

*Федеральное государственное автономное образовательное учреждение высшего образования  
«Российский университет дружбы народов»*

*Факультет физико-математических и естественных наук*

Рекомендовано МССН  
02.00.00 «Компьютерные и  
информационные науки»

## **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**Наименование дисциплины**

Java и его приложения

**Рекомендуется для направления подготовки**

02.03.02 — Фундаментальная информатика и информационные технологии

Квалификация (степень) выпускника

бакалавр

### 1. Цели и задачи дисциплины

Целями освоения дисциплины являются: получение студентами представления о возможностях одного из самых современных языков программирования высокого уровня для решения задач в различных областях деятельности, получение навыков построения приложений и апплетов.

Основными задачами освоения дисциплины являются: освоение студентами современных технологий программирования, богатой библиотеки классов языка для решения практических задач.

### 2. Место дисциплины в структуре ООП

Цикл, к которому относится дисциплина: Б1 «Дисциплины (модули)», обязательная часть.

В таблице № 1 приведены предшествующие и последующие дисциплины, направленные на формирование компетенций дисциплины в соответствии с матрицей компетенций ОП ВО.

Таблица № 1

#### Предшествующие и последующие дисциплины, направленные на формирование компетенций

№ п/п	Шифр и наименование компетенции	Предшествующие дисциплины	Последующие дисциплины (группы дисциплин)
Универсальные компетенции			
	-	-	-
Общепрофессиональные компетенции			
	ОПК-2, ОПК-3	Основы программирования, Технология программирования, Алгоритмы и анализ сложности	Компьютерная графика, модуль «Научные исследования в области искусственного интеллекта», Компьютерный практикум по интеллектуальным системам
Профессиональные компетенции (вид профессиональной деятельности – организационно-управленческая деятельность)			
	ПК-1, ПК-2	Основы программирования, Технология программирования, Алгоритмы и анализ сложности	Компьютерная графика, модуль «Научные исследования в области искусственного интеллекта», Компьютерный практикум по интеллектуальным системам
Профессионально-специализированные компетенции специализации			
	-	-	-

### 3. Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

**ОПК-2; ОПК-3; ПК-1; ПК-2**

*(указываются в соответствии с ОС ВО РУДН)*

ОПК-2. Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности.

- ОПК-2.1 Знает основные положения и концепции в области программирования, архитектуру языков программирования, знает основную терминологию, знаком с содержанием Единого Реестра Российских программ
- ОПК-2.2 Умеет анализировать типовые языки программирования, составлять программы
- ОПК-2.3 Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения

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.

- ОПК-3.1 Знает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей
- ОПК-3.2 Умеет соотносить знания в области программирования, интерпретацию прочитанного, определять и создавать информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем
- ОПК-3.3 Имеет практический опыт применения разработки программного обеспечения

ПК-1 Способен разрабатывать и отлаживать программный код

- ПК-1.1 Знает основы программирования; современные объектно-ориентированные языки программирования; современные структурные языки программирования; языки современных бизнес-приложений
- ПК-1.2 Умеет кодировать на языках программирования; тестировать результаты кодирования
- ПК-1.3 Владеет навыками разработки кода информационной системы; навыками верификации кода информационной системы

ПК-2 Способен выполнять работы и управлять работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы

- ПК-2.1 Знает инструменты и методы разработки архитектуры, проектирования и дизайна информационных систем; инструменты верификации программного кода
- ПК-2.2 Умеет проектировать и верифицировать архитектуру информационной системы; кодировать на языках программирования; тестировать результаты кодирования
- ПК-2.3 Владеть инструментами и методами проектирования и верификации архитектуры информационной системы, разработки и верификации структуры программного кода информационной системы

В результате изучения дисциплины студент должен:

***Знать:***

- принципы использования языков программирования;
- способы описания данных;
- основные синтаксические конструкции языка программирования высокого уровня;
- принципы алгоритмизации;

***Уметь:***

- разрабатывать алгоритмы решения вычислительных задач;
- реализовывать алгоритмы на языке программирования высокого уровня;
- работать на компьютере с использованием среды программирования на языке программирования высокого уровня;

**Владеть:**

- методами разработки алгоритмов;
- методами описания структур данных;
- навыками программирования на языке высокого уровня;
- навыками работы в различных средах программирования.

