

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)

ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ, ФИЗИКИ И ИНФОРМАТИКИ
(Наименование института)

УТВЕРЖДАЮ:

Директор института

Хорьков К.С.

« 30 » 08 2021 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

РАЗРАБОТКА КРОСС-ПЛАТФОРМЕННЫХ ПРИЛОЖЕНИЙ

(наименование дисциплины)

направление подготовки / специальность

01.03.02. «Прикладная математика и информатика»

(код и наименование направления подготовки (специальности))

направленность (профиль) подготовки

Математическое и компьютерное моделирование, программирование и системный анализ

(направленность (профиль) подготовки))

г. Владимир

2021 год

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью данного курса является изучение объектно-ориентированного языка программирования Java и основных приемов разработки кросс-платформенных приложений на платформе Java 2 Standart Edition (J2SE).

Задачи:

- изучение теоретической части (кросс-платформенный объектно-ориентированный язык программирования Java, основные пакеты платформы Java 2 Standart Edition)
- закрепление на практике полученных теоретических знаний (разработка клиент-серверных приложений на языке Java с многопоточной архитектурой и оконным пользовательским интерфейсы).
-

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «РАЗРАБОТКА КРОСС-ПЛАТФОРМЕННЫХ ПРИЛОЖЕНИЙ» относится к обязательным дисциплинам учебного плана.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций) БК

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	
ОПК-2. Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	ОПК-2.1. Знает математические основы, основные положения и концепции в области программирования. ОПК-2.1. Умеет осуществлять обоснованный выбор математических и компьютерных методов, а также необходимого программного обеспечения при решении задач профессиональной деятельности. ОПК-2.3. Владеет навыками применения математических и компьютерных методов и программного обеспечения при решении конкретных задач.	Знает: <input type="checkbox"/> расширенные знания в области математики; <input type="checkbox"/> математические основы, основные положения и концепции в области программирования; <input type="checkbox"/> архитектура языков программирования; <input type="checkbox"/> основная терминология в области программного обеспечения. Умеет: <input type="checkbox"/> умеет осуществлять обоснованный выбор математических и компьютерных методов, а также необходимого программного обеспечения при решении задач профессиональной деятельности. Владеет: <input type="checkbox"/> имеет навыки применения данных методов и программного обеспечения при решении конкретных задач.	<input type="checkbox"/> вопросы к текущему контролю знаний и промежуточной аттестации <input type="checkbox"/> отчеты по лабораторным работам <input type="checkbox"/> отчет по заданию в рамках самостоятельной работы в семестре
ОПК-5. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.	ОПК-5.1. Знает основные положения и концепции прикладного и системного программирования, архитектуры компьютеров и сетей. ОПК-5.2. Умеет разрабатывать алгоритмы и компьютерные программы с использованием современных языков и	Знает: <input type="checkbox"/> основные положения и концепции прикладного и системного программирования, архитектуры компьютеров и сетей; Умеет: <input type="checkbox"/> разрабатывать алгоритмы и компьютерные программы с использованием современных языков и инструментов программирования. Владеет: <input type="checkbox"/> технологиями создания и эксплуатации программных продуктов и программных комплексов	<input type="checkbox"/> вопросы к текущему контролю знаний и промежуточной аттестации <input type="checkbox"/> отчеты по лабораторным работам <input type="checkbox"/> отчет по заданию в рамках самостоятельного

	инструментов программирования. ОПК-5.3. Владеет технологиями создания и эксплуатации программных продуктов и программных комплексов.		й работы в семестре
ПК-4. Способен анализировать требования к программному средству.	ПК-4.1. Знает методы анализа и тестирования требований к программному средству, методы анализа, проектирования и разработки программного обеспечения. ПК-4.2. Умеет тестировать требования к программному средству, оформлять документацию по тестированию, анализировать требования на соответствие принятым стандартам и методам проектирования, использовать современные CASE-средства. ПК-4.3. Владеет навыками проверки осуществимости функционирования и сопровождения программного средства, определения возможности введения изменений и дополнений требований к программному средству.	Знает: <input type="checkbox"/> методы анализа и тестирования требований к программному средству; <input type="checkbox"/> методы анализа, проектирования и разработки программного обеспечения. Умеет: <input type="checkbox"/> требования к программному средству; <input type="checkbox"/> оформлять документацию по тестированию; <input type="checkbox"/> анализировать требования на соответствие принятым стандартам и методам проектирования; <input type="checkbox"/> использовать современные CASE-средства. Владеет: <input type="checkbox"/> навыки выявления противоречий в требованиях одинакового и различных уровней к программному средству; <input type="checkbox"/> навыки проверки осуществимости функционирования и сопровождения программного средства; <input type="checkbox"/> навыки определения возможности введения изменений и дополнений требований к программному средству.	<input type="checkbox"/> вопросы к текущему контролю знаний и промежуточной аттестации <input type="checkbox"/> отчеты по лабораторным работам <input type="checkbox"/> отчет по заданию в рамках самостоятельной работы в семестре
ПК-1. Способен проектировать и реализовывать программное обеспечение в соответствии с требованиями	ПК-1.1. Знает методологии разработки программного обеспечения и технологии программирования, языки формализации функциональных спецификаций, принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения, основные концепции и атрибуты качества программного обеспечения. ПК-1.2. Умеет проводить анализ исполнения	Знает: <input type="checkbox"/> возможности современных и перспективных средств разработки программных продуктов, технических средств; <input type="checkbox"/> методологии разработки программного обеспечения и технологии программирования; <input type="checkbox"/> методологии и технологии проектирования и использования баз данных; <input type="checkbox"/> методы и средства проектирования программного обеспечения и баз данных; <input type="checkbox"/> языки формализации функциональных спецификаций; <input type="checkbox"/> принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; <input type="checkbox"/> типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке	<input type="checkbox"/> вопросы к текущему контролю знаний и промежуточной аттестации <input type="checkbox"/> отчеты по лабораторным работам <input type="checkbox"/> отчет по заданию в рамках самостоятельной работы в семестре

