

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

УТВЕРЖДЕНО
Заместитель директора

Ю.О. Соболев

	Рабочая программа дисциплины (модуля)
по дисциплине:	Язык Java
по направлению:	Информатика и вычислительная техника
профиль подготовки:	Разработка IT-продукта
	центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
	центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
курс:	2
квалификация:	магистр

Семестр, формы промежуточной аттестации: 3 (осенний) - Дифференцированный зачет

Аудиторных часов: 24 всего, в том числе:

лекции: 12 час.

семинары: 12 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 66 час.

Всего часов: 90, всего зач. ед.: 2

Программу составили:

О.А. Культепина, старший методист

К.А. Лапин, методист

Программа обсуждена на заседании центра дополнительного, дополнительного профессионального и
онлайн-образования "Пуск" 19.03.2024

Аннотация

Данный курс направлен на формирование навыков разработки приложений с использованием возможностей объектно-ориентированных языков программирования, на примере Java. В ходе обучения студенты получают знания о базовом синтаксисе, основных конструкциях языка Java, научиться работать с файлами и обрабатывать ошибки. Изучат основы объектно-ориентированного программирования, получают навыки разработки серверной части программного приложения на Java. В результате прохождения данного курса студенты получают все необходимые знания и навыки для разработки программного обеспечения на языке Java в качестве бэкенд разработчика.

1. Цели и задачи

Цель дисциплины

- изучение основ языка программирования Java, понимание объектно-ориентированной парадигмы программирования, формирование навыков разработки приложений с использованием возможностей объектно-ориентированных языков программирования.

Задачи дисциплины

- изучение базового синтаксиса языка;
- изучение основ объектно-ориентированного программирования, таких как инкапсуляция, наследование и полиморфизм;
- изучение принципов SOLID и паттернов проектирования;
- изучение основных библиотек и фреймворков Java, таких как Spring и JUnit, а также работу с API других систем;
- изучение вопросов тестирования и отладки приложений.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	УК-1.2 Осуществляет поиск вариантов решения поставленной проблемной ситуации на основе доступных источников информации
УК-6 Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки	УК-6.1 Умеет решать задачи собственного личностного и профессионального развития, определять и реализовывать приоритеты совершенствования собственной деятельности
ОПК-5 Способен и готов к профессиональному росту и руководству коллективом в области информатики и вычислительной техники, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия	ОПК-5.3 Стремится к получению новых знаний, профессиональному и личностному росту
ПК-4 Способен проектировать информационные системы, их компоненты и протоколы их взаимодействия, собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности; способен к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности	ПК-4.1 Знает основы проектирования и элементы архитектурных решений информационных систем
	ПК-4.2 Умеет применять в практической деятельности профессиональные стандарты в области информационных технологий
	ПК-4.3 Имеет опыт оценки качества, надежности и эффективности информационной системы в конкретной профессиональной сфере
ПК-6 Способен применять в профессиональной деятельности	ПК-6.1 Знает современные языки программирования и методы параллельной обработки данных

профессиональной деятельности
современные языки программирования и
программное обеспечение; операционные
системы и сетевые технологии, электронные
библиотеки и пакеты программ

ПК-6.2 Умеет реализовывать и применять численные
методы решения прикладных задач в профессиональной
сфере деятельности, используя пакеты программного
обеспечения, операционные системы, электронные
библиотеки, сетевые технологии

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- основы языка программирования Java, включая синтаксис, типы данных, операторы и выражения;
- принципы объектно-ориентированного программирования и понимание объектно-ориентированной парадигмы;
- как работать с API других систем и настраивать интеграцию приложения на Java с внешними сервисами.

уметь:

- применять объектно-ориентированную парадигму программирования при разработке приложений на Java, включая создание и использование классов, наследование, полиморфизм и инкапсуляцию;
- использовать основные конструкции языка Java, такие как условные операторы, циклы, массивы и коллекции;
- разрабатывать серверную часть программного приложения на Java, включая работу с сетевыми приложениями, создание сервлетов и фильтров, работу с базами данных через JDBC и применение лямбда-выражений;
- работать с библиотеками и фреймворками Java, включая Spring и JUnit, а также тестировать и отлаживать приложения.

