

**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

	Рабочая программа дисциплины (модуля)
по дисциплине:	Технологии программирования и операционные системы
по направлению:	Прикладная математика и информатика
профиль подготовки:	А1360: Передовые методы искусственного интеллекта Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
курс:	4
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 7 (осенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 30 час.

Всего часов: 90, всего зач. ед.: 2

Программу составил: О.Н. Ивченко, заведующий кафедрой

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 12.02.2024

Аннотация

Этот курс предоставляет студентам глубокие знания о технологиях программирования и операционных системах, необходимых для разработки программного обеспечения и понимания работы компьютерных систем. В рамках курса студенты также могут решать практические задачи и проекты, связанные с применением изученных концепций.

1. Цели и задачи

Цель дисциплины

- понимание основ программирования;
- изучение операционных систем;
- работа с различными языками программирования;
- практические навыки;
- понимание взаимодействия между программами и операционной системой.

Задачи дисциплины

- написание программ;
- работа с операционными системами;
- отладка и тестирование программ;
- оптимизация производительности;
- работа с сетью;
- выполнение проектов.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-3 Способен составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты)	ОПК-3.1 Знает основные правила оформления научных публикаций и научно-технической документации, в том числе с использованием прикладного программного обеспечения
	ОПК-3.2 Владеет на практике методологией составления научно-технических отчетов (проектов)
	ОПК-3.3 Владеет методами визуального и графического представления результатов научной (научно-технической, инновационной технологической) деятельности в виде отчетов, научных публикаций
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
	ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- основные принципы работы в Unix-системах;
- основные понятия и терминологию современных систем контроля версий, виртуализации, компьютерных сетей;
- основы программирования;
- технологии разработки ПО.

уметь:

- применять полученные знания для работы в командных проектах;
- работать с операционными системами;
- разрабатывать чистый поддерживаемый код.

владеть:

- инструментарием командной строки Unix;
- современными системами контроля версий и платформами виртуализации;
- методологиями разработки в 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	Семейство ОС Unix. Современные инструменты для работы в консоли	6	6		6
2	Системы контроля версий	6	6		6
3	Виртуализация	6	6		6
4	Непрерывная интеграция	6	6		6
5	Компьютерные сети	6	6		6
Итого часов		30	30		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. Семейство ОС Unix. Современные инструменты для работы в консоли

Семейство ОС Unix, основные дистрибутивы и их отличия. Основные командные интерпретаторы, их отличия. Утилита sed, язык awk. Использование Python как заменителя shell. Jupyter и TmpNb как его модификация. Терминальные мультиплексоры, автоматизация работы с ними.

2. Системы контроля версий

Системы контроля версий в современных проектах. CVS – первая система контроля версий. SVN. Современные VCS – Git, Mercurial и работа с ними. Работа над проектами в команде. Автоматизация работы с Git.

3. Виртуализация

Виртуализация. Её виды (контейнерная, на уровне ОС). Современные платформы виртуализации и работа с ними. Автоматизация работы с виртуальными окружениями.

4. Непрерывная интеграция

Непрерывная интеграция (continuous integration) и её этапы. Основные инструменты CI и их связь с VCS.

5. Компьютерные сети

Компьютерные сети, модель OSI, её уровни. Сетевые протоколы обмена информацией.

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. Основы работы с Linux, Электрон. версия печ. публикации / Н. М. Войтов. — Москва, ДМК Пресс, 2010
2. Современные операционные системы, Электрон. версия печ. публикации / Э. Таненбаум, Х. Бос ; [пер. с англ.]. — Санкт-Петербург, Питер, 2019

Дополнительная литература

1. Компьютерные науки. Деревья, операционные системы, сети, Электронная версия печатной публикации / И. Ф. Астахова, И. К. Астанин, И. Б. Крыжко, Е. А. Кубряков. — Москва, Физматлит, 2013

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

Не используются

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

Не требуется.

