

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский Нижегородский государственный университет  
им. Н.И. Лобачевского»**

Институт информационных технологий, математики и механики

(факультет / институт / филиал)

УТВЕРЖДЕНО  
решением ученого совета ННГУ  
протокол от  
«30» ноября 2022 г. № 13

**Рабочая программа дисциплины**

Разработка сетевых приложений на Java

(наименование дисциплины (модуля))

Уровень высшего образования

Бакалавриат

(бакалавриат / магистратура / специалитет)

Направление подготовки / специальность

01.03.02 Прикладная математика и информатика

(указывается код и наименование направления подготовки / специальности)

Направленность образовательной программы

Прикладная математика и информатика (общий профиль)

(указывается профиль / магистерская программа / специализация)

Форма обучения

очная

(очная / очно-заочная / заочная)

Нижний Новгород

2023 год

## 1. Место дисциплины в структуре ООП

Дисциплина относится к части, формируемой участниками образовательных отношений.  
Б1.В.ДВ.11.03.

№ варианта	Место дисциплины в учебном плане образовательной программы	Стандартный текст для автоматического заполнения в конструкторе РПД
2	Блок 1. Дисциплины (модули) Часть, формируемая участниками образовательных отношений	Дисциплина Б1.В.ДВ.11.03 «Разработка сетевых приложений на Java» относится к части ООП направления подготовки 01.03.02 «Прикладная математика и информатика», формируемой участниками образовательных отношений.

## 2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	
ПК-5. Способен проектировать программное обеспечение	ПК-5.1. Знает типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения	Знать способы оценки количества информации и особенности их применения к количественной оценке разных видов информации	Собеседование
	ПК-5.2. Знает методы и средства проектирования программного обеспечения	Знать методы и средства проектирования программного обеспечения	Собеседование
	ПК-5.3.: Знает методы и средства проектирования баз данных	Знать методы и средства проектирования баз данных	Собеседование

	<i>ПК-5.4.: Умеет использовать существующие типовые решения и шаблоны проектирования программного обеспечения</i>	Уметь понимать содержательную сторону информационных процессов, знать способы передачи, приёма, обработки, анализа и хранения информации;	<i>Задания</i>
	<i>ПК-5.5.: Умеет применять методы и средства проектирования программного обеспечения, структур данных, баз данных</i>	Уметь профессионально разрабатывать и использовать программное обеспечение для поддержки информационных систем и процессов, владеть современными инструментальными вычислительными средствами;	<i>Задания</i>

### 3. Структура и содержание дисциплины

#### 3.1. Трудоемкость дисциплины

	Очная форма обучения
<b>Общая трудоемкость</b>	<b>2 ЗЕТ</b>
<b>Часов по учебному плану</b>	<b>72</b>
<b>в том числе</b>	
<b>аудиторные занятия (контактная работа):</b>	<b>37</b>
- занятия лекционного типа	24
- занятия семинарского типа	12
- занятия лабораторного типа	0
- текущий контроль (КСР)	1
<b>самостоятельная работа</b>	<b>35</b>
<b>Промежуточная аттестация – зачет</b>	

#### 3.2. Содержание дисциплины

Наименование и краткое содержание разделов и тем дисциплины	Всего (часы)	В том числе				
		Контактная работа (работа во взаимодействии с преподавателем), часы. Из них				Самостоятельная работа обучающегося, часы
		Занятия лекционного типа	Занятия семинарского типа	Занятия лабораторного типа	Всего	

Введение в Java	4	2	1		3	1
Синтаксис Java	4	2	1		3	1
Повторение основ ООП	6	2	1		3	3
Демонстрация выполненного учебного примера	6	2	2		4	2
Обзор постановок задач для самостоятельной работы	6	2	1		3	3
Библиотека классов Java	6	2	1		3	3
Разработка визуальных приложений	6	2	1		3	3
Разработка мобильных приложений	6	2	1		3	3
Java и элементы параллельного программирования	7	2	1		3	4
Организация взаимодействия с бинарным кодом	7	2	1		3	4
Разработка сетевых приложений	7	2	1		3	4
Организация взаимодействия с базами данных в Java	6	2	0		2	4
Текущий контроль (КСР)	1				1	
Промежуточная аттестация – зачет						
<b>Итого</b>	<b>72</b>	<b>24</b>	<b>12</b>		<b>37</b>	<b>35</b>

Текущий контроль успеваемости реализуется в формах опросов на занятиях семинарского типа.