владеть:

- навыками разработки приложений на Java, используя возможности объектно-ориентированных языков программирования, отвечающих требованиям бэкенд разработчиков.

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	Разработка на Java	2			14
2	Работа с основными конструкциями Java	2	2		14
3	Реализация объектно-ориентированного программирования на Java	2	2		14
4	Разработка серверной части программного приложения на Java	4	6		12
5	Работа с библиотеками и фреймворками Java	2	2		12
Итого часов		12	12		66
Подготовка к экзамену		0 час.			
Общая трудоёмкость		90 час., 2 зач.ед.			

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

Семестр: 3 (Осенний)

1. Разработка на Java

Введение в Java. Основные принципы объектно-ориентированного программирования.

Основные элементы языка Java: переменные, типы данных, операторы, выражения, классы и объекты.

2. Работа с основными конструкциями Java

Основные конструкции языка: условные операторы, циклы, массивы, коллекции. Обработка исключений. Работа с файлами и потоками.

3. Реализация объектно-ориентированного программирования на Java

Основы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм. Принципы SOLID. Паттерны проектирования.

4. Разработка серверной части программного приложения на Java

Разработка сетевых приложений. Создание сервлетов и фильтров. Работа с базой данных через JDBC. Применение лямбда-выражений в разработке приложений.

5. Работа с библиотеками и фреймворками Java

Основные библиотеки и фреймворки Java: Spring, JUnit. Работа с API других систем. Тестирование и отладка приложений.

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

Занятия по учебной дисциплине проводятся с применением дистанционных образовательных технологий. Каждый обучающийся обеспечен доступом к образовательной платформе (LMS).

6. Перечень рекомендуемой литературы

Основная литература

1. Java для Internet в Windows и Linux [Текст] / С. Б. Дунаев .— М. : ДИАЛОГ-МИФИ, 2004 .— 496 с.

Дополнительная литература

Пономарчук, Ю. В. Программирование на языке Java: учебное пособие / Ю. В. Пономарчук, И. В. Кузнецов. — Хабаровск: ДВГУПС, 2021. — 103 с. — Текст: электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/259451> — Режим доступа: для авториз. пользователей.

Хабитуев, Б. В. Программирование на языке Java: практикум: учебное пособие / Б. В. Хабитуев. — Улан-Удэ: БГУ, 2020. — 94 с. — ISBN 978-5-9793-1548-5. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/171791> — Режим доступа: для авториз. пользователей.

Федоричев, Л. А. Реализация многопоточности в языке Java / Л. А. Федоричев, О. В. Букунова. — Санкт-Петербург: Лань, 2024. — 72 с. — ISBN 978-5-507-48153-8. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/367400> — Режим доступа: для авториз. пользователей.

Строганкова, Н. В. Шаблоны программных платформ языка Java: учебное пособие / Н. В. Строганкова, К. В. Касьяненко, А. В. Хозяинов; составители практикума входят: использование. — Москва: РТУ МИРЭА, 2021. — 83 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/182466> — Режим доступа: для авториз. пользователей.

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)

Студенту для занятий потребуются:

1. Google Drive / Yandex disk для доступа к материалам курса
2. Zoom
3. LMS МФТИ
4. Приложение для коммуникации с преподавателями
5. Ноутбук для участия в интерактивных занятиях

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. В результате изучения дисциплины студент должен знать основные определения дисциплины, уметь применять полученные знания для решения прикладных задач.

Успешное освоение курса требует:

- посещения всех онлайн-занятий, предусмотренных учебным планом по дисциплине;
- ведения конспекта занятий;
- напряжённой самостоятельной работы студента.

