

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)



УТВЕРЖДАЮ

Проректор
по образовательной деятельности

А.А.Панфилов

« 02 » 09 20 19 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
РАЗРАБОТКА КРОСС-ПЛАТФОРМЕННЫХ ПРИЛОЖЕНИЙ
(наименование дисциплины)

Направление подготовки: 02.03.02 «Фундаментальная информатика и информационные технологии»

Профиль/программа подготовки: Мобильные и Интернет-технологии

Уровень высшего образования: бакалавриат

Форма обучения: очная

Семестр	Трудоемкость зач. ед./ час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	СРС, час.	Форма промежуточной аттестации (экзамен/зачет/за чет с оценкой)
3	5 / 180	36	—	36	81	Экзамен (27 ч.), КР
Итого	5 / 180	36	—	36	81	Экзамен (27 ч.), КР

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). Для достижения цели предполагается решение следующих задач:

- изучение теоретической части (кроссплатформенный 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 интерфейсом).

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина относится к обязательным дисциплинам блока Блок 1. Дисциплины (модули) учебного плана.

Изучение данной дисциплины проходит в 3-м семестре и базируется на знаниях, приобретённых студентами в рамках общеобразовательных курсов по программированию:

- “Основы программирования”
- “Объектно-ориентированное программирование”

Для усвоения курса необходимо:

- знание основ процедурного программирования
- знание основ объектно-ориентированного программирования
- умение самостоятельно разрабатывать и тестировать приложения на одном из языков программирования высокого уровня (Pascal, C, C++, C#).

Для успешного усвоения курса приветствуется знание языка C++ и глубокое понимание его объектно-ориентированных возможностей.

Знания и практические навыки данного курса могут быть применены:

- при написании курсовых работ и проектов по смежным дисциплинам, требующим знания языков и технологий программирования
- при написании выпускной квалификационной работы
- для профессионального использования при трудоустройстве в IT-компаниях, занимающиеся разработкой программного обеспечения на платформе Java 2 Standart Edition (J2SE)

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП:

Коды формируемых компетенций	Уровень освоения компетенций	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций (показатели освоения компетенций)
ОПК-2	Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной	Знать: <ul style="list-style-type: none">– основные положения и концепции в области программирования;– архитектура языков программирования;– основная терминология в области программного обеспечения;– знаком с содержанием Единого реестра российских программ. Уметь: <ul style="list-style-type: none">– умеет осуществлять обоснованный

Коды формируемых компетенций	Уровень освоения компетенций	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций (показатели освоения компетенций)
	деятельности	<p>выбор компьютерных/суперкомпьютерных методов и необходимого программного обеспечения при решении задач профессиональной деятельности.</p> <p>Владеть:</p> <ul style="list-style-type: none"> – имеет навыки применения данных методов и программного обеспечения при решении конкретных задач.
ОПК-3	Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	<p>Знать:</p> <ul style="list-style-type: none"> – методы теории алгоритмов; – методы системного и прикладного программирования; – принципы и методологии тестирования программного обеспечения; – принципы математического моделирования; – типовые (универсальные) математические (включая информационные и имитационные) модели, формулы, теоремы и методы, используемые в широком наборе областей применения прикладной математики. <p>Уметь:</p> <ul style="list-style-type: none"> – умеет соотносить знания в области программирования; – умеет определять и составлять информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем; – умеет осуществлять обоснованный выбор адекватных поставленной задаче базовых математических моделей; – умеет модифицировать базовые и (или) разрабатывать оригинальные математические модели в соответствии со спецификой поставленной задачи моделирования. <p>Владеть:</p> <ul style="list-style-type: none"> – навыки разработки программного обеспечения; – навыки выполнения математического моделирования от анализа постановки задачи до анализа результатов.
ОПК-4	Способен участвовать в разработке технической документации	<p>Знать:</p> <ul style="list-style-type: none"> – модели жизненного цикла информационных систем;