Промежуточная аттестация проходит в традиционной форме (зачет).

#### **4. Учебно-методическое обеспечение самостоятельной работы обучающихся**

##### **Виды самостоятельной работы студентов**

В начале семестра группам студентов выдаются задания, используемые в практических заданиях и самостоятельной работе.

Пример учебной задачи для практикума:

##### ***Система взаимодействия группы людей***

*Разработать программу, взаимодействия пользователей по средствам отправки мгновенных сообщений. Должно быть предусмотрено два типа обмена сообщениями: обмен личными сообщениями между двумя пользователями и конференция между несколькими участниками. Программные клиенты должны предоставлять возможность общего доступа к документам. Для файлов в форматах “\*.txt, \*.csv, \*.jpg” реализовать возможность просмотра. Для файлов в форматах отличных от “\*.txt, \*.csv, \*.jpg” реализовать возможность загрузки на локальный диск.*

Приведенный пример используется для демонстрации основных средств и подходов используемы при разработке программ в самостоятельной работе.

В рамках курса студентам предлагается самостоятельно разработать программу по одному из заданий приведенных ниже. Для каждого задания необходимо реализовать консольный, визуальный и мобильный вариант программы. Все программы предполагают клиент-серверное взаимодействие.

Примеры заданий для самостоятельных работ:

### **Эмулятор сетей**

*Необходимо реализовать эмулятор сетей с устройствами первых трех уровней. Т.е. сетевые карты, коммутаторы и концентраторы, а так же роутеры. В эмуляторе у пользователя должна быть предусмотренная возможность конструировать топологию сети. С помощью полученной топологии пользователь должен суметь проверить корректность настройки сетевых устройств. В приложении, должна быть предусмотрена возможность топологию сети конструировать в нескольких приложениях связанных по сети.*

### **Удаленная виртуальная машина**

*Необходимо реализовать клиент серверное приложение. В клиенте для пользователя должна быть предусмотрена возможность вводить программу на псевдо языке программирования. На сервере программа, разработанная на клиенте, должна приниматься и исполняться. Результат исполнения программы должен возвращаться клиенту. Реализовать возможность совместного редактирования кода. Реализовать отладку кода (исполнение по шагам, исполнение до контрольной точки, вывод значений переменных).*

### **Удаленное управление и исполнение задач**

*Необходимо разработать систему, состоящую из двух частей. Первая, клиент, должна позволять загружать программы на сервер, обозревать и скачивать результирующие файлы. Вторая часть, сервер, должна уметь поддерживать деятельность клиента по работе с программами и результатами исполнения. Реализовать контроль версий запускаемых программ. На клиенте отображать гистограмму времен исполнения программ.*

### **Террариум**

*Необходимо разработать программу, которая позволила бы устраивать «соревнование» программ. Серверная часть системы должна предоставлять «арену» соревнования. Клиентская часть системы должна предоставлять набор функций, используя которые, разрабатываются «состязующиеся программы». Клиенты должны иметь возможность загружать коды «состязующихся программ» на сервер. Сервер должен визуально отображать состояние «арены».*

### **Удаленный, авторулевой лодок**

*Пусть есть набор лодок, которые плывут на определенной «арене». Лодки имеют текущее положение и место, куда бы хотели приплыть. Лодки управляются диспетчерскими программами. Диспетчерская программа является удаленной. Основная ее задача привести*

все подчиненные лодки в заданное положение и обеспечить их безопасность. Т.е. лодки не должны столкнуться, и врезаться в местность. Разработать серверную программу реализующую отображение и формирование «арены». Реализовать клиентскую программу управляющую лодками.

### **Реализовать многопользовательскую игру «Пакмен»**

Разработать многопользовательскую игру «Пакмен». Серверная часть программы должна включать отслеживание положения игроков и разрешать вопросы их взаимодействия. Клиентская часть программы должна обеспечивать возможность подключения к игре, отображение поля игры и давать возможность управлять своим «пакменом». Реализовать редактор уровней.