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 дисциплины требует:

- посещения студентом всех видов аудиторных занятий;
- ведения конспекта в ходе аудиторных 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)

по направлению: Прикладная математика и информатика
профиль подготовки: АІ360: Передовые методы искусственного интеллекта
Физтех-школа Прикладной Математики и Информатики
кафедра алгоритмов и технологий программирования
курс: 4
квалификация: бакалавр

Семестр, формы промежуточной аттестации: 7 (осенний) - Дифференцированный зачет

Разработчик: О.Н. Ивченко, заведующий кафедрой

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-3 Способен составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты)	ОПК-3.1 Знает основные правила оформления научных публикаций и научно-технической документации, в том числе с использованием прикладного программного обеспечения
	ОПК-3.2 Владеет на практике методологией составления научно-технических отчетов (проектов)
	ОПК-3.3 Владеет методами визуального и графического представления результатов научной (научно-технической, инновационной технологической) деятельности в виде отчетов, научных публикаций
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
	ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты

2. Показатели оценивания компетенций

В результате изучения дисциплины «Технологии программирования и операционные системы» обучающийся должен:

знать:

- основные принципы работы в Unix-системах;
- основные понятия и терминологию современных систем контроля версий, виртуализации, компьютерных сетей;
- основы программирования;
- технологии разработки ПО.

уметь:

- применять полученные знания для работы в командных проектах;
- работать с операционными системами;
- разрабатывать чистый поддерживаемый код.

владеть:

- инструментарием командной строки Unix;
- современными системами контроля версий и платформами виртуализации;
- методологиями разработки в проектной деятельности.

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

1. Чем отличаются процедурное, объектно-ориентированное и функциональное программирование? Приведите примеры языков программирования, поддерживающих каждый из этих подходов.
2. Что такое операционная система? Какие функции выполняет операционная система?
3. Что такое SOLID-принципы в объектно-ориентированном программировании? Объясните каждый из принципов.
4. Чем отличаются многозадачность, многопоточность и мультипрограммирование в операционных системах?

5. Что такое шаблоны проектирования (Design Patterns) и зачем они используются в разработке программного обеспечения?
6. Что такое планирование процессов в операционных системах? Какие алгоритмы планирования процессов вы знаете?
7. Что такое RESTful API и какие принципы лежат в его основе?
8. Что такое виртуальная память и как она реализуется в операционных системах?
9. В чем разница между синхронным и асинхронным программированием? Какие преимущества и недостатки у каждого из подходов?
10. Чем отличается однопользовательская операционная система от многопользовательской? Как обеспечивается безопасность данных в многопользовательских системах?

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

Список примерных вопросов:

1. Семейство ОС Unix, основные дистрибутивы и их отличия.
2. Основные командные интерпретаторы, их отличия.
3. Утилита sed, язык awk.
4. Использование Python как заменителя shell.
5. Jupyter и TmpNb как его модификация.
6. Терминальные мультиплексоры, автоматизация работы с ними.
7. Системы контроля версий в современных проектах.
8. CVS – первая система контроля версий.
9. SVN. Современные VCS – Git, Mercurial и работа с ними.
10. Работа над проектами в команде.
11. Автоматизация работы с Git.
12. Виртуализация. Её виды (контейнерная, на уровне ОС).
13. Современные платформы виртуализации и работа с ними.
14. Автоматизация работы с виртуальными окружениями.
15. Непрерывная интеграция (continuous integration) и её этапы.
16. Основные инструменты CI и их связь с VCS.
17. Компьютерные сети.
18. Модель OSI, её уровни.
19. Сетевые протоколы обмена информацией.

Критерии оценивания

Оценка "Отлично" (10) - полностью и вовремя решены все задачи без ошибок. Продemonстрирован грамотный подход к решению задач, реализованы оптимальные алгоритмы, код оформлен в едином удобочитаемом стиле.

Оценка "Отлично" (9) - полностью и вовремя решены все задачи без ошибок. Продemonстрирован грамотный подход к решению задач, реализованы оптимальные алгоритмы.

Оценка "Отлично" (8) - полностью и вовремя решены все задачи без ошибок. Продemonстрирован грамотный подход к решению задач.

Оценка "Хорошо" (7) - полностью решены все задачи. Допущены несущественные ошибки.

Оценка "Хорошо" (6) - полностью решено большинство задач. В некоторых задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

Оценка "Хорошо" (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.

Оценка "Удовлетворительно" (3) - полностью решено более половины задач.

Оценка "Неудовлетворительно" (2) - решено менее половины задач.

Оценка "Неудовлетворительно" (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.