Коды формируемых компетенций	Уровень освоения компетенций	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций (показатели освоения компетенций)
	программных продуктов и комплексов с использованием стандартов, норм и правил, а также в управлении проектами создания информационных систем на стадиях жизненного цикла	<ul style="list-style-type: none"> – стандарты, нормы и правила документирования программных продуктов и комплексов. <p>Уметь:</p> <ul style="list-style-type: none"> – осуществлять управление проектами информационных систем. <p>Владеть:</p> <ul style="list-style-type: none"> – навыки документирования процесса создания информационных систем на разных стадиях жизненного цикла.
ОПК-5	Способен устанавливать и сопровождать программное обеспечение информационных систем и баз данных, в том числе отечественного происхождения, с учётом информационной безопасности	<p>Знать:</p> <ul style="list-style-type: none"> – принципы и основные процедуры установки и администрирования информационных систем и баз данных; – основные требования информационной безопасности; – знаком с содержанием Единого реестра российских программ. <p>Уметь:</p> <ul style="list-style-type: none"> – осуществлять обоснованный выбор и реализацию процессов установки и технического сопровождения информационных систем, и баз данных. <p>Владеть:</p> <ul style="list-style-type: none"> – навыки инсталляции и настройки программных комплексов, применения основ сетевых технологий.
ПК-1	Способен разрабатывать требования и проектировать программное обеспечение.	<p>Знать:</p> <ul style="list-style-type: none"> – возможности современных и перспективных средств разработки программных продуктов, технических средств; – методологии разработки программного обеспечения и технологии программирования; – методологии и технологии проектирования и использования баз данных; – методы и средства проектирования программного обеспечения и баз данных; – языки формализации функциональных спецификаций; – принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; – типовые решения, библиотеки

Коды формируемых компетенций	Уровень освоения компетенций	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций (показатели освоения компетенций)
		<p>программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения;</p> <ul style="list-style-type: none"> – основные концепции и атрибуты качества программного обеспечения; – механизмы авторизации и аутентификации; – стили написания кода; – методики разработки требований к системе и шаблоны документов требований к системе; – методы классического системного анализа; – методы проведения интервью с поставщиками требований; – стандарты оформления технических заданий. <p>Уметь:</p> <ul style="list-style-type: none"> – проводить анализ исполнения требований; – вырабатывать варианты реализации требований; – проводить оценку и обоснование рекомендуемых решений; – выбирать средства и варианты реализации программного обеспечения; – использовать существующие типовые решения и шаблоны проектирования программного обеспечения; – применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов; – выбирать методики разработки требований к системе и шаблоны документов требований к системе; – определять источники информации для требований к системе; – формулировать цели, исходя из анализа проблем, потребностей и возможностей; – оценивать соответствие требованиям существующих систем и их аналогов; – выполнять тестирование системы с целью проверки её реализации на соответствие требованиям; – формулировать и оформлять запросы на изменение требований.

Коды формируемых компетенций	Уровень освоения компетенций	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций (показатели освоения компетенций)
		<p>Владеть:</p> <ul style="list-style-type: none"> – навыки оценки возможностей, времени и трудоемкости реализации требований к программному обеспечению; – навыки согласования требований к программному обеспечению с заинтересованными сторонами; – навыки оценки и согласования сроков выполнения поставленных задач; – навыки разработки и согласования технических спецификаций на программное обеспечение; – навыки формирования и предоставления отчетности в соответствии с установленными регламентами; – навыки проектирования структур данных, баз данных, программных интерфейсов; – навыки выбора стиля написания кода; – навыки планирования работ по разработке требований к системе; – навыки анализа проблемной ситуации; – навыки участия в совещаниях рабочих групп; – навыки оформления требований заинтересованных лиц в документе бизнес-требований; – навыки согласования целей создания системы с заинтересованными лицами; – навыки оформления технического задания на систему; – навыки представления концепции, технического задания и изменений в них заинтересованным лицам
ПК-2	Способен анализировать требования к программному средству.	<p>Знать:</p> <ul style="list-style-type: none"> – методы анализа и тестирования требований к программному средству; – методы анализа, проектирования и разработки программного обеспечения. <p>Уметь:</p> <ul style="list-style-type: none"> – тестировать требования к программному средству; – оформлять документацию по тестированию; – анализировать требования на соответствие принятым стандартам и методам проектирования;