**4. Объем дисциплины и виды учебной работы**

Общая трудоемкость дисциплины составляет **3** зачетные единицы.

№	Вид учебной работы	Всего часов	Семестры
			Семестр 4, модуль 8
1.	<b>Аудиторные занятия (всего)</b>	54	54
	В том числе:		
1.1	Лекции	18	18
1.2	<i>Лабораторные работы (ЛР)</i>	36	36
2.	<b>Самостоятельная работа студентов (ак. часов)</b>	54	54
3.	<b>Общая трудоемкость (ак. часов)</b>	108	108
4.	<b>Общая трудоемкость (зачетных единиц)</b>	3	3

**5. Содержание дисциплины****5.1. Содержание разделов дисциплины**

№ п/п	Наименование раздела дисциплины	Содержание раздела
1.	Ядро языка Java	<ol style="list-style-type: none"> <li>1. История создания языка Java. Области применения. Идеология языка. Различия между C++ и Java. Характеристики простых типов данных.</li> <li>2. Операции, выражения, правила приведения типов. Операторы. Блок операторов. Управляющие операторы. Операторы перехода. Массивы в языке Java. Массив как параметр и тип возвращаемого значения метода.</li> <li>3. Аргументы метода main(). Классы в языке Java. Компоненты класса: данные и методы. Конструкторы. Ссылка this.</li> <li>4. Перегрузка методов. Final-компоненты. Статические компоненты класса. Операция «сборка мусора». Наследование в Java.</li> <li>5. Суперкласс и подклассы. Конструкторы подкласса. Доступ к компонентам при наследовании. Переопределение методов.</li> <li>6. Создание пакетов в Java. Определение, импорт, доступ к компонентам классов. Соответствие между иерархией пакетов и файловой системой.</li> <li>7. Абстрактные методы. Абстрактные классы и интерфейсы и их реализация. Оболочки простых типов. Обзор пакета java.lang. Обработка исключительных ситуаций. Иерархия классов</li> </ol>

		<p>исключений.</p> <p>8. Создание собственных классов исключений. Работа со строками. Классы String и StringBuffer. Многопоточное программирование. Класс Thread и интерфейс Runnable.</p> <p>9. Главный поток. Создание потока. Создание множественных потоков. Ожидание завершения потока. Приоритеты потоков.</p> <p>10. Межпоточные связи. Синхронизация потоков. Блокировка. Приостановка, возобновление и остановка потока.</p>
2.	Библиотека классов языка Java	<p>1. Пакет java.io. Ввод-вывод в языке Java. Байтовые и символьные потоки. Иерархия классов ввода-вывода. Предопределенные константы System.in, System.out, System.err. Стандартный ввод-вывод. Ввод-вывод в/из файл(а).</p> <p>2. Пакет java.awt. Создание графического интерфейса пользователя (ГИП). Основные компоненты AWT. Использование менеджеров компоновки. Оформление ГИП компонентами Swing.</p> <p>3. Пакет java.awt.event. Обработка событий. Обзор классов, описывающих события AWT.</p> <p>4. Пакет java.applet. Создание апплетов.</p> <p>5. Пакет java.awt.image. Создание, загрузка, обработка и вывод изображений. Создание анимации.</p> <p>6. Пакет java.util. Коллекции. Интерфейсы Collection, List, Set, SortedSet и классы их реализующие. Работа с картами отображений. Итераторы. Компараторы. Алгоритмы коллекций. Наследованные классы.</p>

## 5.2. Разделы дисциплины и виды занятий

№ п/п	Наименование раздела дисциплины	Лекции	Лабораторные работы	СРС	Всего час.
1.	Ядро языка Java	9	18	27	<b>54</b>
2.	Библиотека Java	9	18	27	<b>54</b>
	<b>Итого:</b>	<b>18</b>	<b>36</b>	<b>54</b>	<b>108</b>