	<p>требований, выработать варианты их реализации, проводить оценку и обоснование рекомендуемых решений, выбирать и использовать средства и варианты реализации программного обеспечения.</p> <p>ПК-1.3. Владеть навыками оценки возможностей, времени и трудоемкости реализации требований к программному обеспечению, разработки и согласования технических спецификаций на программное обеспечение, формирования и предоставления отчетности в соответствии с установленными регламентами, проектирования структур данных, баз данных, программных интерфейсов.</p>	<p>программного обеспечения;</p> <ul style="list-style-type: none"> <input type="checkbox"/> основные концепции и атрибуты качества программного обеспечения; <input type="checkbox"/> механизмы авторизации и аутентификации; <input type="checkbox"/> стили написания кода. <p>Умеет:</p> <ul style="list-style-type: none"> <input type="checkbox"/> проводить анализ исполнения требований; <input type="checkbox"/> выработать варианты реализации требований; <input type="checkbox"/> проводить оценку и обоснование рекомендуемых решений; <input type="checkbox"/> выбирать средства и варианты реализации программного обеспечения; <input type="checkbox"/> использовать существующие типовые решения и шаблоны проектирования программного обеспечения; <input type="checkbox"/> применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов. <p>Владеет:</p> <ul style="list-style-type: none"> <input type="checkbox"/> навыки оценки возможностей, времени и трудоемкости реализации требований к программному обеспечению; <input type="checkbox"/> навыки согласования требований к программному обеспечению с заинтересованными сторонами; <input type="checkbox"/> навыки оценки и согласования сроков выполнения поставленных задач; <input type="checkbox"/> навыки разработки и согласования технических спецификаций на программное обеспечение; <input type="checkbox"/> навыки формирования и предоставления отчетности в соответствии с установленными регламентами; <input type="checkbox"/> навыки проектирования структур данных, баз данных, программных интерфейсов; <input type="checkbox"/> навыки выбора стиля написания кода. 	
--	---	--	--

4. ОБЪЕМ И СТРУКТУРА ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 5 зачетных единиц, 180 часов.

Тематический план форма обучения – очная

№ п/п	Наименование тем и/или разделов/тем дисциплины	Семестр	Неделя семестра	Контактная работа обучающихся с педагогическим работником				Самостоятельная работа	Формы текущего контроля успеваемости, форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные работы	в форме практической подготовки		
1	Раздел 1 Введение в дисциплину. Основы языка Java.	5	1-8	16	-	16	16	-	Рейтинг-контроль №1
2	Раздел 2. Основные пакеты платформы J2SE.	5	9-13	10	-	10	10	-	
3	Раздел 3. Разработка клиент-серверных приложений с оконным интерфейсом	5	14-17	8	-	8	89	81	Рейтинг-контроль №2

4	Подведение итогов	5	18	2	-	2	-	-	Рейтинг-контроль №3
Всего за 5 семестр		-	-	36	-	36	-	81	Экзамен (27ч)
Наличие в дисциплине КП/КР		-	-	-	-	-	-	-	
Итого по дисциплине		5	18	36	-	36	117	81	Экзамен (27ч)

Содержание лекционных занятий по дисциплине

Раздел 1 Введение в дисциплину. Основы языка Java

Тема 1 Введение в дисциплину

Обзор программного обеспечения, необходимого для обучения. Понятие кроссплатформенности. Краткий обзор платформы J2SE. Структура Java-приложения. Пакеты. Понятие модуля компиляции в Java.

