадать с основным шрифтом текста.

- Размер шрифта стоит выбирать так, чтобы на слайде умещалось около 10-15 строк, не более.
- Для смыслового выделения текста используйте цвет или полужирную интенсивность.

Цвета:

- Аккуратно используйте цвета. Для добавления каждого нового цвета у вас должна быть веская причина.
- Будьте осторожны в использовании светлых цветов на белом фоне, особенно зеленого. То, что хорошо выглядит у вас на мониторе, плохо выглядит при докладе, поскольку мониторы, проекторы и принтеры по-разному представляют цвета. Используйте темные, насыщенные цвета, если у вас светлый фон.
- Используйте контрастные цвета. Нормальный текст должен быть черным на белом фоне, или как минимум, чем-то темным на чём-то очень светлом. Никогда не используйте такие сочетания как «светло-зеленый текст на не-слишком-темно-зеленом фоне».
- Тени уменьшают четкость без увеличения информативности. Не используйте тени только потому что это выглядит «красивей».
- Инверсные цвета (светлый текст на темном фоне) могут стать проблемой в светлых (не затемненных) помещениях. Инверсные цвета также тяжелее воспроизводятся в раздаточном материале и на прозрачках.

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

1. Системы контроля версий Git.
2. Установка и настройка Git.
3. Создание изменений: индексация и коммиты.
4. Ветвление. Создание и удаление веток, переключение между ними.
5. Перемещение курсора и отмена изменений.
6. Функциональные требования к программному продукту или сервису
7. Этапы постановки задач на проектирование и разработку
8. Команды: branch, checkout, status, log, diff.
9. Команды git restore, git rm, git reset, git checkout, git commit, git revert.
10. Команды: init, status, add, commit.Git.

Критерии оценивания

- оценка «отлично (10)» 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 (9)» выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений
- оценка «отлично (8)» выставляется студенту, показавшему всесторонние систематизированные, глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, и правильное обоснование принятых решений
- оценка «хорошо (7)» 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 (6)» выставляется студенту, если он 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 знает материал, и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

- 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 владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;
- оценка «удовлетворительно (3)» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет фрагментарно основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;
- оценка «неудовлетворительно (2)» выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач
- оценка «неудовлетворительно (1)» выставляется студенту, который не знает формулировок основных понятий дисциплины

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

Дифференцированный зачет проводится по итогам текущей успеваемости и сдачи заданий и других видов работ, предусмотренных программой дисциплины и (или) путем организации специального опроса, проводимого в устной и (или) письменной форме.