## 6. Лабораторный практикум

№ п/п	№ раздела дисциплины	Наименование лабораторных работ	Трудоемкость (час.)
1.	1	Операторы. Блок операторов. Управляющие операторы. Операторы перехода.	2
		Массивы в языке Java.	2
		Массив как параметр и тип возвращаемого значения метода.	2
		Аргументы метода main(). Классы в языке Java. Компоненты класса: данные и методы.	2
		Конструкторы. Ссылка this. Перегрузка методов. Final-компоненты. Статические компоненты класса. Операция «сборка мусора».	3
		Наследование в Java. Суперкласс и подклассы. Конструкторы подкласса. Доступ к компонентам при наследовании. Переопределение методов. Создание	3

		пакетов в Java.	
		Использование интерфейсов для вычисления различных характеристик алгебраических функций.	2
		Создание многопоточных программ.	2
2.	2	Пакет java.io. Ввод-вывод в языке Java. Байтовые и символьные потоки.	2
		Пакет java.io. Ввод-вывод в языке Java. Байтовые и символьные потоки.	2
		Создание, загрузка, обработка и вывод изображений.	2
		Создание анимации.	2
		Написание программ, создающих ГИП и осуществляющих обработку событий	2
		Работа с картами отображений.	2
		Алгоритмы коллекций.	3
		Работа с полутоновыми и цветными изображениями	3

## 7. Практические занятия (семинары) не предусмотрены.

## 8. Материально-техническое обеспечение дисциплины

Учебная аудитория с ПК и проектором для проведения учебных занятий (в том числе для практического и лекционного типов занятий, индивидуальных и групповых консультаций, текущего контроля и промежуточной аттестации).

Компьютерные (дисплейные) классы с доступом к сети Интернет и электронно-образовательной среде Университета для выполнения обучающимися заданий лабораторного практикума, выполнения обучающимися самостоятельной работы и проведения компьютерного тестирования обучающихся (при необходимости).

## 9. Информационное обеспечение дисциплины

### а) программное обеспечение

- продукты Microsoft - операционная система, пакет офисных приложений, MS Teams и др. (подписка Enrollment for Education Solutions (EES)).
- Программное обеспечение со свободной лицензией (free):
  - браузер Chrome (лицензия Google Chrome Terms of Service)
  - Adobe Reader (лицензия Adobe Software License Agreement)
  - офисный пакет LibreOffice (лицензия MPL-2.0)
  - NetBeans (лицензия Apache License 2.0)
  - OpenJDK (лицензия GNU GPL 2, GPL)
- ОС Linux.
  - Офисный пакет LibreOffice (лицензия MPL-2.0).
  - ПО для просмотра формата pdf (например, evince (лицензия GPL-2+ CC-BY-SA-3.0)).
  - GNU Midnight Commander (Лицензия GNU GPL 3).

### б) базы данных, информационно-справочные и поисковые системы

1. Сайт библиотеки РУДН <http://lib.rudn.ru/>.
2. Сайт ТУИС <http://esystem.pfur.ru/>.

## 10. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

1. П. Ноутон, Г. Шилдт. Java-ТМ 2. Наиболее полное руководство. В подлиннике. Санкт-Петербург. «БХВ-Петербург», 2003.
2. Л.И. Кулькова, П.П. Фролов. Программирование на языке. Java. Пособие для начинающих. Москва, изд-во РУДН, 2005.
3. Кулькова Л.И., Салпагаров С.И. Сборник задач и упражнений по Java Москва, изд-во РУДН, 2016
4. И. Хабибуллин. Самоучитель Java 2. «БХВ-Петербург», 2007.
5. Русскоязычная версия сайта Sun Microsystems. <http://www.sun.ru/java>.
6. Документация от Sun доступна по адресу <http://java.sun.com/products/jdk/1.3/docs/index.html>.
7. Русскоязычный сайт по Java <http://www.javable.com/>.
8. Популярный англоязычный сайт <http://www.javaworld.com/>.
9. Лекции фирмы SUN. Русскоязычная версия сайта Sun Microsystems. <http://www.sun.ru/java>.