### **Распределенный визуальный редактор**

Разработать клиент-серверную программу, которая бы позволила пользователям одновременно создавать и редактировать изображения. В качестве базовых объектов используются точки, линии, прямоугольники, дуги и окружности. Из примитивных объектов реализовать возможность конструирования более сложных. Реализовать возможность простейшей анимации объектов.

### **Распределенный администратор сайта**

Разработать программу, в которой администраторы совместно могли бы редактировать html код страниц сайта. Администратор должен видеть, кто и какие страницы редактирует и в случае необходимости присоединяться к редактированию. Реализовать возможность блокировки изменений для части текста страницы, от открывающего до закрывающего тега. Клиент должен подкрашивать захваченный текст и отображать информацию об редакторе.

Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведены в п. 5.2.

## **5. Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю), включающий:**

### **5.1.Описание шкал оценивания результатов обучения по дисциплине**

Уровень сформированности компетенций (индикатора достижения компетенций)	Шкала оценивания сформированности компетенций						
	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	Не зачтено		Зачтено				
<u>Знания</u>	Отсутствие знаний	Уровень знаний ниже	Минимально допустимый	Уровень знаний в	Уровень знаний в	Уровень знаний в	Уровень знаний в

	теоретическо го материала.  Невозможнос ть оценить полноту знаний вследствие отказа обучающего от ответа	минималь- ных требований. Имели место грубые ошибки.	уровень знаний. Допущено много негрубых ошибки.	объеме, соответствующ ем программе подготовки. Допущено несколько негрубых ошибок	объеме, соответствующ ем программе подготовки. Допущено несколько несущественн ых ошибок	объеме, соответствующ ем программе подготовки, без ошибок.	объеме, превышающе м программу подготовки.
<u>Умения</u>	Отсутствие минималь- ных умений. Невозмож- ность оценить наличие умений вследствие отказа обучающего- от ответа	При решении стандартных задач не продемонстр ированы основные умения.  Имели место грубые ошибки.	Продемонстр ированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме.	Продемонстри рованы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продемонстри рованы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продемонстр ированы все основные умения, решены все основные задачи с отдельными несущест- венным недочетами, выполнены все задания в полном объеме.	Продемонстр ированы все основные умения, решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
<u>Навыки</u>	Отсутствие владения материалом. Невозможнос ть оценить наличие навыков вследствие отказа обучающего от ответа	При решении стандартных задач не продемонстр ированы базовые навыки.  Имели место грубые ошибки.	Имеется минимальны й набор навыков для решения стандартных задач с некоторыми недочетами.	Продемонст- рированы базовые навыки при решении стандартных задач с некоторыми недочетами	Продемонстри рованы базовые навыки при решении стандартных задач без ошибок и недочетов.	Продемонстр ированы навыки при решении нестандартн ых задач без ошибок и недочетов.	Продемонстр ирован творческий подход к решению нестандартн ых задач.

### Шкала оценки при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	Превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно»
	Отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично», при этом хотя бы одна компетенция сформирована на уровне «отлично»
	Очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо», при этом хотя бы одна компетенция сформирована на уровне «очень хорошо»
	Хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо», при этом хотя бы одна компетенция сформирована на уровне «хорошо»

	Удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
не зачтено	Неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно», ни одна из компетенций не сформирована на уровне «плохо»
	Плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

## 5.2. Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения

### 5.2.1 Контрольные вопросы для собеседования

вопросы	Код формируемой компетенции
1. Что такое включает в себя понятие виртуальная машина?	ПК-5
2. Чем отличается исполнение кода написанного на java и на c?	ПК-5
3. Расскажите основные элементы объектно-ориентированного программирования на примере языка программирования Java	ПК-5
4. Какие основные структуры данных реализованы в Java? Приведите примеры	ПК-5
5. Какие обертки существуют для работы с входным и выходным потоком данных?	ПК-5
6. Расскажите основные принципы построения сети.	ПК-5
7. Как в java можно организовать клиент-серверное взаимодействие?	ПК-5
8. Расскажите о способах построения визуальных приложений на основе AWT и Swing.	ПК-5
9. Что такое MIDlet? Приведите примеры	ПК-5
10. Чем отличается построение клиент-серверного приложения в Java ME от Java SE?	ПК-5
11. Расскажите о способах разработки мобильных приложений исполняемых под операционной системой Android	ПК-5
12. Какие основные отличия существуют между разработкой приложений для операционной системы Android и Windows?	ПК-5