Тема 2 Введение в язык программирования Java

Лексика. Имена и идентификаторы. Соглашение по именованию. Типы данных. Операции. Операторы.

Тема 3 Объектная модель Java

Классы и объекты. Модификаторы видимости. Конструкторы. Статические элементы класса. Инициализаторы. Наследование. Абстрактные классы. Интерфейсы. Механизм позднего связывания и полиморфизм. Класс java.lang.Object. Класс java.lang.Class.

Тема 4 Приведение типов

Приведение примитивных и ссылочных типов. Запрещенные приведения.

Тема 5 Массивы

Массивы примитивных и ссылочных типов: объявление, способы инициализаций, особенности работы.

Тема 6 Исключения

Понятие исключительной ситуации (ИС). Причины возникновения и классификация ИС. Семейство базовых классов ИС. Пользовательские классы исключений. Порождение ИС (оператор throw). Обработка ИС (конструкции try-catch, try-catch-finally и try-finally). Оператор throws.

Раздел 2 Основные пакеты платформы J2SE.

Тема 1 Пакет java.util

Java-коллекции. Основные семейства классов Java-коллекций, их особенности. Generics. Классы java.util.HashMap, java.util.HashSet, java.util.ArrayList, java.util.LinkedList, java.util.Properties.

Тема 2 Пакет java.io

Потоки данных (stream). Семейство базовых классов потокового ввода/вывода. Классы java.io.InputStream, java.io.OutputStream, java.io.ByteArrayInputStream, java.io.ByteArrayOutputStream, java.io.DataInputStream, java.io.DataOutputStream. Сериализация. Классы java.io.Reader и java.io.Writer. Класс java.io.File.

Тема 3 Пакет java.awt.

Дерево компонент. Принципы отрисовки. Модель сообщений. Менеджеры компоновки. Основные принципы организации оконного пользовательского интерфейса.

Тема 4 Пакет java.lang.

Потоки выполнения (thread). Класс java.lang.Thread. Интерфейс java.lang.Runnable. Демон-потоки. Синхронизация. Методы wait(), notify(), notifyAll() класса Object.

Тема 5 Пакет java.net.

Сетевые протоколы. Классы java.net.InetAddress, java.net.Socket и java.net.ServerSocket.

Раздел 3 Разработка клиент-серверных приложений с оконным интерфейсом

Тема 1 Разработка приложений с оконным интерфейсом

Архитектурный шаблон проектирования Model-View-Controller (MVC). Концепция MVC. Применение шаблона проектирования MVC на примере тестовой задачи.

Тема 2 Разработка клиент-серверных приложений

Принципы организации клиент-серверного взаимодействия в приложениях. Разделение функционала приложения на клиентскую и серверную части. Многопоточная архитектура в клиент-серверных приложениях.

Содержание лабораторных занятий по дисциплине

Раздел 1 Введение в дисциплину. Основы языка Java

Тема 1 Введение в дисциплину.

Тема 2 Введение в язык программирования Java

Тема 3 Объектная модель Java

Тема 4 Приведение типов

Тема 5 Массивы

Лабораторная работа № 1. Разработка объектной модели задачи по любой выбранной предметной области. Создание объектов. Демонстрация реализованного в объектной модели функционала.

Тема 6 Исключения

Лабораторная работа № 2. Использование базовых классов исключений. Разработка пользовательских классов исключений. Порождение и обработка исключений.

Раздел 2 Основные пакеты платформы J2SE.

Тема 1 Пакет java.util

Лабораторная работа № 3. Использование Java-коллекций Set, Map и List.

Тема 2 Пакет java.io

Лабораторная работа № 4. Реализация потокового ввода\вывода данных с использованием классов java.io.InputStream, java.io.OutputStream, java.io.ByteArrayInputStream, java.io.ByteArrayOutputStream, java.io.DataInputStream, java.io.DataOutputStream. Реализация записи и чтения объектов в поток\ из потока через сериализацию\ десериализацию.

Тема 3 Пакет java.awt

Лабораторная работа № 5. Разработка оконного пользовательского интерфейса: отрисовка интерфейса и реализация функциональной части интерфейса (обработка событий элементов формы).

Тема 4 Пакет java.lang

Лабораторная работа № 6. Реализация многопоточности через наследование от класса java.lang.Thread и через реализацию интерфейса java.lang.Runnable..

Тема 5 Пакет java.net

Лабораторная работа № 7. Реализация клиент-серверного взаимодействия: разработка серверной части, разработка клиенткой части, реализация передачи данных между клиентской и серверной частями.

Раздел 3 Разработка клиент-серверных приложений с оконным интерфейсом

Тема 1 Разработка приложений с оконным интерфейсом

Тема 2 Разработка клиент-серверных приложений

Лабораторная работа № 8. Разработка клиент-серверного приложения с оконным интерфейсом. Применение шаблона проектирования MVC. Реализация многопоточной архитектуры в серверной и клиентской частях приложения.

5. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

5.1. Текущий контроль успеваемости

Рейтинг-контроля №1

Примерные вопросы:

Раздел №1

1. Понятие кроссплатформенного программного обеспечения
2. Платформа J2SE: основные пакеты
3. Из какой директории необходимо запускать компилятор, чтобы скомпилировать Java-приложение, состоящее из одного класса: test.first.Start, описание которого сохранено в файле Start.java по маршруту - c:\Java\programs\test\first\?
4. Из какой директории необходимо запускать интерпретатор Java для выполнения приложения, состоящее из одного класса: test.first.Start, описание которого сохранено в файле Start.java по маршруту - c:\Java\programs\test\first\?
5. Выберите верное утверждение - в Java определены следующие лексемы:
 - 1) идентификаторы, ключевые слова, литералы, разделители, операторы
 - 2) ключевые слова, литералы, разделители
 - 3) ключевые слова, литералы, операторы
 - 4) ключевые слова, литералы
 - 5) идентификаторы, ключевые слова, операторы
 - 6) нет верных вариантов
6. Перечислите все виды литералов в Java
7. Какие из перечисленных идентификаторов являются корректными?
 - 1) abc
 - 2) 1ab
 - 3) _bc
 - 4) _1c \$ac
 - 5) \$ac
 - 6) for_
 - 7) Int
 - 8) Byte
8. Что такое комментарий разработчика и для чего он служит?
9. Являются ли следующие слова ключевыми:
 - 1) true
 - 2) goto
 - 3) null
 - 4) const
 - 5) false
10. Чему будет равно следующее выражение и значения переменных x и y после вычислений?

```
int x=0, y=0;
print((++x==1) || (y++==1));
```
11. Что будет в результате работы программного кода:

```
byte b=3;
int c=b;
c++;
print(++b==c);
```
12. Что будет в результате выполнения выражений:
 - 1) 1/2
 - 2) 1./2
 - 3) 1/2.
 - 4) 1./2.
13. Выберите верные утверждения:
 - 1) 123 является целочисленным литералом
 - 2) -5 является целочисленным литералом
 - 3) целочисленные литералы имеют тип int по умолчанию
 - 4) инициализация переменных числовых типов литералами, выходящими за допустимый

диапазон значений, даст ошибку времени выполнения

5) инициализация переменных числовых типов литералами, выходящими за допустимый диапазон значений, на этапе выполнения будет выполнена с потерей точности (произойдет отбрасывание старших бит, искажение знакового разряда, округление до последнего допустимого разряда и т.п.)

6) нет верных утверждений

14. Проанализировать программный код на предмет синтаксической и семантической корректности - даст ли код ошибки компиляции и ошибки времени выполнения. Если код корректен, то что будет выведено на экран. Ответ пояснить

- 1) float a = 1.0f / 0.0f;
System.out.println("a = " + a);
- 2) float b = 0.0f / 0.0f;
System.out.println("b = " + b);
- 3) float c = (1.0f / 0.0f) * 0.0f;
System.out.println("c = " + c);
- 4) float d = 1.0f / +0.0f;
System.out.println("d = " + d);
- 5) float e = 5.5;
System.out.println("e = " + e);
- 6) boolean f = 1;
System.out.println("f = " + f);

15. Корректно ли следующее объявление с точки зрения формального выполнения соглашений по именованию:

```
public class flat{  
private int floor_number;  
private int r; // количество комнат  
public int rooms() {  
return r;  
}  
public int GetFloorNumber() {  
return floor_number;  
}  
}
```

16. Могут ли пакет и вложенный пакеты содержать одноименные классы?

17. Объявление класса: сигнатура и тело класса

18. Полная сигнатура методов класса

19. Если метод использует поле класса, должно ли оно быть объявлено выше объявления метода?

20. Что такое конструктор? Какие бывают конструкторы?

21. Может ли класс не иметь ни одного конструктора?

22. Что такое this?

23. Что такое super?

24. Можно ли при наследовании не реализовывать абстрактный метод родительского класса?

25. Если есть ссылка на абстрактный класс, можно ли с ее помощью обращаться к абстрактным методам этого класса?

26. Возможно ли не реализовывать все методы из интерфейса, указанного в выражении implements?

27. Какие модификаторы позволяют обращаться к элементу из классов того же пакета?

28. Какие модификаторы элементов интерфейса подставляются по умолчанию, а потому не рекомендованы для явного указания?