Коды формируемых компетенций	Уровень освоения компетенций	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций (показатели освоения компетенций)
		<ul style="list-style-type: none"> – использовать современные CASE-средства. <p>Владеть:</p> <ul style="list-style-type: none"> – навыки выявления противоречий в требованиях одинакового и различных уровней к программному средству; – навыки проверки осуществимости функционирования и сопровождения программного средства; – навыки определения возможности введения изменений и дополнений требований к программному средству.

4. ОБЪЕМ И СТРУКТУРА ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 5 зачетных единиц, 180 часов.

№ п/п	Наименование тем и/или разделов/тем дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Объем учебной работы, с применением интерактивных методов (в часах / %)	Формы текущего контроля успеваемости, форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные работы	СРС		
1	Введение в дисциплину	3	1	2	-	2	-	2/50%	Рейтинг-контроль 1
2	Основы языка Java	3	2-3	4	-	-	-	2/50%	
3	Объектная модель Java	3	4-5	4	-	-	-	8/50%	
			2-7	-	-	12	-		
4	Массивы	3	6	2	-	-	3	1/50%	
5	Приведение типов	3	7	2	-	-	3	1/50%	
6	Пакет java.util: коллекции	3	8	2	-	2	3	2/50%	Рейтинг-контроль 2
7	Исключения	3	9	2	-	2	3	2/50%	
8	Потоки данных(stream), пакет java.io	3	10	2	-	2	3	2/50%	
9	Работа с сетью, пакет java.net	3	11	2	-	2	3	2/50%	

10	Потоки выполнения	3	12-13	4	-	4	3	4/50%	
11	Пользовательский интерфейс	3	14-17	2	-	8	20	5/50%	
12	Архитектурный шаблон проектирования MVC.	3	15	2	-	-	-	1/50%	
13	Разработка клиент-серверных приложений	3	16-17	4	-	-	40	4/100%	
14	Подведение итогов	3	18	2	-	2	-	-	Рейтинг-контроль № 3
Наличие в дисциплине КП/КР		3	18	+	-	+	+		Зачет
Итого по дисциплине		3		36	-	36	81	36 (50%)	Экзамен (27ч.)

Содержание лекционных занятий по дисциплине

Раздел 1 Введение в дисциплину

Обзор программного обеспечения, необходимого для обучения. Понятие кроссплатформенности. Обзор платформы J2SE. Понятие модуля компиляции

Раздел 2 Основы языка Java

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

Раздел 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.

Раздел 4 Массивы

Массивы примитивных и ссылочных типов.

Раздел 5 Приведение типов

Приведение примитивных и ссылочных типов. Запрещенные приведения.

Раздел 6 Пакет java.util: коллекции

Java- коллекции

Раздел 7 Исключения

Понятие исключительной ситуации (ИС). Причины возникновения ИС. Классификация ИС. Обработка ИС (конструкция try-catch и try-catch-finally). Оператор throw. Пользовательские классы исключений.

Раздел 8 Потоки данных(stream), пакет java.io

Система ввода/вывода. Сериализация. Классы java.io.Reader и java.io.Writer. Класс java.io.File.

Раздел 9 Работа с сетью, пакет java.net:

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

Раздел 10 Потоки выполнения

Класс java.lang.Thread. Интерфейс java.lang.Runnable. Демон-потоки. Синхронизация.

Методы `wait()`, `notify()`, `notifyAll()` класса `Object`. Могопоточная архитектура в клиент-серверных приложениях.

Раздел 11 Пользовательский интерфейс

Пакет `java.awt`. Дерево компонент. Принципы отрисовки. Модель сообщений. Менеджеры компоновки.