б) дополнительная литература:

1. М. Морган. Java-ТМ 2. Руководство разработчика. Москва - Санкт-Петербург- Киев, 2000.
2. Л.И. Кулькова. Обработка изображений на языке Java. Москва, изд-во РУДН, 2013
3. Дж. Вебер, Технология Java-ТМ в подлиннике. «БХВ-Петербург», 2001

### 11. Методические указания для обучающихся по освоению дисциплины

Учебным планом на изучение дисциплины отводится один семестр. В течение семестра выполняются лабораторные работы и контрольные мероприятия. В конце семестра производится итоговый контроль знаний.

#### 11.1 Структура лабораторных (практических) занятий

1. Задания по лабораторным работам выполняются индивидуально в дисплейных классах в соответствии с календарным планом и методическими указаниями по выполнению лабораторных работ по дисциплине.
2. По результатам выполнения каждой лабораторной работы студентом готовится компьютерная программа с корректным набором данных.

#### 11.2. Самостоятельная работа студента

1. Часть лабораторных работ предусматривает задания для индивидуальной самостоятельной работы студента, обязательные для выполнения.
2. Выполнение заданий для самостоятельной работы позволяет студенту приобрести дополнительные навыки и закрепить знания по изучаемой теме.

## 12. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

ФОС по дисциплине представлен в приложении к данной программе.

Программа составлена в соответствии с требованиями ОС ВО РУДН.

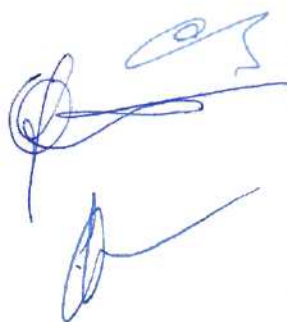
### Разработчик:

доцент кафедры информационных технологий

Зав. кафедрой информационных технологий

### Руководитель программы

Заведующий кафедрой  
прикладной информатики  
и теории вероятностей, д.т.н., проф.



С.И. Салпагаров

Ю.Н. Орлов

К.Е. Самуйлов

*Федеральное государственное автономное образовательное учреждение высшего образования  
«Российский университет дружбы народов»*

*Факультет физико-математических и естественных наук*

*Кафедра информационных технологий*

## **ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

**Наименование дисциплины**

Java и его приложения

**Рекомендуется для направления подготовки:**

02.03.02 — Фундаментальная информатика и информационные технологии

**Квалификация (степень) выпускника**

Бакалавр



## Паспорт фонда оценочных средств по дисциплине

Направление: 02.03.02 — Фундаментальная информатика и информационные технологии

Дисциплина: Java и его приложения

Код контролируемой компетенции	Контролируемый раздел дисциплины	Контролируемая тема дисциплины	Наименование оценочного средства				Баллы темы	Баллы раздела
			Текущий контроль			Промежуточная аттестация		
			Опрос	Выполнен е ЛР	Выполнен е ДЗ			
ОПК-2, ОПК-3, ПК-1, ПК-2	Раздел 1: Ядро языка Java	Тема 1: Классы в языке Java.	2	3			10	40
		Тема 2: Многопоточное программирование	2	8		5	15	
		Тема 3: Интерфейсы в языке Java.	5				10	
		Тема 4: Обработка исключительных ситуаций.	5				5	
	Раздел 2: Библиотека пакетов Java	Тема 1: Пакет java.awt. Создание графического интерфейса пользователя (ГИП).Использование менеджеров компоновки.	5	10			15	60
		Тема 2: Пакет java.awt.event. Обработка событий.	5		0		15	
		Тема 3: Пакет java.awt.image Создание, загрузка, обработка и вывод изображений. Создание анимации	5	10		10	25	
		Тема 4: Пакет java.util. Коллекции.				5	5	
		<b>ИТОГО:</b>	29	31	0	20	100	100

Процесс изучения дисциплины направлен на формирование следующих компетенций: ОПК-2; ОПК-3; ПК-1; ПК-2  
(указываются в соответствии с ОС ВО РУДН)

ОПК-2. Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности.