### 5.2.2. Типовые задания/задачи для оценки сформированности компетенции ПК-5

#### Вариант 1. Эмулятор сетей

Необходимо реализовать эмулятор сетей с устройствами первых трех уровней. Т.е. сетевые карты, коммутаторы и концентраторы, а так же роутеры. В эмуляторе у пользователей должна быть предусмотрена возможность совместного редактирования и настройки виртуальной сети. С помощью полученной топологии пользователи должны иметь возможность проверить корректность настройки сетевых устройств и сбора статистики. Для проверки необходимо реализовать на сетевых картах консоль с командами ipconfig, trace и



ping. В качестве статистики необходимо выдавать количество пройденных сетевых устройств и длину пути.

### **Вариант 2. Удаленная виртуальная машина (разработка кода)**

Необходимо реализовать клиент серверное приложение. В клиенте для пользователя должна быть предусмотрена возможность вводить программу на псевдо языке программирования. Разработанная на клиенте программа должна приниматься и исполняться на сервере. На клиенте реализовать возможность совместного редактирования кода. При изменении кода одним из пользователей изменения должны видеть все пользователи. Реализовать возможность сохранения изменений в коде в виде патчей (<https://code.google.com/archive/p/java-diff-utils/>). Для каждого патча реализовать возможность обсуждения и отката.

### **Вариант 3. Удаленная виртуальная машина (исполнение кода)**

Необходимо реализовать клиент серверное приложение. В клиенте для пользователя должна быть предусмотрена возможность вводить программу на псевдо языке программирования. Разработанная на клиенте программа должна приниматься и исполняться на сервере. В один и тот же момент времени может исполняться  $N$  программ. На клиенте реализовать возможность отображения очереди задач и истории запусков. Пользователь должен иметь возможность по истории запусков восстановить код и результаты экспериментов. С каждым запуском должна быть реализована возможность обсуждения результатов.

## **6. Учебно-методическое и информационное обеспечение дисциплины**

а) основная литература:

1. Н.Вязовик. Программирование на Java - <http://www.intuit.ru/studies/courses/16/16/info>
2. Построение распределенных систем на Java - <http://www.intuit.ru/studies/courses/633/489/info>
3. Новиков Ю.В., Кондратенко С.В. Учебный курс «Основы локальных сетей». <http://www.intuit.ru/studies/courses/57/57/info>

---

б) дополнительная литература:

1. Чарльз Петзолд. Програмируемо Windows Phone 7. – Microsoft Press. [http://rusdpe.blob.core.windows.net/downloads/Programming\\_Windows\\_Phone\\_7\\_ru.pdf](http://rusdpe.blob.core.windows.net/downloads/Programming_Windows_Phone_7_ru.pdf)

---

в) программное обеспечение и Интернет-ресурсы

1. Виртуальная машина Java - <https://java.com/ru/download/>
2. Дистрибутивы средств разработки ПО - <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
3. Официальная документация по языку программирования Java - <https://docs.oracle.com/javase/tutorial/>

## **7. Материально-техническое обеспечение дисциплины**

Помещения представляют собой учебные аудитории для проведения учебных занятий,

предусмотренных программой, оснащенные оборудованием и техническими средствами обучения.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду ННГУ.

Программа составлена в соответствии с требованиями ОС ННГУ 01.03.02 Прикладная математика и информатика.

Авторы: к.т.н., доцент кафедры МОСТ Козинов Е.А.

к.т.н., доцент кафедры МОСТ Мееров И.Б.

Рецензент: д.т.н., профессор НГТУ им. Р.Е. Алексеева Ломакина Л.С.

Заведующий кафедрой МОСТ: д.ф.-м.н. Стронгин Р.Г.

Программа одобрена на заседании методической комиссии института информационных технологий, математики и механики

от 30 ноября 2022 года, протокол № 3.