29. Выберите верные утверждения:

- 1) у пакетов нет модификаторов доступа, любой пакет может использоваться из любой точки программы
- 2) объявление локального класса с явным указанием модификатора доступа вызовет ошибку компиляции
- 3) все элементы интерфейсов являются public.
- 4) конструкторы не наследуются, поэтому не могут иметь модификатор доступа protected
- 5) прямое обращение к закрытому элементу класса из методов других классов вызовет ошибку компиляции
- 6) прямое обращение к закрытому элементу класса из методов других классов не вызовет ошибку компиляции, но даст исключение SecurityException на этапе выполнения
- 7) нет верных утверждений

30. При объявлении классов и интерфейсов существует возможность указать:

- 1) модификатор public или не указывать его (уровень доступа назначается по умолчанию)
- 2) модификаторы public, protected или не указывать никакой (уровень доступа назначается по умолчанию)
- 3) модификатор public, protected, private или не указывать никакой (уровень доступа назначается по умолчанию)
- 4) классы и интерфейсы не имеют модификаторов доступа, любой класс или интерфейс может использоваться из любой точки программы.
- 5) нет верных утверждений

31. Для каких элементов класса работает полиморфизм?

32. Какое значение появится на консоли после выполнения следующего программного кода:

```
public class Parent {
    int x = 2;
    public void print() {
        System.out.println(x);
    }
}

public class Child extends Parent {
    int x = 3;
    public static void main(String s[]) {
        new Child().print();
    }
}
```

33. Эквивалентны ли две следующие операции над ссылочными переменными x1 и x2, если SomeClass2 – это тип переменной x2:

```
x1 instanceof SomeClass2
x1.getClass().getName().equals(x2.getClass().getName());
```

34. При каком значении ссылочной переменной x следующее выражение всегда будет возвращать истину:

```
x.getClass() == x;
```

35. Как реализованы в классе Object методы equals(), toString()?

36. При переопределении метода equals(), будет ли требоваться переопределение каких-либо других методов, для корректного использования объектов рассматриваемого класса в хэш-таблицах?

37. От какого класса наследуются классы массивов?

38. Какая размерность у следующих массивов:

- 1) int x[], y[][];
- 2) String s, s1[], s2={{}, {"a", "b"}, null};

39. Можно ли использовать инициализатор в выражении выделения памяти под элементы массива? Если да, то дополнить программный код инициализатором:

```
int x[][] = new int[2][3];
```

40. Сколько объектов порождается при инициализации массивов:

- 1) `int a[][] = new int[3][4];`
- 2) `int b[][] = new int[3][][];`

41. Что будет в результате выполнения следующего программного кода:

```
Point p2[][] = (Point[][])p1.clone();  
p2[0] = new Point[]{new Point(2, 2)};  
System.out.println(p1[0][0]);
```

42. Как осуществляются сужающие и расширяющие приведения на целочисленной группе типов?

43. Как осуществляются сужающие и расширяющие приведения на вещественной группе типов?

44. Произойдет ли потеря точности при следующем преобразовании:

```
float f = -16777217;
```

Ответ пояснить.

45. Проанализировать код на предмет синтаксической корректности. Если код корректен, то какие значения примут все переменные, если нет, то какие ошибки будут сгенерированы:

- 1) `byte a = 100 - 100;`
- 2) `byte b = 100 + 100;`
- 3) `byte c = 100 * 100;`

46. Верны ли следующие выражения для переменной `d` вещественного типа (`float` или `double`):

- 1) `(short)d==(short)(int)d;`
- 2) `(int)d==(int)(long)d;`

47. Корректны ли следующие преобразования:

- 1) `Object o = (String)null;`
- 2) `String s = o;`

48. Пусть классы `Wolf` и `Rabbit` являются наследниками класса `Animal`. Корректен ли следующий программный код:

```
Wolf w = new Wolf();  
Animal a = (Animal)w;  
Rabbit r = (Rabbit)a;
```

Рейтинг-контроль №2

Примерные вопросы:

Раздел №2

49. Перечислите основные интерфейсы, используемые при работе с коллекциями?

50. Реализация какого интерфейса позволяет сравнивать экземпляры класса друг с другом и сортировать их, например, в коллекциях?

51. Выберите верные утверждения:

- 1) в коллекциях содержатся ссылки на объекты, а не их копии
- 2) классы-коллекции, реализующие интерфейс `List`, поддерживают порядок элементов
- 3) классы-коллекции, реализующие интерфейс `Set`, не допускают дублирования

элементов

4) классы-коллекции, реализующие интерфейс `Map`, используют уникальные ключи для размещения и поиска элементов

5) в коллекциях могут сохраняться ссылки как на объекты, так и на примитивные типы

6) ссылки на объекты в коллекциях `Vector` хранятся в порядке их добавления.

7) в качестве ключей для коллекций типа `Hashtable` должны передаваться только

объекты типа `String`

8) нет верных утверждений

52. Какой из перечисленных ниже интерфейсов реализует класс `HashTable`:

- 1) `SortedMap`
- 2) `Map`

- 3) List
- 4) SortedSet
- 5) нет верных утверждений

53. Причины возникновения синхронных и асинхронных исключений?