- ОПК-2.1 Знает основные положения и концепции в области программирования, архитектуру языков программирования, знает основную терминологию, знаком с содержанием Единого Реестра Российских программ
- ОПК-2.2 Умеет анализировать типовые языки программирования, составлять программы
- ОПК-2.3 Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения

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.

- ОПК-3.1 Знает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей
- ОПК-3.2 Умеет соотносить знания в области программирования, интерпретацию прочитанного, определять и создавать информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем
- ОПК-3.3 Имеет практический опыт применения разработки программного обеспечения

ПК-1 Способен разрабатывать и отлаживать программный код

- ПК-1.1 Знает основы программирования; современные объектно-ориентированные языки программирования; современные структурные языки программирования; языки современных бизнес-приложений
- ПК-1.2 Умеет кодировать на языках программирования; тестировать результаты кодирования
- ПК-1.3 Владеет навыками разработки кода информационной системы; навыками верификации кода информационной системы

ПК-2 Способен выполнять работы и управлять работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы

- ПК-2.1 Знает инструменты и методы разработки архитектуры, проектирования и дизайна информационных систем; инструменты верификации программного кода
- ПК-2.2 Умеет проектировать и верифицировать архитектуру информационной системы; кодировать на языках программирования; тестировать результаты кодирования
- ПК-2.3 Владеть инструментами и методами проектирования и верификации архитектуры информационной системы, разработки и верификации структуры программного кода информационной системы

## Примерный перечень оценочных средств

по дисциплине Java и его приложения

п/п	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
<b><i>Аудиторная работа</i></b>			
1	Опрос	Средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу или теме.	Вопросы по темам/разделам дисциплины
2	Лабораторная работа	Система практических заданий, направленных на формирование практических навыков у обучающегося.	Фонд практических заданий
3	Экзамен	Оценка работы студента в течение семестра (года, всего срока обучения и др.) и призван выявить уровень, прочность и систематичность полученных им теоретических и практических знаний, приобретения навыков самостоятельной работы, развития творческого мышления, умение синтезировать полученные знания и применять их в решении практических задач.	Примеры заданий/вопросов, пример экзаменационного билета
<b><i>Самостоятельная работа</i></b>			
1	Выполнение домашних заданий	Различают задачи и задания: А) репродуктивного уровня, позволяющие оценивать и диагностировать знание фактического материала (базовые понятия, алгоритмы, факты) и умение правильно использовать специальные термины и понятия, узнавание объектов изучения в рамках определенного раздела дисциплины; Б) реконструктивного уровня, позволяющие оценивать и диагностировать умения синтезировать, анализировать и обобщать фактический и теоретический материал с формулированием конкретных выводов, установлением причинно-следственных связей; В) творческого уровня, позволяющие оценивать и диагностировать умения, интегрировать знания различных областей, аргументировать собственную точку зрения	Комплект разноуровневых задач и заданий.
2	Курсовая работа/курсовый проект	Вид самостоятельной письменной работы, направленный на творческое освоение общепрофессиональных и профильных профессиональных дисциплин и выработку соответствующих профессиональных компетенций.	Темы курсовых заданий

# Комплект примеров экзаменационных билетов

Дисциплина Java и его приложения

## ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Что такое конструктор класса? Может ли класс не содержать ни одного конструктора? Чем конструктор отличается от любого другого метода класса?
2. Результаты сдачи сессии хранятся в связном списке (*LinkedList*). Элементами связного списка являются объекты класса *Student*. Информация о каждом студенте содержит фамилию и массив отметок. Сформировать другой связный список, каждый объект которого содержит фамилию студента и средний балл. Распечатать его на экран.

## ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 2

1. Что такое интерфейс? Можно ли создавать объекты типа интерфейс? Как они используются?
2. В файле "*a.txt*" записан список студентов, сдававших экзамен, с указанием фамилии, группы и отметки. Информация о каждом студенте хранится в отдельной строке. Прочитать информацию в связный список (*LinkedList*), отсортировать его по убыванию отметки, распечатать на экран сначала список студентов, получивших отлично, затем – хорошо и т.д.

## ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 3

1. Что такое переопределенные и перегруженные методы. Привести примеры.
2. В файле "*a.txt*" записано несколько строк. Каждая строка содержит название государства с указанием через пробел его столицы. Сформировать коллекцию типа *TreeSet*, отсортированную по названию столиц. Распечатать коллекцию на экран.

## Комплект разноуровневых задач и заданий.

*А. Задачи репродуктивного уровня.*

### Вариант 1.

- Объявить класс *Point* (точка) на плоскости. Точка задается декартовыми координатами.
- Определить
  - конструктор с двумя действительными параметрами.
  - метод:
    - расстояние *dist()* от данной точки до начала координат.
- Перегрузить метод *dist()* до любой точки на плоскости.
- В методе *main()* протестировать определенные Вами методы.

### Вариант 2.

- Объявить класс *Point* (точка) на плоскости. Точка задается декартовыми координатами.
- Определить
  - конструктор с двумя действительными параметрами.
  - методы:
    - сдвиг *move()* точки на плоскости
    - переопределить метод *toString()*
- Объявить подкласс круг *Circle* класса *Point*

- Для класса *Circle*
  - определить методы
    - area()* площадь круга.
    - getx()* координата по *x* самой правой точки
  - переопределить метод *toString()*
  - В методе *main()* объявить ссылку на суперкласс и создать объект класса *Circle*
- Протестировать все определенные Вами методы, используя ссылку на суперкласс и объект класса *Circle*.

### Вариант 3.

- Объявить класс *Square* (квадрат). Квадрат задается длиной стороны.
- Определить
  - конструктор с одним действительным параметром.
  - методы:
    - периметр *perimetr()* квадрата
    - площадь *area()* квадрата
- Объявить подкласс вписанная окружность *InCircle*.
- Определить метод *dlna()*, вычисляющий длину окружности
- Для класса *InCircle*. переопределить метод *area()* площадь круга.
- В методе *main()* объявить ссылку на суперкласс и создать объект класса *InCircle*.
- Протестировать определенные Вами методы, используя ссылку на суперкласс и объект класса *InCircle*.

*Б. Задачи реконструктивного уровня.*

### Вариант 1.

- В методе *main()* объявить действительную квадратную матрицу *A* порядка *n*. Значение *n* и элементы матрицы передать из командной строки.
- Добавить в тот же класс метод, который формирует целочисленную квадратную матрицу *B* того же порядка. Элемент матрицы *B* равен 1, если соответствующий ему элемент исходной матрицы *A* больше соседних элементов, расположенных в этой же строке, и равен 0 в противном случае.
- Целочисленную матрицу *B* распечатать на экран. Печать матрицы оформить в виде отдельного метода.

### Вариант 2.

- В методе *main()* объявить действительную квадратную матрицу *A* порядка *n*. Значение *n* и элементы матрицы передать из командной строки.
- В тот же класс добавить метод, который формирует одномерный массив *X* из *n* элементов. Каждый элемент массива *X* равен максимальному элементу соответствующей строки матрицы.
- Вычисление значения выражения  $X[1]*X[n]+X[2]*X[n-1]+...+X[n]*X[1]$  оформить в виде метода того же класса.
- Результат распечатать в методе *main()*.

### Вариант 3.

- В методе *main()* объявить действительную квадратную матрицу *A* порядка *n* и натуральные числа *i* и *j* ( $0 \leq i < n, 0 \leq j < n$ ). Значения *n*, *i*, *j* и элементы матрицы передать из командной строки.
- В тот же класс добавить метод, который из матрицы *A* удаляет *i*-ую строку и *j*-ый столбец.
- Печать матрицы оформить в виде отдельного метода того же класса.

