

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

**УТВЕРЖДЕНО**

**Директор физтех-школы  
прикладной математики и  
информатики**

**А.М. Райгородский**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Технологии программирования
<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Математическое моделирование и компьютерные технологии Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
<b>курс:</b>	1
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 2 (весенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 75 час.

Всего часов: 135, всего зач. ед.: 3

Количество контрольных работ, заданий: 2

Программу составил: В.В. Яковлев, канд. физ.-мат. наук, заведующий кафедрой

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 12.02.2024

## Аннотация

Курс посвящен базовым вопросам обеспечения и организации качественной промышленной разработки ПО. Рассматриваются необходимые инструменты и технологии для организации сборки ПО, тестирования, CI/CD процессов. Большое внимание уделяется базовым принципам построения архитектуры ПО, паттернам и антипаттернам разработки.

Заключительным этапом всего курса является дифференцированный зачет, целью которого является проверка теоретических знаний студентов, а также выявление практических навыков применения полученных знаний при выполнении практических заданий.

### 1. Цели и задачи

#### Цель дисциплины

Овладение студентами технологических приемов, повсеместно применяемых при разработке программного обеспечения.

#### Задачи дисциплины

- приобретение студентами навыков работы в командной строке, инструментами сборки и системами контроля версий;
- овладение студентами современными практиками разработки и типовыми шаблонами проектирования.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- шаблоны проектирования программного обеспечения.

уметь:

- работать с интерфейсом командной строки;
- выполнять сборку программ из исходных текстов и их отладку, без использования интегрированных средств разработки;
- пользоваться системами контроля версий;
- настраивать окружение для непрерывной интеграции разработки проекта;
- проектировать программное обеспечение таким образом, чтобы его поддержка осуществлялась коллективом из нескольких разработчиков.

владеть:

- навыками работы с GitLab и GitLab CI.

### 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	Введение в ОС Linux	3	3		5
2	Системы контроля версий	3	3		5
3	Процесс компиляции	4	4		5
4	Организация процесса сборки	3	3		5
5	Кросс-компиляция и методы отладки	3	3		10
6	Порождающие паттерны проектирования	4	4		10
7	Структурные паттерны	3	3		10
8	Поведенческие паттерны	3	3		10
9	Модели	4	4		15
Итого часов		30	30		75
Подготовка к экзамену		0 час.			
Общая трудоёмкость		135 час., 3 зач.ед.			

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

Семестр: 2 (Весенний)

##### 1. Введение в ОС Linux

Работа с командной строкой. Организация файловой системы.

##### 2. Системы контроля версий

Работа с ответвлениями. Организация совместной работы.

##### 3. Процесс компиляции

Стадии компиляции. Промежуточные артефакты сборки и их исследование.

##### 4. Организация процесса сборки

Сборка с помощью сценария Makefile. Высокоуровневые системы сборки. Непрерывная интеграция.

##### 5. Кросс-компиляция и методы отладки

Кросс-компиляция для другой архитектуры процессора. Кросс-компиляция для другой операционной системы. Отладка программ с использованием средств виртуализации.

##### 6. Порождающие паттерны проектирования

Синглтоны. Фабричные методы. Прототипы.

##### 7. Структурные паттерны

Адаптеры. Связки. Композиты. Прокси и декораторы.

8. Поведенческие паттерны

Стратегии. Интерпретатор, итератор, состояние. Стратегия «наблюдателя».

9. Модели

Паттерн «модель-контроллер-представление».

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

Учебная аудитория, оснащенная компьютером и мультимедийным оборудованием (проектор, звуковая система).

## **6. Перечень рекомендуемой литературы**

Основная литература

1. Введение в программирование , учебное пособие / И. Ю. Баженова, В. А. Сухомлин. — Москва, ИНТУИТ, 2016.— URL: <https://e.lanbook.com/book/100695> (дата обращения: 30.12.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

Дополнительная литература

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

<http://docs.oracle.com/javase/specs/jls/se8/html/index.html> - The Java Language Specification.  
<https://google-styleguide.googlecode.com/svn/trunk/javaguide.html> — Google Java Style Guide.

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

1. <https://git-scm.com/book/ru/v2>
2. <https://www.gnu.org/software/bash/manual/bash.html>

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

На лекционных занятиях используются мультимедийные технологии, включая демонстрацию презентаций.

Операционная система Linux с правами системного администратора.