Самостоятельная работа включает в себя:

- чтение материалов на платформе дистанционного обучения и рекомендованной литературы;
- проработку учебного материала, подготовку ответов на вопросы, предназначенных для самостоятельного изучения;
- решение задач, предлагаемых студентам на занятиях;
- подготовку к выполнению заданий текущей и промежуточной аттестации.

Показателем владения материалом служит умение без вспомогательных материалов и конспектов отвечать на вопросы по темам дисциплины.

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

Возможен промежуточный контроль знаний студентов в виде решения задач в соответствии с тематикой занятий.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Информатика и вычислительная техника
профиль подготовки:	Разработка IT-продукта центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск" центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
курс:	2
квалификация:	магистр

Семестр, формы промежуточной аттестации: 3 (осенний) - Дифференцированный зачет

Разработчики:

О.А. Культепина, старший методист

К.А. Лапин, методист

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	УК-1.2 Осуществляет поиск вариантов решения поставленной проблемной ситуации на основе доступных источников информации
УК-6 Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки	УК-6.1 Умеет решать задачи собственного личностного и профессионального развития, определять и реализовывать приоритеты совершенствования собственной деятельности
ОПК-5 Способен и готов к профессиональному росту и руководству коллективом в области информатики и вычислительной техники, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия	ОПК-5.3 Стремится к получению новых знаний, профессиональному и личностному росту
ПК-4 Способен проектировать информационные системы, их компоненты и протоколы их взаимодействия, собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности; способен к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности	ПК-4.1 Знает основы проектирования и элементы архитектурных решений информационных систем
	ПК-4.2 Умеет применять в практической деятельности профессиональные стандарты в области информационных технологий
	ПК-4.3 Имеет опыт оценки качества, надежности и эффективности информационной системы в конкретной профессиональной сфере
ПК-6 Способен применять в профессиональной деятельности современные языки программирования и программное обеспечение; операционные системы и сетевые технологии, электронные библиотеки и пакеты программ	ПК-6.1 Знает современные языки программирования и методы параллельной обработки данных
	ПК-6.2 Умеет реализовывать и применять численные методы решения прикладных задач в профессиональной сфере деятельности, используя пакеты программного обеспечения, операционные системы, электронные библиотеки, сетевые технологии

2. Показатели оценивания компетенций

В результате изучения дисциплины «Язык Java» обучающийся должен:

знать:

- основы языка программирования Java, включая синтаксис, типы данных, операторы и выражения;
- принципы объектно-ориентированного программирования и понимание объектно-ориентированной парадигмы;
- как работать с API других систем и настраивать интеграцию приложения на Java с внешними сервисами.

уметь:

- применять объектно-ориентированную парадигму программирования при разработке приложений на Java, включая создание и использование классов, наследование, полиморфизм и инкапсуляцию;
- использовать основные конструкции языка Java, такие как условные операторы, циклы, массивы и коллекции;
- разрабатывать серверную часть программного приложения на Java, включая работу с сетевыми приложениями, создание сервлетов и фильтров, работу с базами данных через JDBC и применение лямбда-выражений;
- работать с библиотеками и фреймворками Java, включая Spring и JUnit, а также тестировать и отлаживать приложения.

владеть:

- навыками разработки приложений на Java, используя возможности объектно-ориентированных языков программирования, отвечающих требованиям бэкенд разработчиков.

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

В начале занятия проводится опрос по материалам предыдущего занятия и даются задачи для совместного решения.

Задание с проверкой преподавателем

Цели:

- приобрести практический опыт разработки серверной части приложения на Java
- получить навык взаимодействия с базой данных через JDBC

Предполагаемый результат: разработанное веб-приложение, которое позволяет пользователям вести список задач и выполнять операции CRUD (создание, чтение, обновление, удаление) над задачами.