54. Приведите пример программного кода порождения и обработки исключительных ситуаций с помощью try-catch-finally конструкции, операторов throw и throws. Пример пояснить

55. Что будет напечатано в результате выполнения программного кода:

```
public class Test {
    float fVal = 0.0f;
    public Test() {
    }
    public static void main(String[] args) {
        Test t = new Test();
        String testVal = "0.123";
        System.out.println("Was returned " + t.testParse(testVal) + "
with value " + t.fVal);
    }
    private boolean testParse(String val){
        try {
            fVal = Float.parseFloat(val);
            return true;
        } catch (NumberFormatException ex) {
            System.out.println("Test.testParse() Bad number -> " + val);
            fVal = Float.NaN;
        } finally{
            System.out.println("Finally part executed");
        }
        return false;
    }
}
```

56. Какие источники могут быть использованы классами стандартных входных потоков Java в качестве источника данных?

57. От какого класса наследуются InputStream и OutputStream? Остальные классы потоков ввода/вывода?

58. Что произойдет, если, используя объект типа ObjectOutputStream, записать в файл данные в порядке long, int, byte, а считать, используя объект типа DataInputStream, - в порядке byte, int, long?

59. Какая кодировка используется классом OutputStreamWriter по умолчанию?

60. Какие методы объявлены в интерфейсе Serializable?

61. Какие из перечисленных ниже характеристик относятся к протоколу TCP:

- 1) образование постоянного соединения
- 2) не гарантирует доставку сообщения
- 3) протокол уровня приложения
- 4) использование в важных сервисах

62. Какие утилиты используются для тестирования работоспособности сети в ОС Windows?

63. Какие классы используются для работы с TCP-протоколом?

64. Можно ли с помощью класса URL пересылать данные на сервер?

65. Для чего служит в Java класс Thread:

- 1) для запуска потоков;
- 2) для остановки потоков;

- 3) для синхронизации потоков;
- 4) для изменения свойств потоков (н-р, приоритетов)
66. Каким образом на однопроцессорной машине исполняются многопоточные приложения?
67. Какие преимущества дает многопоточная архитектура?
68. Какое преимущество дает использование интерфейса Runnable перед прямым наследованием класса Thread?
69. Как получить объект класса Color, описывающий чистый синий цвет:
 - 1) new Color ("blue");
 - 2) new Color (0,0,255);
 - 3) Color.getBlue();
 - 4) Color.blue
70. В чем разница между компонентами List и Choice?
71. Какой метод нужно переопределить, чтобы реализовать отрисовку внешнего вида компонента?
72. Какими параметрами в Java характеризуется шрифт?
73. Перечислите основные события AWT
74. Какой интерфейс и какой метод нужно реализовать, чтобы обработать событие нажатия кнопки.

Рейтинг-контроль №3

Лабораторные работы № 1-8. Контрольными мероприятиями для оценивания являются отчеты по выполненным лабораторным работам.

5.2. Промежуточная аттестация по итогам освоения дисциплины проводится в форме экзамена.

Примерные вопросы к экзамену:

1. Платформа J2SE: основные пакеты
2. Лексемы, операторы, операции
3. Базовые примитивные типы данных
4. Базовые ссылочные типы данных
5. Приведение примитивных типов: расширяющие, сужающие, тождественные
6. Приведение ссылочных типов
7. Запрещенные приведения типов
8. Массивы примитивных и ссылочных типов
9. Классы и объекты, конструкторы, статические элементы класса, модификаторы видимости
10. Наследование, абстрактные классы, интерфейсы
11. Механизм позднего связывания и полиморфизм
12. Классы java.lang.Object и java.lang.Class: назначение, основные методы
13. Понятие исключительной ситуации, причины возникновения, классификация, обработка
14. Пользовательские классы исключений
15. Потоки данных, сериализация
16. Работа с сетью, основные классы пакета java.net
17. Потоки выполнения, класс java.lang.Thread, интерфейс java.lang.Runnable;
18. Многопоточная архитектура в клиент-серверных приложениях
19. Архитектурный шаблон проектирования MVC

5.3. Самостоятельная работа обучающегося.

На самостоятельное изучение выносятся изучение специальных графических библиотек для разработки оконных приложений:

- java.awt
- java.swt
- Swing

□ JavaFX

Так же в рамках самостоятельной работы в семестре студентам предлагается самостоятельно разработать полноценное приложение в соответствии с индивидуальным заданием.