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

Успешное освоение дисциплины требует:

- посещения студентом всех видов аудиторных занятий;
- ведения конспекта в ходе лекционных занятий;
- качественной самостоятельной подготовки к практическим занятиям, активной работы на них;
- активной самостоятельной и аудиторной работы студента;
- своевременной сдачи преподавателю заданий по аудиторным видам работ.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Математическое моделирование и компьютерные технологии Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
<b>курс:</b>	1
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 2 (весенний) - Дифференцированный зачет

**Разработчик:** В.В. Яковлев, канд. физ.-мат. наук, заведующий кафедрой

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Технологии программирования» обучающийся должен:

### знать:

- шаблоны проектирования программного обеспечения.

### уметь:

- работать с интерфейсом командной строки;
- выполнять сборку программ из исходных текстов и их отладку, без использования интегрированных средств разработки;
- пользоваться системами контроля версий;
- настраивать окружение для непрерывной интеграции разработки проекта;
- проектировать программное обеспечение таким образом, чтобы его поддержка осуществлялась коллективом из нескольких разработчиков.

### владеть:

- навыками работы с GitLab и GitLab CI.

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

Примеры контрольных заданий:

1. Написать программу, измеряющую время работы методов для нескольких стандартных реализаций интерфейсов List и Map, с использованием Dynamic Proxy.
2. Написать класс-наследник класса String, реализующий интерфейс Iterable, предоставляющий итератор со значениями типа char по своим символам.
3. Написать программу, которая скачивает 10 самых популярных страниц с сайта Хабрахабр и подсчитывает суммарный рейтинг комментариев, зарегистрированных по пользователям, с использованием сторонней библиотеки парсинга HTML.
4. Написать программу, которая предоставляет консольный интерфейс для добавления записей в базу данных книжного магазина с сущностями «Автор» и «Книга» (отношение один-ко-многим между автором и книгами), с использованием сторонней библиотеки для работы с базой данных sqlite3.
5. Написать программу, которая предоставляет консольный интерфейс для добавления записей в базу данных книжного магазина с сущностями «Автор» и «Книга» (отношение один-ко-многим между автором и книгами), с использованием сторонней библиотеки, реализующей Object-Relation Mapping.
6. Написать программу, которая запрашивает у Foursquare API и выводит популярные кафе в окрестности города, вводимого пользователем.
7. Написать программу, вычисляющую PageRank и найти ТОП 100 самых популярных страниц в википедии (русскоязычной, англоязычной).

8. Форматы XML, JSON, CSV.
9. Кодировки текстовых данных.
10. Система контроля версий git.

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

Перечень вопросов для дифференцированного зачета:

1. Сборщик мусора.
2. Массивы, многомерные массивы.
3. Методы класса Object. Контракт equals()/hashCode().
4. Интерфейсы.
5. Исключения: иерархия исключений, перехват исключений, декларация throws.
6. Порядок инициализации объекта.
7. Java Reflection API. Dynamic proxy.
8. Юнит-тестирование, средства JUnit.
9. Аннотации.
10. Итераторы.
11. Сериализация средствами стандартной библиотеки.
12. Лямбда-функции.
13. TCP/IP в Java. HTTP-сервер средствами Java.
14. Многопоточность: примитивы синхронизации в языке.
15. Многопоточность: Java Memory Model.
16. Многопоточность: инструменты библиотеки java.util.concurrent.
17. Коллекции стандартной библиотеки, потокобезопасность коллекций.

#### **Критерии оценивания**

Оценка "отлично":

- 10 - Полностью и вовремя решены все задачи без ошибок. Продемонстрирован грамотный подход к решению задач, реализованы оптимальные алгоритмы, код оформлен в едином удобочитаемом стиле.
- 9 - Полностью и вовремя решены все задачи без ошибок. Продемонстрирован грамотный подход к решению задач, реализованы оптимальные алгоритмы.
- 8 - Полностью и вовремя решены все задачи без ошибок. Продемонстрирован грамотный подход к решению задач.

Оценка "хорошо":

- 7 - Полностью решены все задачи. Допущены несущественные ошибки.
- 6 - Полностью решено большинство задач. В некоторых задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.
- 5 - Полностью решено две трети задач. В некоторых задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

Оценка "удовлетворительно":

- 4 - Полностью решено более половины задач. В остальных задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.
- 3 - Полностью решено более половины задач.

Оценка "неудовлетворительно":

- 2 - Решено менее половины задач.
- 1 - Не решено ни одной задачи.

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

Дифференцированный зачет может проводиться 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.

При проведении устного дифференцированного зачета обучающемуся предоставляется 30 минут на подготовку. Опрос обучающегося не должен превышать одного астрономического часа.

Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины, а также справочной литературой, конспектами лекций или другими материалами.