Раздел 12 Архитектурный шаблон проектирования MVC

Архитектурный шаблон проектирования MVC на примере тестовой задачи.

Раздел 13 Разработка клиент-серверных приложений

Анализ предметной области. Use case diagram, class diagram, activity diagram. Разработка архитектуры приложения в концепции MVC.

Содержание лабораторных занятий по дисциплине

Лабораторная работа № 1

Раздел 1 Введение в дисциплину

Обзор программного обеспечения, необходимого для обучения. Обзор платформы J2SE. Понятие модуля компиляции

Лабораторная работа № 2

Раздел 3 Объектная модель Java

Классы и объекты. Модификаторы видимости. Конструкторы. Статические элементы класса. Инициализаторы. Наследование. Абстрактные классы. Интерфейсы. Механизм позднего связывания и полиморфизм.

Лабораторная работа № 3

Раздел 6 Пакет `java.util`: коллекции

Java- коллекции

Лабораторная работа № 4

Раздел 7 Исключения

Понятие исключительной ситуации (ИС). Обработка ИС (конструкция `try-catch` и `try-catch-finally`). Оператор `throw`. Пользовательские классы исключений.

Лабораторная работа № 5

Раздел 8 Потоки данных(`stream`), пакет `java.io`

Система ввода/вывода. Сериализация. Классы `java.io.Reader` и `java.io.Writer`. Класс `java.io.File`.

Лабораторная работа № 6

Раздел 9 Работа с сетью, пакет `java.net`:

Классы `java.net.Socket` и `java.net.ServerSocket`.

Лабораторная работа № 7

Раздел 10 Потоки выполнения

Класс `java.lang.Thread`. Интерфейс `java.lang.Runnable`. Могопоточная архитектура в клиент-серверных приложениях.

Лабораторная работа № 8

Раздел 11 Пользовательский интерфейс

Пакет java.awt. Дерево компонент. Принципы отрисовки. Модель сообщений. Менеджеры компоновки.

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В преподавании дисциплины «Объектно-ориентированное программирование» используются разнообразные образовательные технологии как традиционные, так и с применением активных и интерактивных методов обучения.

Активные и интерактивные методы обучения:

- *Интерактивная лекция (все разделы);*
- *Групповая дискуссия (все разделы);*
- *Анализ ситуаций (раздел № 13);*
- *Проектная работа в малых группах (раздел № 13);*

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ¹

ТЕКУЩИЙ КОНТРОЛЬ

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\?

Раздел №2

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

¹ При разработке оценочных средств были использованы материалы из источников, указанных в разделе 7 Рабочей программы

- 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?

### Раздел №3

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?

#### Раздел №4

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]);
```

#### Рейтинг-контроля №2

##### Примерные вопросы:

#### Раздел №5

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;
```