Требования к выполнению

1. Проектирование и реализация объектной модели
 2. Реализация серверной и клиентской части: данные размещаются на сервере, вся бизнес-логика выполняется так же на сервере, клиентское приложение через пользовательский интерфейс отображает данные объектной модели, принимает запросы пользователя и отправляет эти запросы на сервер для обработки
 3. В качестве хранилища данных - файловая система (все данные сохраняются в файловой системе)
 4. Задача должна быть реализована с соблюдением основных принципов ООП (инкапсуляция, наследование, полиморфизм) и строго в концепции Model-View-Controller (отделение на уровне программного кода представления от бизнес-логики)
- Задачи могут выполняться как индивидуально, так и в небольших группах по 2-3

человека.

Список задач

1. File Server. Сервер предоставляет свою файловую систему. Клиент может просматривать файловую систему сервера и обмениваться с сервером файлами (скачивать и закачивать файлы с сервера \ на сервер). Задача не предполагает регистрацию клиента. Авторизация клиента и определение раздела файловой системы сервера, открытого для клиента, определяется ip-адресом клиента (например, если ip-адрес клиента лежит в диапазоне 100.100.50.*, то клиенту доступен раздел ФС сервера: - D:\documents).
1-2 человека
2. File Server. Сервер предоставляет клиентам информационные ресурсы для скачивания (фильмы, фотографии, музыку и проч). Для получения доступа к ресурсам клиенту необходимо завести на сервере свой аккаунт и получить статус – простой пользователь, vip-пользователь. От статуса будет зависеть скорость скачивания. Каждая сессия клиента начинается с процедуры авторизации (ввод логина\пароля).
2-3 человека
3. Клиент-серверный чат
1-2 человека
4. Сервер предоставляет клиенту HTML-страницы. Клиент скачивает с сервера эти страницы, находит по HTML-странице все ссылки, скачивает все связанные страницы и закачивает все найденные ресурсы на сервер.
1-2 человека
5. Сервер рассылает сигналы точного времени, клиент синхронизирует свое время.
1 человек
6. Мониторинг рабочего времени персонала. Сервер осуществляет мониторинг рабочего времени сотрудников: фиксирует время прихода\ухода на рабочее место; прихода\ухода на обеденный перерыв; а также периодически в основное рабочее время посылает контрольный вопрос сотруднику, для фиксации его фактического присутствия на рабочем месте. Результаты мониторинга для каждого сотрудника записываются в отдельный файл
1-2 человека
7. Система учета студентов (картотека).
1-2 человека
8. Сапер
1-2 человека
9. Пятнашки

1-2 человека

10. Тетрис

1-2 человека

11. Морской бой: клиент с клиентом. Сервер выполняет функцию авторизации пользователей, предоставляет список возможных соперников, организует обмен данными между игроками в процессе игры, осуществляет мониторинг игры и сохранение состояния игры в случае ее досрочного завершения.

1-2 человека

12. Морской бой: клиент с сервером. Сервер выполняет функцию авторизации пользователей, реализует логику соперника и сохранение состояния игры в случае ее досрочного завершения.

2-3 человека

13. Крестики-нолики: клиент с клиентом. Сервер выполняет функцию авторизации пользователей, предоставляет список возможных соперников, организует обмен данными между игроками в процессе игры, осуществляет мониторинг игры и сохранение состояния игры в случае ее досрочного завершения.

1-2 человек

14. Крестики-нолики: клиент с сервером. Сервер выполняет функцию авторизации пользователей, реализует логику соперника и сохранение состояния игры в случае ее досрочного завершения.

2-3 человека

15. Томагочи \ Виртуальный Зоопарк \ Виртуальный заповедник

1 человек

Для выполнения заданий в рамках самостоятельной работы в семестре следует использовать лекционный материал и источники, указанные в разделе 6 рабочей программы.

Контрольными мероприятиями для оценки выполнения студентом самостоятельной работы являются отчеты по выполненным заданиями

Фонд оценочных материалов (ФОМ) для проведения аттестации уровня сформированности компетенций обучающихся по дисциплине оформляется отдельным документом.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Книгообеспеченность

Наименование литературы: автор, название, вид издания, издательство	Год издания	КНИГООБЕСПЕЧЕННОСТЬ
		Наличие в электронном каталоге ЭБС
Основная литература		
1. Вязовик Н.А. Программирование на Java [Электронный ресурс]/ Вязовик Н.А.— Электрон. текстовые данные.— Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 603 с	2021	Режим доступа: http://www.iprbookshop.ru/73710.html .— ЭБС «IPRbooks».
2. Мухаметзянов Р.Р. Основы программирования на Java [Электронный ресурс]: учебное пособие/ Мухаметзянов Р.Р.— Электрон. текстовые данные.— Набережные Челны: Набережночелнинский государственный педагогический университет, 2017.— 114 с	2017	Режим доступа: http://www.iprbookshop.ru/66812.html .— ЭБС «IPRbooks»
3. Леднева С.Ю. Программирование на Java. Ч.1 [Электронный ресурс]: лабораторный практикум/ Леднева С.Ю., Ефимушкина Н.В., Цыганов А.А.— Электрон. текстовые данные.— Самара: Самарский государственный технический университет, ЭБС АСВ, 2015.— 159 с	2015	Режим доступа: http://www.iprbookshop.ru/90875.html .— ЭБС «IPRbooks»
Дополнительная литература		
1. Гуськова О.И. Объектно ориентированное программирование в Java [Электронный ресурс]: учебное пособие/ Гуськова О.И.— Электрон. текстовые данные.— Москва: Московский педагогический государственный университет, 2018.— 240 с	2018	Режим доступа: http://www.iprbookshop.ru/97750.html .— ЭБС «IPRbooks»