– Результат распечатать в методе *main()*.

### *В. Задачи творческого уровня*

#### **Вариант 1.**

Нарисовать на экране лампочку и обеспечить возможность зажигать и гасить ее с помощью объектов «on» и «off» типа Button.

Спираль зажженной и погашенной лампочек окрашивается в разные цвета.

#### **Вариант 2.**

На экране изображена окружность радиуса 20 с центром в точке (100,100). Необходимо предоставить возможность управлять размером и положением окружности.

Клавиша ">" увеличивает радиус на 5 точек.

Клавиша "<" уменьшает радиус на 5 точек.

Клавиши управления курсором ←↑↓→ перемещают окружность по экрану в соответствующем направлении.

#### **Вариант 3.**

Четыре жука расположены в вершинах квадрата со стороной *a*. Заставить жуков двигаться по часовой стрелке можно с помощью клавиш управления курсором ←↑↓→.

Так, например, жука расположенного на верхней стороне квадрата можно заставить двигаться клавишей → и т.д. Остановить движение жуков можно клавишей S.

## **Критерии оценки**

по дисциплине Java и его приложения

В связи с отсутствием в программе дисциплины лекционных часов, итоговая оценка выставляется по сумме набранных баллов за лабораторные работы, реферат и доклад по материалам реферата. Для учащихся, которые набрали от 31 до 50 баллов, проводится экзамен (зачет), в рамках которого за ответы на вопросы билетов учащиеся могут набрать недостающее до зачета (51) число баллов.

#### **95-100 баллов:**

- полное выполнение лабораторных работ;
- высокий уровень подготовки реферата;
- высокий уровень подготовки доклада;
- активное участие в мероприятиях, предусмотренных программой дисциплины;
- систематизированное, глубокое и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- умение эффективно использовать методiku программы дисциплины в постановке и решении научных и профессиональных задач;
- выраженная способность самостоятельно и творчески решать поставленные задачи;
- полная самостоятельность и творческий подход при изложении материала по программе дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной программой дисциплины и преподавателем.

#### **86- 94 балла:**

- полное выполнение лабораторных работ;
- хороший уровень подготовки реферата;

- хороший уровень подготовки доклада;
- участие в мероприятиях, предусмотренных программой дисциплины;
- систематизированное, глубокое и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- умение эффективно использовать методику программы дисциплины в постановке и решении научных и профессиональных задач;
- способность самостоятельно решать поставленные задачи в нестандартных производственных ситуациях;
- усвоение основной и дополнительной литературы, рекомендованной программой дисциплины и преподавателем.

**69-85 баллов:**

- частичное выполнение лабораторных работ;
- подготовка реферата;
- подготовка доклада;
- участие в мероприятиях, предусмотренных программой дисциплины;
- систематизированное и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- умение использовать методику программы дисциплины в постановке и решении научных и профессиональных задач;
- усвоение основной литературы, рекомендованной программой дисциплины.

**51-68 баллов:**

- частичное выполнение лабораторных работ;
- участие в мероприятиях, предусмотренных программой дисциплины;
- систематизированное и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- удовлетворительное умение использовать методику программы дисциплины в постановке и решении научных и профессиональных задач;
- удовлетворительное усвоение основной литературы;

**31 - 50 баллов – НЕ ЗАЧТЕНО:**

- частичное выполнение лабораторных работ;
- недостаточно полный объем навыков и компетенции в рамках программы дисциплины;
- неумение использовать в практической деятельности научной терминологии, изложение ответа на вопросы с существенными стилистическими и логическими ошибками;
- слабое умение использовать методику программы дисциплины в постановке и решении научных и профессиональных задач;
- удовлетворительное усвоение основной литературы.

**0-30 баллов, НЕ ЗАЧТЕНО:**

- отсутствие умений, навыков, знаний и компетенции в рамках программы дисциплины;
- невыполнение лабораторных заданий; отказ от ответа по программе дисциплины;
- игнорирование занятий по дисциплине по неуважительной причине.