*Раздел №6*

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

*Раздел №7*

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;
    }
}

```

Раздел №8

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`?

Раздел №9

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?

Раздел №10

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?

Раздел №11

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 внешне 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

САМОСТОЯТЕЛЬНАЯ РАБОТА

На самостоятельное изучение выносятся раздел №11 - изучение специальных графических библиотек для разработки оконных приложений (на лекционных и лабораторных занятиях студенты знакомятся с базовой библиотекой java.awt). Графическую библиотеку студенты выбирают самостоятельно из списка:

- java.awt
- java.swt
- Swing
- JavaFX

Контрольными мероприятиями для оценки выполнения студентом самостоятельной работы является отчет по выполненному заданию.

КУРСОВАЯ РАБОТА

Студентам предлагается 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. Раздел файловой системы сервера, открытый для клиента, определяется ip-адресом клиента (например, если ip-адрес клиента лежит в диапазоне 100.100.50.*, то клиент 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.

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 так же 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), сервер на основе файловой системы.

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

ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ (экзамен)

Промежуточная аттестация 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

Фонд оценочных средств для проведения аттестации уровня сформированности компетенций обучающихся по дисциплине оформляется отдельным документом.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1. Книгообеспеченность

Наименование литературы: автор, название, вид издания, издательство	Год издания	КНИГООБЕСПЕЧЕННОСТЬ	
		Количество экземпляров изданий в библиотеке ВлГУ в соответствии с ФГОС ВО	Наличие в электронной библиотеке ВлГУ
1	2	3	4
Основная литература			
1.Вязовик Н.А. Программирование на Java [Электронный ресурс]/ Вязовик Н.А.— Электрон. текстовые данные.— Москва: Интернет-Университет Информационных Технологий	2016	0	Режим доступа: http://www.iprbookshop.ru/73710.html .— ЭБС «IPRbooks».

(ИНТУИТ), 2016.— 603 с.— 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.— Режим доступа: http://www.iprbookshop.ru/66812.html .— ЭБС «IPRbooks»	2017	0	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.— Режим доступа: http://www.iprbookshop.ru/90875.html .— ЭБС «IPRbooks»	2015	0	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.— Режим доступа: http://www.iprbookshop.ru/97750.html .— ЭБС «IPRbooks»	2018	0	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.— Режим доступа: http://www.iprbookshop.ru/64057.html .— ЭБС «IPRbooks»	2017	0	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.— Режим доступа: http://www.iprbookshop.ru/68692.html .— ЭБС «IPRbooks»	2015	0	Режим доступа: http://www.iprbookshop.ru/68692.html .— ЭБС «IPRbooks»

7.2. Интернет-ресурсы

1. Вязовик Н.А. Программирование на Java. Режим доступа:
<http://www.intuit.ru/studies/courses/16/16/info>.

2. The Java® Language Specification, Java SE 7 Edition.. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. Режим доступа: <http://docs.oracle.com/javase/specs/jls/se7/html/index.html>.
3. The Java® Virtual Machine Specification, Java SE 7 Edition. Tim Lindholm, Frank Yellin, Gilad Bracha, Alex Buckley. Режим доступа: <http://docs.oracle.com/javase/specs/jvms/se7/html/index.html>.
4. Java™ Platform, Standard Edition 7, API Specification. Режим доступа: <http://docs.oracle.com/javase/7/docs/api/index.html>
5. Java™ Tutorials. Режим доступа: <http://docs.oracle.com/javase/tutorial/index.html>

7.3. Программное обеспечение (ПО)

1. Комплект разработки приложений Java 2 SDK, Standart Edition, v1.7 (JDK). Режим для скачивания и установки: <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>
2. Среда исполнения приложений Java 2 Runtime Environment, Standart Edition, v 1.7 (JRE). Режим для скачивания и установки: <http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html>
3. Среда разработки программного кода Eclipse v 4.2.2. Режим для скачивания и установки: <http://www.eclipse.org/downloads/packages/eclipse-classic-422/junosr2>

Все ПО из списка является бесплатным, находится в открытом доступе и загружается для последующей установки с соответствующих сайтов

8. 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 - Microsoft Visual Studio.

Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО по направлению 02.03.02 «Фундаментальная информатика и информационные технологии»

Рабочую программу составили ст. преп. каф. ФиПИМ Воронова Н.М.
(ФИО, подпись)

Рецензент
(представитель работодателя) генеральный директор ООО «ФС Сервис» Квасов Д.С.
(место работы, должность, ФИО, подпись)

Программа рассмотрена и одобрена на заседании кафедры ФиПИМ

Протокол № 1 от 02.09.2019 года

Заведующий кафедрой _____ Аракелян С.М.
(ФИО, подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии направления 02.03.02 «Фундаментальная информатика и информационные технологии»

Протокол № 1 от 02.09.2019 года

Председатель комиссии _____ Аракелян С.М.
(ФИО, подпись)

ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Рабочая программа одобрена на 2020-2021 учебный год

Протокол заседания кафедры № 1 от 31.08.2020 года

Заведующий кафедрой _____ С. М. Аракелян

Рабочая программа одобрена на _____ учебный год

Протокол заседания кафедры № _____ от _____ года

Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год

Протокол заседания кафедры № _____ от _____ года

Заведующий кафедрой _____