2. Джошуа Блох Java. Эффективное программирование [Электронный ресурс]/ Джошуа Блох— Электрон. текстовые данные.— Саратов: Профобразование, 2017.— 310 с	2019	Режим доступа: http://www.iprbookshop.ru/64057.html .— ЭБС «IPRbooks»
3. Программирование на языке Java [Электронный ресурс]: конспект лекций/ А.В. Гаврилов [и др.].— Электрон. текстовые данные.— Санкт-Петербург: Университет ИТМО, 2015.— 123 с	2015	Режим доступа: http://www.iprbookshop.ru/68692.html .— ЭБС «IPRbooks»

6.2. Периодические издания

1. Журнал РАН «Программирование» // Режим доступа: <https://www.ispras.ru/programming/>
2. Журнал «Прикладная информатика» // Режим доступа: <http://appliedinformatics.ru/>
3. Журнал «Информационные технологии» // Режим доступа: <http://novtex.ru/IT/index.htm>

6.3. Интернет-ресурсы

1. Вязовик Н.А. Программирование на Java. Режим доступа: <http://www.intuit.ru/studies/courses/16/16/info>.
2. Java Language and Virtual Machine Specifications. Режим доступа: <https://docs.oracle.com/javase/specs/index.html>.
3. Java™ Tutorials. Режим доступа: <https://docs.oracle.com/javase/tutorial/tutorialLearningPaths.html>
4. Java® Platform, Standard Edition & Java Development Kit Version 17 API Specification. Режим доступа: <https://docs.oracle.com/en/java/javase/17/docs/api/index.html>
5. Java Object Serialization Specification: Contents. Режим доступа: <https://docs.oracle.com/en/java/javase/17/docs/specs/serialization/index.html>
6. Documentation Comment Specification for the Standard Doclet (JDK 17). Режим доступа: <https://docs.oracle.com/en/java/javase/17/docs/specs/javadoc/doc-comment-spec.html>
7. Комплект разработки приложений Java SE Development Kit 17. Режим для скачивания и установки: <https://www.oracle.com/java/technologies/downloads/#java17>.
8. Среда разработки программного кода Eclipse v 4.2.2. Режим для скачивания и установки: <http://www.eclipse.org/downloads/packages/eclipse-classic-422/junosr2>

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Лекционные аудитории, оснащённые доской (для мела или маркера), экраном для проекционных систем, проектором и ноутбуком (420-3, 430-3).

Аудитории для проведения лабораторных занятий, оснащённые современными персональными компьютерами, объединёнными в локальную вычислительную сеть и укомплектованными необходимым системным и прикладным программным обеспечением (1226-3, 100-3, 511-3), аудитории вычислительного центра.

Компьютеры должны иметь выход в Интернет для получения доступа к справочной информации.

Среда разработки - Eclipse.

Рабочую программу составил ст. преп. каф. ФиПМ Воронова Н.М.

(ФИО, должность, подпись)

Рецензент

генеральный директор ООО «ФС Сервис»

Д.С. Квасов

(место работы, должность, ФИО, подпись)

Программа рассмотрена и одобрена на заседании кафедры ФиПМ

Протокол №1 от 30.08.2021 года

Заведующий кафедрой

С.М. Аракелян

(ФИО, подпись)

Рабочая программа рассмотрена и одобрена

на заседании учебно-методической комиссии направления 01.03.02

Протокол №1 от 30.08.2021 года

Председатель комиссии

С.М. Аракелян

(ФИО, должность, подпись)

ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Рабочая программа одобрена на 20²² / 20²³ учебный года

Протокол заседания кафедры № 1 от 30.08.21 года

Заведующий кафедрой

С.М. Аракелян

Рабочая программа одобрена на 20___ / 20___ учебный года

Протокол заседания кафедры № ___ от ___ года

Заведующий кафедрой

Рабочая программа одобрена на 20___ / 20___ учебный года

Протокол заседания кафедры № ___ от ___ года

Заведующий кафедрой