Описание задания:

1. Создать базу данных, содержащую таблицу "Задачи", с полями "ID", "Название", "Описание" и "Статус".
2. Разработать класс сервлета, который будет обрабатывать HTTP-запросы пользователя и выполнять соответствующие операции CRUD в базе данных.
3. Реализовать методы для создания, получения, обновления и удаления задач через JDBC.
4. Реализовать веб-интерфейс, который позволит пользователю просматривать список задач, добавлять новые задачи, редактировать существующие и удалять задачи.
5. Проверить работу приложения, отправляя HTTP-запросы и проверяя результаты операций CRUD.

Критерии оценивания:

корректность реализации основных методов создания, чтения, обновления и удаления задач через JDBC

работоспособность приложения, возможность добавления, редактирования и удаления задач

правильная обработка HTTP-запросов и взаимодействие с базой данных

качество кода, правильное использование объектно-ориентированных принципов и синтаксиса Java

возможность расширения функциональности приложения, добавление дополнительных операций или функций (например, фильтрация задач по статусу)

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

Промежуточный контроль проходит в форме дифференцированного зачета:

Типовые задания:

1. Выберите верное имя переменной согласно соглашениям по именованию переменных в Java:
 - a) myDog
 - b) myData
 - c) my_data
 - d) MyDog
2. Какой код скомпилируется без ошибок?
 - a) String name = "Petya";
 - b) String name = String "Petya";
 - c) int age = "18";
 - d) int age = Integer.parseInt("18");
3. Что означает public static void main(String[] args) в Java?
 - a) Объявление пакета для основной программы.

- b) Статический поле класса, содержащее код для выполнения.
 - c) Стартовый метод приложения, с которого начинается выполнение программы.
 - d) Импорт классов стандартной библиотеки Java.
4. Какой способ позволяет обратиться к статическому полю класса из другого пакета?
- a) Просто написать название поля, Java автоматически найдет класс с этим полем.
 - b) Импортировать класс и обратиться к полю через его имя и точку (например, Class.field).
 - c) Написать полный путь к полю: пакет.класс.поле.
 - d) Переместить класс, из которого доступ к полю осуществляется, в тот же пакет и обратиться к полю через его имя и точку.
5. Как объявить поле, которое будет иметь отдельное значение для каждого экземпляра объекта?
- a) Объявить поле с модификатором static.
 - b) Объявить поле без модификатора static.
 - c) Не объявлять поле, а обращаться к нему через имя объекта и точку.
 - d) Невозможно объявить такое поле, оно всегда будет иметь общее значение для всех экземпляров объекта.
6. Какие методы могут быть вызваны из статического метода?
- a) Только другие статические методы.
 - b) Только нестатические методы.
 - c) И статические, и нестатические методы.
 - d) Ни один метод нельзя вызвать из статического метода.
7. Как получить длину массива?
- a) arr[arr.length]
 - b) arr[arr.length - 1]
 - c) arr[arr.length + 1]
 - d) arr.length
8. Какой код приведет к компиляционной ошибке?
- a) (new String[5])[0] = "Hello";
 - b) (new String[5])[0] = new String[5];
 - c) (new String[5])[0] = 5;
 - d) (new String[5])[0] = null;
9. Какой оператор позволяет создать экземпляр класса в Java?
- a) new
 - b) create
 - c) instanceof
 - d) instance
10. В чем состоит полиморфизм в Java?
- a) Возможность одного объекта иметь несколько типов данных.
 - b) В наследовании одного класса от другого.
 - c) В переопределении методов в дочернем классе.
 - d) В использовании абстрактных классов и интерфейсов.

Критерии оценивания

Дифференцированный зачет по дисциплине проводится в форме выполнения тестирования.

Оценка отлично (10 баллов) - 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 баллов) - 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 баллов) - выставляется студенту, показавшему всесторонние, 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 баллов) - 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 баллов) - 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 баллов) - 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 балла) - 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 балла) - 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 балла) - 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 балл) - 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 проходит в письменном формате, на lms платформе.

Время проведения письменного дифференцированного зачета оставляет 2 академических часа.

Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины.