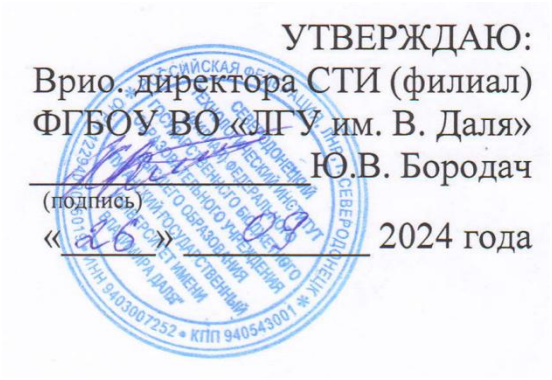
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Луганский государственный университет имени Владимира Даля»
(ФГБОУ ВО «ЛГУ им. В. Даля»)

Северодонецкий технологический институт
Кафедра информационных технологий, приборостроения и электротехники



РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

«Кроссплатформенное программирование»

По направлению подготовки: 09.03.02 «Информационные системы и технологии»

Профиль «Информационные ситемы и технологии»

Северодонецк – 2024

Лист согласования РПУД

Рабочая программа учебной дисциплины «Кроссплатформенное программирование» по направлению подготовки: 09.03.02 «Информационные системы и технологии» (программа бакалавриата «Информационные ситемы и технологии») – 15 с.

Рабочая программа учебной дисциплины «Кроссплатформенное программирование» разработана в соответствии Федеральным государственным образовательным стандартом высшего образования по направлению подготовки 09.03.02 «Информационные системы и технологии» утвержденный приказом Министерства науки и высшего образования Российской Федерации от 19 сентября 2017 г. № 926 (с изменениями и дополнениями в соответствии с приказами Министерства образования и науки Российской Федерации №1456 от 26.11.2020 г., №83 от 08.02.2021 г., №662 от 19.07.2022 г., №208 от 27.02.2023 г.).

СОСТАВИТЕЛЬ:
старший преподаватель Кузнецова Е.В.

Рабочая программа дисциплины утверждена на заседании кафедры информационных технологий, приборостроения и электротехники «_05_» _сентября_ 2024 г., протокол №_1_.

Заведующий кафедрой ИТПЭ _____ В.Г. Чебан

Переутверждена: «____» _____ 20___г., протокол №___.

Рекомендована на заседании учебно-методической комиссии Северодонецкого технологического института (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Луганский государственный университет имени Владимира Даля» «_16_» _сентября_ 2024 г., протокол №_1_.

Председатель учебно-методической комиссии
СТИ (филиал) ФГБОУ ВО «ЛГУ им. В.Даля» _____ Ю.В. Бородач

Структура и содержание дисциплины

1. Цели и задачи дисциплины, ее место в учебном процессе

Цель изучения дисциплины – приобретение студентами теоретических знаний и практических навыков по программированию в кроссплатформенной системе программирования Java и разработке кроссплатформенных клиентских и серверных приложений различного назначения приложений.

Задачи:

- изучение принципов функционирования микропроцессоров семейства Intel 8086/8088, набора регистров, организации памяти, сегментирования памяти, способов адресации, основного набора команд ассемблера
- приобретение студентами набора знаний из области архитектуры микропроцессорных систем и программирования на языке низкого уровня.
- освоение подходов к созданию консольных и визуальных кроссплат-форменных программ;
- изучение синтаксиса и семантики языка Java;
- ознакомление с основными технологиями разработки кроссплат-форменных клиентских и серверных приложений различного назначения приложений;
- изучение особенностей объектно-ориентированного программирования в Java;
- изучение основных принципов разработки программ на Java;
- ознакомление с основными библиотеками языка Java;

2. Место дисциплины в структуре ООП ВО.

Дисциплина «Кроссплатформенное программирование» относится к дисциплинам входящим в часть, формируемую участниками образовательных отношений, учебного плана подготовки студентов по направлению подготовки 09.03.02 Информационные системы и технологии.

Необходимыми условиями для освоения дисциплины являются: знания одного из классических процедурно-ориентированных языков, предпочтительно языка C++, умения разрабатывать алгоритмы решения типовых задач программирования, навыки использования текстовых редакторов для работы с текстами программ.

Содержание дисциплины является логическим продолжением содержания дисциплин «Объектно-ориентированное программирование», «Дискретная математика», «Технологии обработки информации» и служит основой для освоения дисциплин «Администрирование баз данных Oracle»,

«Построение распределенных систем мониторинга», «Прикладное программное обеспечение для управления предприятиями», «Разработка приложений для мобильных устройств».

3. Требования к результатам освоения содержания дисциплины

Студенты, завершившие изучение дисциплины «Кроссплатформенное программирование», должны

знать

- архитектуру микропроцессоров семейства Intel 8086/8080, назначение и состав регистров, организацию памяти микропроцессорных систем, способы адресации к памяти.
- состав команд микропроцессора,
- принципы передачи параметров между подпрограммами;
- основные принципы и этапы разработки, внедрения и адаптации прикладного программного обеспечения;
- основные принципы и этапы программирования приложений и создания программных прототипов решения прикладных задач
- основные инструменты разработки Java программ;
- конструкции классов в языке Java;
- особенности использования механизма исключений для создания устойчивых приложений;
- основные принципы объектно-ориентированного программирования: абстракция, полиморфизм, инкапсуляция, полиморфизм;
- концепции ООП;
- принципы создания эффективных иерархий классов;
- основные способы организации данных.

уметь

- разрабатывать программы на языке ассемблера,
- разрабатывать подпрограммы на языке ассемблера для программ на языках высокого уровня;
- разрабатывать структуры классов для решения проблемной задачи;
- использовать программы, построенные на основе взаимодействия объектов;
- использовать стандартные шаблонные конструкции для решения типовых задач взаимодействия объектов;
- выполнять декомпозицию поставленной задачи на стандартные подзадачи, для решения которых существуют типовые паттерны;
- разрабатывать, внедрять и адаптировать прикладное программное обеспечение;

- программировать приложения и создавать программные прототипы решения прикладных задач

владеть
- основными принципами разработки, внедрения и адаптации прикладного программного обеспечения
- основными принципами программирования приложений и создания программных прототипов решения прикладных задач
- навыками пользования современными интегрированными пакетами разработки и отладки объектно-ориентированных программ.

Перечисленные результаты образования являются основой для формирования компетенций в соответствии с государственными образовательными стандартами ВО и требованиями к результатам освоения основной образовательной программы (ООП):

профессиональных

ПК-03 Способность осуществлять разработку, отладку, проверку работоспособности и безопасности, расчет экономической эффективности информационных систем и технологий, модификацию программного обеспечения

4. Структура и содержание дисциплины

4.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов (з.е.)		
	Очная форма	Очно-заочная форма	Заочная форма
Объем учебной дисциплины (всего)	252 (7 з.е.)	-	252 (7 з.е.)
Обязательная аудиторная учебная нагрузка дисциплины (всего) в том числе:	140	-	28
Лекции	70	-	14
Семинарские занятия	-	-	-
Практические занятия	-	-	-
Лабораторные работы	70	-	14
Курсовая работа (курсовой проект)	-	-	-
Индивидуальное задание	-	-	-
Самостоятельная работа студента (всего)	112	-	224
Форма аттестации	экзамен	-	экзамен

4.2. Содержание разделов дисциплины

Тема 1. Ассемблирование и выполнение программы.

Введение в семейство микропроцессоров Intel x86. Архитектура 16-ти и 32-х разрядных МП. Регистры процессора и их назначение. Логическая структура МП. Принцип работы устройства выполнения и шинного интерфейса. Очередь команд. Процессы ассемблирования. Назначение компилятора, компоновщика, загрузчика и отладчика. Машинная адресация. Сегменты памяти. Функциональная организация ЭВМ. Компоновка, исполнения, трассировки программы. Директивы сегментации.

Тема 2. Организация программы и определение данных

Язык Assembler, его синтаксис. Структура com и. exe программ. Формат предоставления базовых данных в ПК. Типы данных. Директивы определения данных. Изучение способов определения данных на языке Assembler. Представление данных в памяти ЭВМ. Стандартные модели памяти. Инструментальные средства разработки программ на языке Assembler. Создание простой. com программы. Способы адресации в командах языка Assembler. Регистровая, непосредственная, прямая, косвенная адресации, масштабирования при адресации.

Тема 3. Арифметические и логические команды

Команды передачи данных. Способы адресации в командах пересылки. Арифметические команды. Формат, типы данных, особенности использования. Логические команды и команды перехода. Формат, типы данных, особенности использования. Программирование линейных вычислительных процессов. Программирование задач с ветвлением. Программирование задач с циклами.

Тема 4. Команды обработки строк.

Команды обработки блоков данных. Формат, особенности использования. Типовые операции с массивами. Организация двумерных массивов. Сложные типы данных: структуры, объединения.

Тема 5. Подпрограммы.

Описание и вызов процедур. Команды передачи управления. Передача аргументов в процедуру через регистры, общую область памяти, стек. Понятие о макросредства языка Assembler. Макрокоманды и макродирективы.

Тема 6. Основные принципы кроссплатформенного программирования.

Определение кроссплатформенного программного обеспечения. Варианты кроссплатформенности на уровне компиляции и на уровне выполнения, дается обзор технологий. В контексте уровней кроссплатформенности характерные для каждого уровня языки программирования, C, C++ для уровня компиляции и C# и Java для уровня выполнения. Достоинства и недостатки кроссплатформенности на уровне выполнения, сравнение вариантов реализации кроссплатформенности в разных языках. Понятие кросс-компиляции и попадающие под это понятие компиляторы. Аппаратная кроссплатформенности. Понятие эмулятора как аппаратной, так и программной платформы. Сравнительная характеристика различных кроссплатформенных среды разработки Code::Blocks, Eclipse, MonoDevelop, QDevelop.

Тема 7. Основные характеристики системы программирования Java.

История создания. Основные версии Java. Инструментальная среда разработки java программ jdk. Среда исполнения jre. Специализированные пакеты j2se, j2ee, j2me. Области применения Java программ. Интегрированные среды разработки Java программ – IntelliJ Idea, NetBeans, Eclipse, JDeveloper. Сравнительная характеристика кроссплатформенных систем программирования Java, C#, Python.

Тема 8. Особенности языка программирования Java.

Типы данных в Java. Выражения и операторы Java. Условные операторы. Операторы цикла. Операторы изменения порядка выполнения операторов. Базовые и ссылочные типы данных.

Тема 9. Объектная модель языка Java.

Объявление классов. Области видимости классов и элементов классов. Поля и методы класса. Члены класса и их характеристики. Области видимости членов класса. Указатель this. Конструкторы. Виды конструкторов. Инициализаторы. Деструкторы. Динамические и статические члены класса. Наследование классов. Особенности использования конструкторов при реализации отношений наследования. Абстрактные классы. Переопределение методов. Полиморфизм. Интерфейсы и их характеристика. Модульное программирование в Java. Пакеты Java. Особенности обработки исключений в Java. Обобщенное программирование в Java.

Тема 10. Стандартные пакеты Java.

Пакет java.util.lang. Классы Object и Class. Их методы и поля. Рефлексия в Java. Ввод-вывод в Java, потоки и их иерархия. Сериализация объектов. Коллекции, их разновидности и методы работы с ними. Итераторы и алгоритмы.

4.3. Лекции

№ п/п	Название темы	Объем часов		
		Очная форма	Очно-заочная форма	Заочная форма
Семестр 5				
1	Введение в семейство микропроцессоров Intel x86. Архитектура 16-ти и 32-х разрядных МП.	2	-	2
2	Регистры процессора и их назначение	2	-	2
3	Язык Assembler, его синтаксис. Структура .com и .exe программ	2	-	2
4	Типы данных. Директивы определения данных.	2	-	-
5	Команды передачи данных. Способы адресации в командах пересылки.	4	-	-
6	Арифметические команды. Формат, типы данных, особенности использования.	4	-	-
7	Логические команды и команды перехода. Формат, типы данных, особенности использования.	4	-	-
8	Команды обработки блоков данных. Формат, особенности использования.	4	-	-
9	Описание и вызов процедур. Команды передачи управления	4	-	-
Семестр 6				

10	Что такое Java? История создания	4	-	2
11	Лексика языка	4	-	2
12	Базовые типы данных	4	-	2
13	Объявление классов.	4	-	2
14	Преобразование типов.	4	-	-
15	Объектная модель Java.	4	-	-
16	Массивы.	4	-	-
17	Операторы и структура кода . Исключения.	4	-	-
18	Пакет java.lang.	4	-	-
19	Пакет java.util.	4	-	-
20	Пакеты java.io, java.nio2	4	-	-
21	Обработка исключительных ситуаций в java	4	-	-
Итого:		70	-	14

4.4. Практические (семинарские) занятия

Практические занятия по дисциплине не предусмотрены.

4.5. Лабораторные работы

№ п/п	Название темы	Объем часов		
		Очная форма	Очно-заочная форма	Заочная форма
Семестр 5				
1	Исследование основных инструментальных утилит Java	2	-	2
2	Исследование методов документирования кода Java программы	2	-	2
3	Основы языка Java, массивы, примитивные типы, объявление классов	2	-	-
4	Основы языка Java, перегрузка и перекрытие методов, наследование	2	-	-
5	Основы языка Java. Перегрузка и перекрытие методов, наследование. Классы-оболочки	4	-	-
6	Основы языка Java. Наследование, тригонометрические функции класса Math	4	-	-
7	Основы языка Java. Работа с изменяемыми и неизменяемыми строками	4	-	-
8	Основы языка Java. Наследование. Сравнение объектов	4	-	-
9	Основы языка Java. Наследование. Сравнение объектов. Запись в файловый поток	4	-	-
Семестр 6				
10	ООП в JAVA, наследование, сериализация, файловые потоки ввода-вывода. Моделирование работы со счетами	3	-	2
11	ООП в JAVA, наследование, сериализация, файловые потоки ввода-вывода. Моделирование	3	-	2

	работы склада			
12	ООП в Java. Наследование	3	-	2
13	Прикладные классы платформы J2SE, коллекции	3	-	2
14	Обработка изменяемых строк, коллекции, карты	3	-	-
15	Исследование особенностей использования коллекций и списков в языке Java	2	-	-
16	Прикладные классы J2SE. Класс Bitset	2		-
17	Наследование. Стандартные потоки ввода-вывода. Обработка исключительных ситуаций	2	--	-
18	Наследование. Стандартные потоки ввода-вывода	2	-	-
19	Файлы. Файловые потоки ввода-вывода	2	-	-
20	Файлы, операции с файлами	2	-	-
21	Многопоточные приложения	2	-	-
22	Многопоточные приложения. Синхронизация	2	-	-
23	Многопоточные приложения, файловый ввод-вывод, синхронизация	2	-	-
24	Распределенные приложения. Пакет java.net.* Протокол TCP/IP.	2		-
25	Распределенные приложения. Пакет java.net.* Протокол UDP/IP.	2	-	-
26	Распределенные приложения. Пакет java.net.*. Клиент-серверные приложения	2	-	-
27	Основы разработки распределенных приложений на базе RMI	2	-	-
Итого:		70	-	14

4.6. Самостоятельная работа студентов

№ п/п	Название темы	Вид СРС	Объем часов		
			Очная форма	Очно-заочная форма	Заочная форма
1	Введение в семейство микропроцессоров Intel x86. Архитектура 16-ти и 32-х разрядных МП.	подготовка к лабораторным работам и оформление отчетов	6	-	11
2	Регистры процессора и их назначение	подготовка к лабораторным работам и оформление отчетов	6	-	11
3	Язык Assembler, его синтаксис. Структура .com и .exe программ	подготовка к лабораторным работам и оформление отчетов	6	-	11
4	Типы данных. Директивы определения данных.	подготовка к лабораторным работам и оформление отчетов	6	-	11
5	Команды передачи данных. Способы адресации в командах пересылки.	подготовка к лабораторным работам и оформление отчетов	6	-	11
6	Арифметические команды. Формат, типы данных, особенности использования.	подготовка к лабораторным работам и оформление отчетов	6	-	11

7	Логические команды и команды перехода. Формат, типы данных, особенности использования.	подготовка к лабораторным работам и оформление отчетов	6	-	11
8	Команды обработки блоков данных. Формат, особенности использования.	подготовка к лабораторным работам и оформление отчетов	6	-	11
9	Описание и вызов процедур. Команды передачи управления	подготовка к лабораторным работам и оформление отчетов	6	-	11
10	Что такое Java? История создания	подготовка к лабораторным работам и оформление отчетов	6	-	11
11	Лексика языка	подготовка к лабораторным работам и оформление отчетов	6	-	11
12	Базовые типы данных	подготовка к лабораторным работам и оформление отчетов	6	-	11
13	Объявление классов.	подготовка к лабораторным работам и оформление отчетов	5	-	11
14	Преобразование типов.	подготовка к лабораторным работам и оформление отчетов	5	-	11
15	Объектная модель Java.	подготовка к лабораторным работам и оформление отчетов	5	-	11
16	Массивы.	подготовка к лабораторным работам и оформление отчетов	5	-	11
17	Операторы и структура кода. Исключения.	подготовка к лабораторным работам и оформление отчетов	5	-	12
18	Пакет java.lang.	подготовка к лабораторным работам и оформление отчетов	5	-	12
19	Пакет java.util.	подготовка к лабораторным работам и оформление отчетов	5	-	12
20	Пакет java.io	подготовка к лабораторным работам и оформление отчетов	5	-	12
Итого:			112	-	224

4.7. Курсовые работы/проекты.

Курсовые работы по дисциплине не предусмотрены.

5. Образовательные технологии

Преподавание дисциплины ведется с применением следующих видов образовательных технологий:

– традиционные объяснительно-иллюстративные технологии, которые обеспеч
времени;

– технологии проблемного обучения, направленные на развитие познавательной которых позволяет студентам активно усваивать знания (используются поисковые методы; постановка познавательных задач);

– технологии развивающего обучения, позволяющие ориентировать учебный процесс на потенциальные возможности студентов, их реализацию и развитие;

– технологии концентрированного обучения, суть которых состоит в создани системного изучения содержания учебных дисциплин за счет объединения занятий в тематические блоки;

– технологии модульного обучения, дающие возможность обеспечения гибкост индивидуальному учебному плану);

– технологии дифференцированного обучения, обеспечивающие возможность реализовать в культурно-образовательном пространстве университета идею создания равных возможностей для получения образования

– технологии активного (контекстного) обучения, с помощью которых осущес методы обучения) и т.д.

Максимальная эффективность педагогического процесса достигается путем конструирования оптимального комплекса педагогических технологий и (или) их элементов на личностно-ориентированной, деятельностной, диалогической основе и использования необходимых современных средств обучения.

6. Формы контроля освоения дисциплины

Текущая аттестация студентов производится в дискретные временные интервалы лектором и преподавателем(ями), ведущими практические занятия по дисциплине в следующих формах*:*

- лабораторные работы;
- защита лабораторных работ

Фонды оценочных средств, включающие вопросы к защите лабораторных работ, позволяющие оценить результаты текущей и промежуточной аттестации обучающихся по данной дисциплине, помещаются в приложении к рабочей программе в соответствии с «Положением о фонде оценочных средств».

Форма аттестации по результатам освоения дисциплины проходит в форме экзамена. Студенты, выполнившие 75% текущих и контрольных мероприятий на «отлично», а остальные 25 % на «хорошо», имеют право на получение итоговой отличной оценки.

В экзаменационную ведомость и зачетную книжку выставляются оценки по национальной шкале, приведенной в таблице.

Шкала оценивания	Характеристика знания предмета и ответов	Зачеты
отлично (5)	Студент глубоко и в полном объеме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает	зачтено

	рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.	
хорошо (4)	Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.	
удовлетворительно (3)	Студент знает только основной программный материал, допускает неточности, недостаточно четкие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.	
неудовлетворительно (2)	Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы.	не зачтено

7. Учебно-методическое и программно-информационное обеспечение дисциплины:

а) основная литература: _____

1. Лучано Рамальо, Python. К вершинам мастерства / Лучано Рамальо - М. : ДМК Пресс, 2016. - 768 с. - ISBN 978-5-97060-384-0 - Текст :

электронный // ЭБС "Консультант студента" : [сайт]. - URL : http://www.studentlibrary.ru/book/ISBN9785970603840.html (дата посещения 25.05.2023). - Режим доступа : по подписке.
2. Нортон П. Руководство Питера Нортона. Программирование на Java. В 2 кн. Кн. 1 [Текст] / П. Нортон, У. Станек. - М. : СК Пресс, 1998. - 552 с. : ил. - 681.3 - Н837
3. Хеффельфингер Д., Java EE 7 и сервер приложений GlassFish 4 / Дэвид Хеффельфингер - М. : ДМК Пресс, 2016. - 332 с. - ISBN 978-5-97060-332-1 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : http://www.studentlibrary.ru/book/ISBN9785970603321.html (дата посещения 25.05.2023). - Режим доступа : по подписке.

б) дополнительная литература: _____

1. Гарнаев А. Web-программирование на Java и JavaScript [Текст] / А. Гарнаев, С. Гарнаев. - СПб. : Питер, 2002. - 1040 с. : ил. - 681.3 - Г 207
2. Зубков С. В. Assembler для DOS, Windows и UNIX [Текст] / С. В. Зубков. - 2-е изд. испр. и доп. - М. : ДМК Пресс, 2000. - 608 с. : ил. - (Для программистов). - 681.3 - З-913 (9 экз).
3. Саммерфилд М., Python на практике / Марк Саммерфилд - М. : ДМК Пресс, 2014. - 338 с. - ISBN 978-5-97060-095-5 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : http://www.studentlibrary.ru/book/ISBN9785970600955.html (дата посещения 25.05.2023). - Режим доступа : по подписке.
4. Нортон П. Руководство Питера Нортона. Программирование на Java. В 2 кн. Кн. 2 [Текст] / П. Нортон, У. Станек. - М. : СК Пресс, 1998. - 400 с. : ил. - 681.3 - Н837
5. Юров В. Assembler [Текст] : учеб. пособие / В. Юров. - СПб. : Питер, 2002. - 624 с. : ил. - (Учебник для вузов). - 681.3 - Ю 78 (9)

в) методические указания:

1. Конспект лекций по дисциплине «Кроссплатформенное программирование» (часть 1) (для студентов, обучающихся по направлению подготовки 09.03.02 – Информационные системы и технологии / Сост.: С.С.Стоянченко . – Луганск: ЛГУ им. В. Даля, 2021. – 327 с.
2. Конспект лекций по дисциплине «Кроссплатформенное программирование» (часть 2) (для студентов, обучающихся по направлению подготовки 09.03.02 – Информационные системы и технологии / Сост.: С.С.Стоянченко . – Луганск: ЛГУ им. В. Даля, 2021. – 295 с.
3. Методические указания к лабораторным занятиям по дисциплине «Кроссплатформенное программирование» (для студентов, обучающихся по направлению подготовки 09.03.02 – Информационные системы и технологии / Сост.: С.С.Стоянченко, И.А.Рубаник .– Луганск: изд-во ЛНУ им. В. Даля, 2019. – 69 с.

4. Методические указания к выполнению контрольных работ по дисциплине «Кроссплатформенное программирование» (для студентов заочной формы обучения, обучающихся по направлению подготовки 09.03.02 «Информационные системы и технологии» / Сост.: С.С.Стоянченко – Луганск: ЛГУ им. В. Даля, 2020. – 32 с.
5. Методические указания к выполнению контрольной работы по дисциплине «Кроссплатформенное программирование» для студентов заочной формы обучения направления подготовки 09.03.02 – Информационные системы и технологии / Сост.: Ганнота Г.В., Рубаник И.А. – Луганск: ЛГУ им. В. Даля, 2022. – 34 с.

г) Интернет-ресурсы:

1. Министерство науки и высшего образования Российской Федерации – https://minobrnauki.gov.ru/
2. Федеральная служба по надзору в сфере образования и науки – http://obrnadzor.gov.ru/
3. Портал Федеральных государственных образовательных стандартов высшего образования – http://fgosvo.ru
4. Федеральный портал «Российское образование» – http://www.edu.ru/
5. Информационная система «Единое окно доступа к образовательным ресурсам» – http://window.edu.ru/
6. Федеральный центр информационно-образовательных ресурсов – http://fcior.edu.ru/

Электронные библиотечные системы и ресурсы

Электронно-библиотечная система «Консультант студента» – http://www.studentlibrary.ru/cgi-bin/mb4x

Информационный ресурс библиотеки образовательной организации

Научная библиотека имени А. Н. Коняева – http://biblio.dahluniver.ru/

8. Материально-техническое обеспечение дисциплины

Освоение дисциплины «Кроссплатформенное программирование» предполагает использование академических аудиторий, соответствующих действующим санитарным и противопожарным правилам и нормам.

Лекционные занятия: комплект электронных презентаций/слайдов; аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук).

Лабораторные работы: компьютерная аудитория, оснащенная компьютерами с установленным специализированным программным обеспечением.

Прочее: рабочее место преподавателя, оснащенное компьютером с доступом в Интернет, проектор, экран, рабочие места студентов, оснащенные компьютерами с доступом в Интернет, предназначенные для работы в электронной образовательной среде.

Программное обеспечение

Функциональное назначение	Бесплатное программное обеспечение	Ссылки
Офисный пакет	Libre Office 6.3.1	https://www.libreoffice.org/ https://ru.wikipedia.org/wiki/LibreOffice
Операционная система	UBUNTU 19.04	https://ubuntu.com/ https://ru.wikipedia.org/wiki/Ubuntu
Браузер	Firefox Mozilla	http://www.mozilla.org/ru/firefox/fx
Браузер	Opera	http://www.opera.com
Набор инструмен-тов для раз-работки программ на Java	jdk1.7	https://www.oracle.com/technetwork/java/javase/downloads/index
Интегрированная среда разработки, отладки и сборки про-граммного обеспечения	Netbeans	http://netbeans.apache.org/
Архиватор	7Zip	http://www.7-zip.org/
Ассемблер	nasm	https://www.nasm.us/
Редактор PDF	PDFCreator	http://www.pdfforge.org/pdfcreator
Аудиоплейер	VLC	http://www.videolan.org/vlc/

Паспорт
фонда оценочных средств по учебной дисциплине
«Кроссплатформенное программирование»

Перечень компетенций (элементов компетенций), формируемых в результате освоения учебной дисциплины

№ п/п	Код контролируемой компетенции	Формулировка контролируемой компетенции	Контролируемые разделы (темы) учебной дисциплины	Этапы формирования (семестр изучения)
1	ПК-03	Способность осуществлять разработку, отладку, проверку работоспособности и безопасности, расчет экономической эффективности информационных систем и технологий, модификацию программного обеспечения	Тема 1. Ассемблирование и выполнение программы Тема 2. Организация программы и определение данных Тема 3. Арифметические и логические команды Тема 4. Команды обработки строк. Тема 5. Подпрограммы. Тема 6. Основные принципы кроссплатформенного программирования. Тема 7. Основные характеристики системы программирования Java. Тема 8. Особенности языка программирования Java. Тема 9. Объектная модель языка Java. Тема 10. Стандартные пакеты Java.	5,6

Показатели и критерии оценивания компетенций, описание шкал оценивания

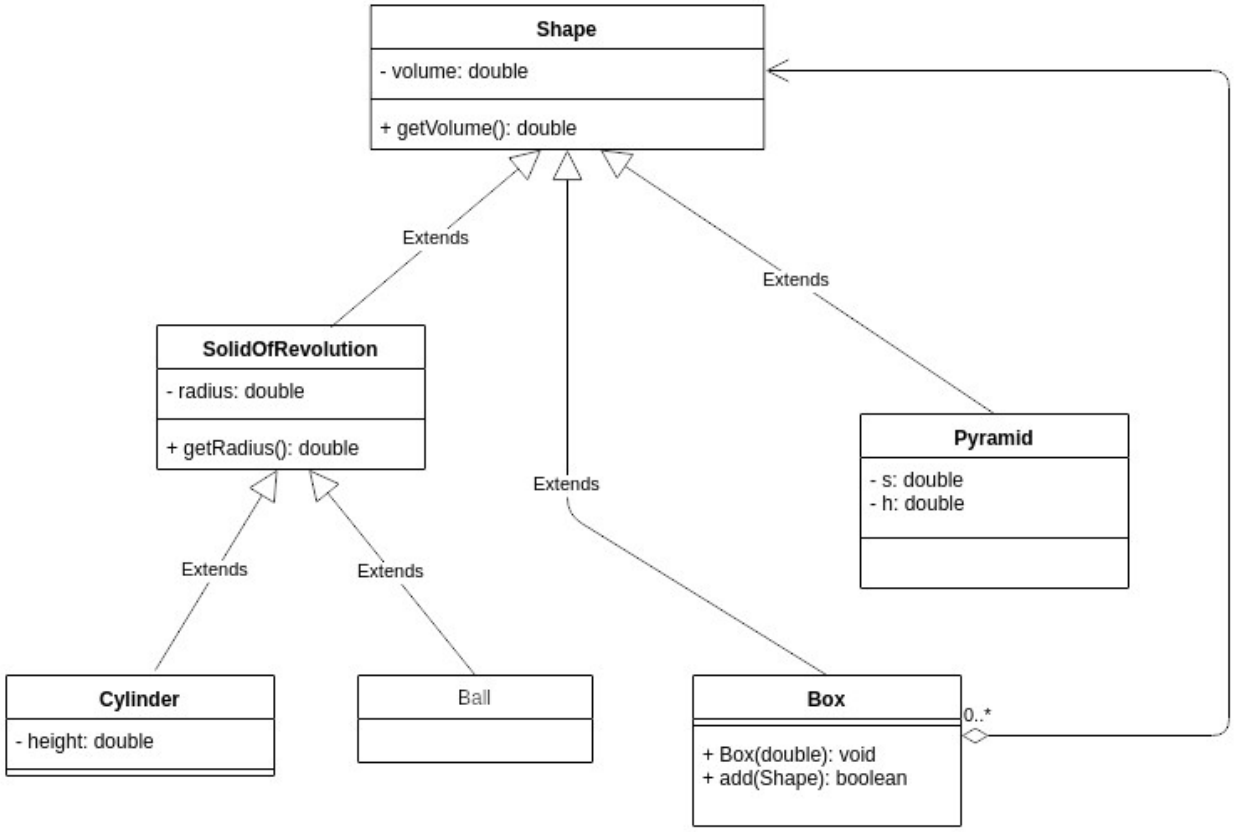
№ п/п	Код контролируемой компетенции	Показатель оценивания (знания, умения, навыки)	Контролируемые разделы (темы) учебной дисциплины (модуля), практики[1]	Наименование оценочного средства
1	ПК-03.1	Знать базовые приемы обработки информации, языки программирования, основные процедуры написания и отладки программ, угрозы безопасности	Тема 1. Тема 2. Тема 3. Тема 4. Тема 5. Тема 6. Тема 7. Тема 8.	Лабораторные работы, контрольные работы; промежуточная аттестация (экзамен)

		информационных систем и способы их предотвращения, методы расчета экономической эффективности информационных систем и технологий	Тема 9. Тема 10.	
2	ПК-03.2	Уметь обоснованно выбирать средства языка программирования, необходимые для решения поставленных задач, выявлять угрозы безопасности, проводить расчет экономической эффективности информационных систем и технологий	Тема 1. Тема 2. Тема 3. Тема 4. Тема 5. Тема 6. Тема 7. Тема 8. Тема 9. Тема 10.	Лабораторные работы, контрольные работы; промежуточная аттестация (экзамен)
3	ПК-03.3	Иметь навыки использования современных интегрированных сред разработки для создания программных продуктов, запуска процедуры резервного копирования, применения методов ценообразования для создаваемых информационных продуктов или услуг	Тема 1. Тема 2. Тема 3. Тема 4. Тема 5. Тема 6. Тема 7. Тема 8. Тема 9. Тема 10.	Лабораторные работы, контрольные работы; промежуточная аттестация (экзамен)

Фонды оценочных средств по дисциплине «Кроссплатформенное программирование»

Пример тем контрольной работы.

1. Разработать метод, который проверяет, входит ли в массив заданный элемент или нет.
2. Реализовать иерархию классов, описывающую трёхмерные фигуры.



3. Разработать метод, который на вход получает коллекцию объектов, а возвращает коллекцию уже без дубликатов.

Критерии и шкала оценивания по оценочному средству «Контрольная работа»

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Контрольная работа выполнена на высоком уровне (правильные ответы даны на 90-100% вопросов/задач)
4	Контрольная работа выполнена на среднем уровне (правильные ответы даны на 75-89% вопросов/задач)
3	Контрольная работа выполнена на низком уровне (правильные ответы даны на 50-74% вопросов/задач)
2	Контрольная работа выполнена на неудовлетворительном уровне (правильные ответы даны менее чем на 50%)

Вопросы для защиты лабораторных работ

Лабораторная работа №1. Исследование основных инструментальных утилит Java.

Цель работы: изучить назначение основных утилит Java, приобрести навыки компилирования и исполнения программ, используя утилиты javac, java, научиться создавать, модифицировать архивы Java, запускать программы из jar-файлов.

Контрольные вопросы:

1. Поясните технологию установки и настройки JDK (JAVA_HOME, PATH, CLASSPATH).

2. Дайте общую характеристику утилитам и средствам JDK.
3. Охарактеризуйте основные средства (Basic Tools) JDK: javac, java, jar, appletviewer, jdb.
4. Опишите работу с утилитами javac, java.
5. Опишите общую технологию работы с утилитой jar.
6. Подробно расскажите, как можно создать, модифицировать jar-файл.
7. Что такое точка входа (Entry Point)? Как можно задать entry point? Как запустить программу из jar-файла?

Лабораторная работа №2. Исследование методов документирования кода Java программы *Цель работы:* Изучить возможности утилиты javadoc. Приобрести навыки составления комментариев в java программах

Контрольные вопросы:

1. Для каких целей предназначена утилита Javadoc?
2. Укажите особенности создания Javadoc комментариев
3. Какие Javadoc дескрипторы Вы знаете?
4. Опишите особенности создания комментариев класса.
5. Опишите особенности создания комментариев методов.
6. Опишите особенности создания комментариев параметров.

Лабораторная работа №3. Основы языка Java, массивы, примитивные типы, объявление классов

Цель работы: ознакомиться о с основными синтаксическими конструкциями языка Java; изучить структуры консольного приложения на языке Java; приобрести навыки работы со стандартными потоками ввода/вывода; приобрести навыки работы с командной строкой.

Контрольные вопросы:

1. К какому виду исключительных ситуаций относится реализованная в пункте 5.
2. Какие классы вы использовали для проверки типа элемента последовательности?
3. Почему все методы, реализующие операции, указанные в задании объявляются как статические?
4. Что такое расширяющее преобразование типов?
5. Что такое примитивный тип в Java?
6. Проанализируйте следующий код метода. Что произойдет в результате его вызова?

Лабораторная работа №4. Основы языка Java, перегрузка и перекрытие методов, наследование

Цель работы: изучить принципов ООП в языке Java, использования перегрузки и перекрытия методов; получить представления о практическом назначении и использовании модификаторов объявлений классов, методов и полей; приобрести навыки проектирования и реализации иерархии классов; изучить методы обработки исключительных ситуациях; приобрести навыки описания собственных исключительных ситуациях; приобрести навыки использования класса java.lang.Math для выполнения математических расчетов.

Контрольные вопросы:

1. Пусть объявлен следующий класс:

```
package javaapplication2;
private class Sample {
  private static int value;
  static{
    value=1;
  }
  private int n;
```

```
  Sample(){
    value++;
    n=value%2;
  }
  Sample(int n){
    this();
    n=this.n;
  }
  public int getN(){
    return(n<10?n=value++:n);
  }
}
```

Что произойдет в результате выполнения следующего метода?:

```
/* **/
public void test(){
  for (int x=0;x<10;x++) {
    System.out.println(new Sample(x));
  }
}
```

2. Какие из перечисленных объявлений полей класса являются недопустимыми и допустимыми и почему?
    1. public final int 1v=2;
    2. public final int 'xxx'=2;
    3. public final int \u1000=1000;
    4. public final static volatile boolean bool = true;
    5. static volatile Boolean v2=Boolean.parseBoolean("true");
3. Дайте определения понятиям статический метод и статическое поле класса.
4. Что такое явное и неявное приведение типов?
5. Какие методы классов-оболочек над примитивными типами используются для получения значения примитивного типа из его строкового представления.
6. Для какого примитивного типа не существует класса-оболочки?

Лабораторная работа №5. Основы языка Java. Перегрузка и перекрытие методов, наследование. Классы-оболочки

Цель работы: изучить принципы ООП в языке Java, использование перегрузки и перекрытия методов; получить представление о практическом назначении и использовании модификаторов объявлений классов, методов и полей; приобрести навыки проектирования и реализации иерархии классов; изучить методы обработки исключительных ситуаций; приобрести навыки описания собственных исключительных ситуаций; приобрести навыки использования класс java.lang.Math для выполнения математических расчетов; приобрести навыки выбора оптимальной структуры библиотеки классов для решения поставленных задач.

Контрольные вопросы:

1. Что произойдет в результате компиляции и выполнения следующего фрагмента исходного кода программы?

```
public class CustomerTwo {
  public static void main (String[] args) {
    Pizza favoritePizza = new Pizza();
    System.out.println("Meat on pizza before baking: " +
         favoritePizza.meat);
    bake(favoritePizza);
```

```
    System.out.println("Meat on pizza after baking: " +
         favoritePizza.meat);
  }
  public static void bake(Pizza pizzaToBeBaked) {
    pizzaToBeBaked.meat = "chicken";
    pizzaToBeBaked = null;
  }
}
class Pizza {
   String meat = "beef";
}
```

2. Каким образом реализуется наследование в Java?
3. Каким образом класс-потомок может обратиться полям и методам суперкласса?
4. В какой последовательности осуществляется вызов конструкторов классов, являющихся суперклассами для данного класса?

Лабораторная работа №6. Основы языка Java. Наследование, тригонометрические функции класса **Math**

Цель работы: изучить принципы ООП в языке Java; получить представление о практическом назначении и использовании модификаторов объявлений классов, методов и полей; приобрести навыки проектирования и реализации иерархии классов; изучить методы обработки исключительных ситуациях; приобрести навыки описания собственных исключительных ситуаций; приобрести навыки использования класса java.lang.Math для выполнения математических расчетов; приобрести навыки выбора оптимальной структуры библиотеки классов для решения поставленных задач.

Контрольные вопросы:

1. Что произойдет в результате компиляции и выполнения следующего фрагмента исходного кода?

```
class MultiArrays {
public static void main(String[] args) {
  int[][] mXnArray = {{16,  7, 12},
    { 9, 20, 18},
    {14, 11,  5},
    { 8,  5, 10}};
  int min = mXnArray[0][0];
  for (int i = 0; i < mXnArray.length; ++i)
    for (int j = 0;j<mXnArray[i].length;++j)
      min = Math.min(min,mXnArray[i][j]);
  System.out.println("Минимальное значение: "
     + min);
  }
}
```

2. Для чего предназначена секция импорта?
3. Какие существуют ограничения и правила именования классов?
4. Какие существуют способы инициализации массивов?

Лабораторная работа №7. Основы языка Java. Работа с изменяемыми и неизменяемыми строками

Цель работы: приобретение навыков выполнения типовых операций над строками и буферами строк; изучение принципиальных отличий между классами String и StringBuffer; приобретение навыков принятия решения по оптимальному выбору между

неизменяемыми и изменяемыми строками для решения поставленных задач; приобретение навыков обработки вводимых данных с консоли и представления их в требуемом виде.

Контрольные вопросы:

1. В приведенном ниже исходном тексте метода, определите сколько ссылок на объект, создаваемый в строке (1) будет содержаться в момент выполнения строки (2).

```
public int getSomeValue(){
  Integer a=1; //(1)
  Integer b=new Integer(a);
  Integer c=a;
  Integer e=b;
  b=a;
  a=new Integer(100);
  return a+b+c+e; //(2)
}
```

2. Какая из строк приведенного ниже метода может выбросить исключительную ситуацию NumberFormatException?

```
public void parseNumbersFromString(String numb){
  java.util.List<Integer> numbers=
      new java.util.LinkedList<Integer>();
  java.util.List<Integer> separators=
      new java.util.LinkedList<Integer>();
  int i=0;
  while ((i!=-1)&&(i<numb.length())) {
    int indsep=numb.indexOf(',', i);
    if (indsep!=-1) separators.add(indsep);
    i=indsep+1;
  }
  separators.add(0, 0);
  separators.add(numb.length());
  Iterator it=separators.iterator();
  while (it.hasNext()) {
    numbers.add(new Integer(numb.substring(
    ((Integer)it.next()).intValue()+1,
    ((Integer)it.next())-1)));
  }
}
```

3. В чем состоит принципиальное отличие между классами String и StringBuffer?

4. Какие типы исключительных ситуаций вы знаете? К исключительным ситуациям какого типа относится ja va.lang.Error?;

5. Допустимо ли преобразование объекта класса Integer к типу String?

6. Какие методы предусмотрены в классе Integer для представления числовых значений в различных системах счисления?

Лабораторная работа №8. Основы языка Java. Наследование. Сравнение объектов

Цель работы: изучить механизмы сравнения объектов в языке Java с использованием интерфейсов Comparable и Comparator; приобрести навыки работы с файловыми потоками ввода-вывода; приобрести навыки реализации библиотек классов со сложной структурой; приобрести навыки работы с классами-оболочками над простыми типами данных.

Контрольные вопросы:

1. Что произойдет в результате выполнения следующего кода:

```
public class Sample {
```

```
  public static void main(String[] args) {
    String[][][] arr ={{ {}, null },
     { {"1", ”2” }, {"1", null,"3" }},
     {},
     { {"1",null }} };
    System.out.println(arr.length +
      arr[1][2].length);
  }
}
```

2. Какие из нижеперечисленных объявлений полей класса являются допустимыми?

а) `int morrow=1;`
б) `public transient static x=new String();`
в) `java.lang.Integer.MAX_INTEGER val=new Integer(100);`
г) `java.math.BigDecimal dbm=new java.math. BigDecimal();`

3. Для чего предназначена перегрузка методов класса?
4. Могут ли перекрываться статические методы класса в классах потомках?
5. Какие основные задачи решает класс File?
6. Для чего предназначен интерфейс Map?
7. Что такое дискретный аргумент?
8. Назовите способы определения дискретного аргумента.
9. Дайте определение следующим понятиям: скаляр, вектор, матрица. Для чего может быть использована переменная ORIGIN?
10. Как можно обратиться к столбцу матрицы?
11. Как следует обратиться к одному из элементов матрицы?
12. Перечислите известные вам векторные и матричные функции и раскройте их назначение.
13. Приведите синтаксис функции if. В чем ее назначение?

Лабораторная работа №9. Основы языка Java. Наследование. Сравнение объектов. Запись в файловый поток

Цель работы: приобретение навыков проектирования и разработки библиотеки классов; приобретение навыков реализации механизмов сравнения объектов в соответствии с заданным порядком; изучение основных классов и интерфейсов, а также их методов и свойств, для организации файлового потока ввода-вывода;

Контрольные вопросы:

1. Что произойдет в результате выполнения следующего программного кода?

```
public class Sample {
  static int a;
  int b;
  public Q275d() {
    int c;
    c = a;
    a++ ;
    b += c;
  }
  public static void main(String[] args) {
    new Sample();
  }
}
```

2. Какие из приведенных ниже объявлений классов являются правильными?

```
1.  public static class Sample{...};
2.  interface LowInterface{...};
3.  protected class Impl extends LowInterface{...};
4.  abstract class A {...}
5.  final abstract class B{...};
6.  final class X{...};
```

3. Можно ли производить запись в файл с использованием класса OutputStreamWriter?
4. Для чего предназначены буферизованные потоки чтения и записи?
5. Что понимается под блоком инициализации класса?
6. Каким образом функционирует «сборщик мусора» виртуальной машины Java?

Лабораторная работа №10. ООП в JAVA, наследование, сериализация, файловые потоки ввода-вывода

Цель работы: приобрести навыки навыков проектирования и разработки библиотек классов; изучить и приобрести навыки использования механизма сериализации объектов.

Контрольные вопросы:

1. В каком из приведенных объявлений методов содержится ошибка:
   а) **`public final someMethod(int x; final long a);`**
   б) **`public abstract final someAnotherMethod(int x; long b);`**
   в) **`public synchronized void sMethod();`**
2. Что произойдет в результате выполнения следующего кода:

```
public class Sample {
   static void test(int i) {
       int j = i/2;
       int k = i >>> 1;
       assert j == k : i;
   }
   public static void main(String[] args) {
      test(0);
      test(2);
      test(-2);
      test(1001);
      test(-1001);
   }
}
```

3. Для чего предназначен интерфейс Serializiable?
4. Какие виды наследования поддерживаются в языке Java?
5. Для чего предназначены итераторы?
6. Какие существуют способы задания констант в Java?

Лабораторная работа №11. ООП в JAVA, наследование, сериализация, файловые потоки ввода-вывода

Цель работы: изучить и приобрести навыки использования механизма сериализации объектов; приобрести навыки проектирования и разработки библиотек классов;

Контрольные вопросы:

1. Что произойдет в результате выполнения следующего кода программы:

```
class MyException extends Exception {}
public class Sample {
```

```
  public void foo() {
    try {
      bar();
    } finally {
       baz();
    } catch (MyException e) {}
  }
  public void bar() throws MyException {
    throw new MyException();
  }
  public void baz() throws RuntimeException {
    throw new RuntimeException();
  }
}
```

2. Какие классы в Java предназначены для работы с файлами?
3. Что подразумевает сериализация объектов?
4. Какие существуют средства поддержки механизма сериализаии в Java?
5. Для чего предназначен класс BufferedReader?

Лабораторная работа №12. ООП в Java. Наследование

Цель работы: изучить принципы ООП в языке Java, использование перегрузки и перекрытие методов; получить представление о практическом назначении и использовании модификаторов объявлений классов, методов и полей; приобрести навыки проектирования и реализации иерархии классов; изучить методы обработки исключительных ситуациях; приобрести навыки описания собственных исключительных ситуаций; приобрести навыки использования класса java.lang.Math для выполнения математических расчетов; приобрести навыки навыки выбора оптимальной структуры библиотеки классов для решения поставленных задач.

Контрольные вопросы:

1. Что произойдет в результате компиляции и выполнения следующего исходного кода программы:

```
public class TestClass{
  private static double a ;
  static{
    a=System.getCurrentTime();
    a=a++;
  }
  public long b;
    TestClass{
    b=a;
  }
}
```

2. Какое из приведенных объявлений констант не является верным:
   1. **`public final static int k=0;`**
   2. **`public final static long x=System.getCurrentTime();`**
   3. **`public final static long c=100000l;`**
   4. **`public final static volatile int z=2;`**
3. Какие основные методы класса **String** вам известны?
4. Какие существуют правила перекрытия методов в Java?
5. Каким образом используются диагностические утверждения?

Лабораторная работа №13. Прикладные классы платформы J2SE, коллекции

Цель работы: изучить возможности платформы J2SE для решения прикладных задач; изучить методы получения и обработки значений вида дата/время; приобрести навыки применения региональных настроек и локализации приложений; приобрести навыки выполнения операций по чтению/записи в файловые потоки ввода/вывода; изучить нестандартный порядок для организации множеств.

Контрольные вопросы:

1. Что произойдет в результате компиляции и выполнения следующего исходного кода?

```
public class Average8 {
  public static void main(String[] args) {
    try {                                          // (1)
      printAverage(100, 0);                        // (2)
    } catch (IntegerDivisionByZero idbze) {        // (3)
    idbze.printStackTrace();
    System.out.println("Исключение отловлено в: "+
     " main().");
     } finally { //(4)
       System.out.println("Finally выполнено в " +
      "main().”);
    }
    System.out.println("Exit main().");          // (5)
  }
  public static void printAverage(int totalSum,
     int totalNumber)
                 throws IntegerDivisionByZero { // (6)
     int average = computeAverage(totalSum
        , totalNumber);
     System.out.println("Average = ” + totalSum +
       ” / ” + totalNumber + ” = ” + average);
    System.out.println("Выход printAverage()."); // (7)
  }
  public static int computeAverage(int sum, int number)
     throws IntegerDivisionByZero {      // (8)
   System.out.println("Вычисление среднего.”);
   if (number == 0)                             // (9)
     throw new IntegerDivisionByZero("Деление на ноль”);
   return sum/number;                          // (10)
  }
}
```

2. Каким образом в Java можно получить значение текущей времени и даты?
3. Для чего предназначен класс Calendar?
4. Какой класс позволяет получить текущую дату в соответствии с региональными параметрами?

Лабораторная работа №14. Обработка изменяемых строк, коллекции, карты.

Цель работы: приобрести навыки выполнения типовых операций над строками и строковыми буферами; приобрести навыки разработки собственных классов, реализующих стандартные интерфейсы платформы J2SE; изучить методы организации сравнения объектов с использованием нестандартных правил упорядочивания; приобрести навыки анализа производительности приложений в рамках платформы J2SE;

Контрольные вопросы:

1. Что будет выведено на экран в результате компиляции и выполнения следующего кода:

```
class A{}
class B extends A{}
class C extends A{}
class D extends B{}
class Main {
  public static void main (String[] args}{
    A a=new D();
    B b=new B();
    D d=new D();
    C c=new A();
    System.out.println (( a instanceof A)
     ||(d instance of A)
     ||c instanceof A)
     ||(d instanceof C));
  }
}
```

2. Что будет выведено на экран в результате выполнения следующего программного кода:

```
public class Q8499 {
  public static void main(String[] args) {
    double d = -2.9;
    int i = (int) d;
    i *= (int) Math.ceil(d);
    i *= (int) Math.abs(d);
    System.out.prmtln(i);
  }
}
```

3. Какой метод организации данных использует класс **Hashmap**?
4. Опишите иерархию классов и интерфейсов коллекций в платформе J2EE?
5. Для чего предназначены цепочки конструкторов?
6. Какие методы существуют для получения значения системного времени в J2SE?

Лабораторная работа №15. Исследование особенностей использования коллекций и списков в языке Java

Цель работы: изучить методы и свойства базовых интерфейсов и классов, представляющих коллекции, списки и карты; приобрести навыки реализации стандартных интерфейсов платформы J2SE; приобрести навыки использования вложенных интерфейсов и классов; приобрести навыки выполнения операций со стандартными потоками ввода-вывода; изучить методы сравнения объектов.

Контрольные вопросы:

1. Верно ли следующее объявление интерфейса:

```
public interface A{
  public final static int A=10;
  public abstract void methodB();
  public int method C();
  public static int methodA(){
    return A;
  }
}
```

2. Что произойдет при компиляции и выполнении следующего исходного кода:

```
public class Sample {
  int a;
  int b;
  public void f() {
    a = 0;
    b = 0;
    int[] c = { 0 };
    g(b, c);
    System.out.println(a + ” ” + b + ” ” + c[0] + ” ”);
  }
  public void g(int b, int[] c) {
    a = 1;
    b = 1;
    c[0] = 1;
  }
  public static void main(String[] args) {
    Sample obj = new Sample(); obj.f();
  }
}
```

3. Для чего предназначены интерфейсы Comparator и Comparable? Реализует ли интерфейс Comparable класс Boolean?
4. Для чего предназначены неизменяемые оболочки коллекций?
5. Какие существуют способы инициализации массивов?
6. Для чего предназначен интерфейс Map?

Лабораторная работа №16. Прикладные классы J2SE. Класс Bitset

Цель работы: приобрести навыки использования прикладных классов J2SE на примере класса, представляющего абстракцию последовательности битов; приобрести навыки выполнения операций со стандартными потоками ввода-вывода.

Контрольные вопросы:

1. Что произойдет при попытке компиляции и запуска следующего фрагмента?

```
public class Q275d {
  static int a;
  int b;
  public Q275d() {
    int c;
    c = a;
    a++;
    b += c;
  }
  public static void main(String[] args) {
    new Q275d();
  }
}
```

2. Что произойдет в результате компиляции и выполнения следующего исходного кода?

```
public abstract class Main {
  class TestOutput{
    LinkedList numbers=new LinkedList();
```

```
    TestOutput(long count ) {
      for (long i=0;i<count;i++){
        numbers.add(Math.random());
    public void addNumbers(long count){
     for (long i=0;i<count;i++)
       numbers.add(Math.random());
    }
    public void writeToFile(String fileToWrite)
           throws IOException {
      java.io.FileWriter fw =
          new java.io.FileWriter(fileToWrite);
      try{
          Iterator i=numbers.iterator();
          while (i.hasNext()){
            Double d=(Double)i.next();
            fw.write(d.toString());
             fw.close();
             i.remove();
      }
    } catch (IOException e){
       throw e;
    }
    finally {
      fw.close();
    }
// fw.write();
  }
  public static void main(String[] args) {
// TODO code application logic here
    try{
     (new TestOutput(Long.parseLong
     (args[1]))).writeToFile(args[0]);
   }
   catch (IOException e){
     System.err.print(e);
   }
  }
}
```

3. Какие виды вложенных классов и интерфейсов вы знаете?
4. В чем заключаются принципиальные различия между абстрактными классами и интерфейсами?
5. Какие модификаторы доступа допустимы для методов класса?

Лабораторная работа №17. Наследование. Стандартные потоки ввода-вывода

Цель работы: изучить принципы объектного-программирования и их реализации в языке Java; приобрести навыки выполнения операций со стандартными потоками ввода-вывода; приобрести навыки работы с классами-оболочками над примитивными типами; приобрести навыки обработки исключительных ситуаций и реализации собственных классов исключительных ситуаций.

Контрольные вопросы:

1. Метод какого класса будет вызван в результате выполнения следующего исходного кода:

```
public class A{};
public class B extends A{
  public void doSomething(){...}
};
public class C extends B{
  public void doSomething(){...}
};
public class Test{
  public void main (String[] s) {
    C c=new C();
    A a=(A)(B)c;
    a.doSomething();
  }
}
```

2. Какой класс удобнее всего использовать для хранения пар типа «ключ-значение»?

3. Существуют ли методы, позволяющие сохранять текст в файл с выбранной кодировкой?

4. Какие кодировки поддерживаются для объектов класс **String**?

Лабораторная работа № 18. Наследование. Стандартные потоки ввода-вывода

Цель работы: изучить принципы в объектно-ориентированного программирования и их реализации в языке Java; приобрести навыки работы с буферизованными потоками ввода-вывода; приобрести навыки выполнения операций со стандартными потоками ввода-вывода; приобрести навыки работы с классами-оболочками над примитивными типами; приобрести навыки обработки исключительных ситуаций и реализации собственных классов исключительных ситуаций;

Контрольные вопросы:

1. Что произойдет в результате компиляции и выполнения следующего исходного кода?

```
class SuperclassA {
  public SuperclassA() {
    System.out.println(”КоНСТруКТор SuperclassA”);
  }
}
class SubclassB extends SuperclassA {
  SubclassB() {
    this(3);
    System.out.println(”Конструктор по-умолчанию SubclassB”);
  }
  SubclassB(int i) {
  System.out.println(«Перегруженный конструктор в SubclassB”);
  value = i;
}
{
  System.out.println(”Блок инициализации SubclassB”);
  value = 2;
}
int value = initializerExpression();
  private int initializerExpression() {
```

```
  System.out.println(”Инициализатор поля класса в SubclassB”);
  }
}
public class ObjectConstruction {
  public static void main(String[] args) {
    SubclassB objRef = new SubclassB(); //   (8)
    System.out.println(”value: ”+ objRef.value);
  }
}
```

2. Какое из объявлений переменных содержит ошибку:
   a. `private transient int a=100;`
   b. `private statict volatile int v=1000;`
   c. `final public String l= «aaaa»+ «bbbb»;`
3. Что подразумевает понятие рефлексии в Java?
4. Для чего предназначен модификатор **`transient`**?
5. Для каких целей предназначен метод **`sleep`** у потока?

Лабораторная работа № 19. Файлы. Файловые потоки ввода-вывода

Цель работы: приобрести навыки выполнения операций с файлами средствами платформы J2SE; приобрести навыки обработки параметров командной строки;

Контрольные вопросы:

1. Что произойдет в результате компиляции и выполнения следующего исходного кода?

```
public class Q28fd {
  public static void main(String[] args) {
  int counter = 0;
  11:
    for (int i=0; i<10; i++) {
  l2:      int j = 0;
    while (j++ < 10) {
      if (j > i) break l2;
      if (j == i) {
       counter++;
       continue l1;
     }
    }
   }
  System.out.println(counter);
 }
}
```

2. Правильно ли следующее объявление класса?

```
public abstract class N implements C{
private long a;
final abstract void doNothing();
}
```

3. Какие существуют классы расширяющие класс InputStream и каково их назначение?
4. Какие классы предназначены для преобразования из байтового потока вывода в символьный поток вывода?
5. Каким образом можно получить данные о размере файла?

Лабораторная работа № 20. Файлы, операции с файлами

Цель работы: получить навыки выполнения операций с файлами (создание, чтение, запись, поиск в файле) средствами платформы J2SE; изучить особенности работы с атрибутами файлов; приобрести навыки обработки параметров командной строки;

Контрольные вопросы:

1. Что произойдет в результате компиляции и выполнения следующего исходного кода?

```
class Base {
  int i;
  Base() {
   add(1);
  }
}
class Extension extends Base {
  Extension() {
    add(2);
  }
  void add(int v) {
    i += v*2;
  }
}
public class Qd073 {
  public static void main(String[] args) {
    bogo(new Extension());
  }
  static void bogo(Base b) {
    b.add(8);
    b.print();
  }
}
```

2. Что произойдет в результате компиляции и выполнения следующего исходного кода?

```
public class Q03e4 {
  public static void main(String[] args) {
    String space = " ";
    String composite = space + "hello" + space + space;
    composite.concat("world");
    String trimmed = composite.trim();
    System.out.println(trimmed.length());
  }
}
```

3. На основе какой структуры данных реализован класс TreeSet?
4. Для каких целей используется модификатор final в объявлении классов?
5. Какова область видимости у классов без явно указанного модификатора доступа?

Лабораторная работа № 21. Многопоточные приложения

Цель работы: изучить принципы реализации многопоточных приложений; приобрести навыки разработки в разработки многопоточных приложений; изучить особенности синхронизации операций над разделяемыми объектами в условиях многопоточности;

Контрольные вопросы:

1. Что произойдет в результате компиляции и выполнения следующего исходного кода?

```
public class Vertical {
  private int alt;
  public synchronized void up() {
    ++alt;
  }
  public void down() {
    --alt;
  }
  public synchronized void jump() {
    int a = alt;
    up();
    down();
    assert(a == alt);
  }
}
```

2. Какая последовательность вызова методов классов пакета **java.io.** при осуществлении чтения из файлового потока?
3. Объясните понятие и назначение финализации объектов.
4. Приведите примеры использования класса **ArrayLis**t.

Лабораторная работа № 22. Многопоточные приложения. Синхронизация

Цель работы: изучить принципы реализации многопоточных приложений; приобрести навыки разработки многопоточных приложений; изучить особенности синхронизации операций над разделяемыми объектами в условиях многопоточности; изучить особенности секций и объектов синхронизации

Контрольные вопросы:

1. Что произойдет в результате компиляции и выполнения следующего исходного кода?

```
class Base {
  int i;
  Base() {
    add(1);
  }
  void add(int v) {
    i += v;
  }
}
class Extension extends Base {
  Extension() {
    add(2);
  }
  void add(int v) {
    i += v*2;
  }
}
public class Qd073 {
  public static void main(String[] args) {
    bogo(new Extension());
  }
  static void bogo(Base b) {
```

```
   b.add(8);
   b.print();
  }
}
```

2. Что произойдет в результате компиляции и выполнения следующего исходного кода:

```
public abstract class Main {
  class TestOutput{
    LinkedList numbers=new LinkedList();
    TestOutput(long count ) {
      for (long i=0;i<count;i++){
       numbers.add(Math.random());
      }
    }
    public void addNumbers(long count){
      for (long i=0;i<count;i++)
        numbers.add(Math.random());
    }
    public void writeToFile(String fileToWrite)
       throws IOException {
     java.io.FileWriter fw=new java.io.FileWriter(fileToWrite);
     try{
      Iterator i=numbers.iterator();
      while (i.hasNext()){
        Double d=(Double)i.next();
        fw.write(d.toString());
        i.remove();
      } catch (IOException e){
         throw e;
      }
      finally{
        fw.close();
      }
    }
  }
  public static void main(String[] args) {
    try{
      (new TestOutput(Long.parseLong(args[1])))
              .writeToFile(args[0]);
    }
    catch (IOException e) {
      System.err.print(e);
    }
    }
  }
}
```

3. Каким образом ведут себя главный поток приложения и порожденные им потоки, после выполнения всех операций?
4. В чем заключаются отличия между классами TreeSet и HashSet?
5. На основе какой структуры данных реализован класс Hashmap?

Лабораторная работа № 23. Многопоточные приложения, файловый ввод-вывод, синхронизация

Цель работы: изучить принципы реализации многопоточных приложений; приобрести навыки разработки многопоточных приложений; изучить особенности синхронизации операций над разделяемыми объектами в условиях многопоточности;

Контрольные вопросы:

1. Какие утверждения верны для нижеприведенного исходного кода?

```
class Counter {
    int v = 0;
    synchronized void inc() {
       v++;
    }
    synchronized void dec() {
       v--;
    }
}
public class Q7ed5 {
    Counter i;
    Counter j;
    Counter k;
    public synchronized void a() {
        i.inc();
        System.out.println("a");
        i.dec();
    }
    public synchronized void b() {
        i.inc();
        j.inc();
        k.inc();
        System.out.println("b");
        i.dec();
        j.dec();
        k.dec();
    }
    public void c() {
        k.inc();
        System.out.println("c");
        k.dec();
    }
}
```

   а) i.v всегда равен 0 или 1;
   б) j.v всегда равен 0 или 1;
   в) k.v всегда равен 0 или 1;
   г) j.v всегда больше или равен k.v в любой момент времени;
   д) k.v всегда больше или равен k.v в любой момент времени;

2. Какие средства в Java существуют для представления чисел в различных системах счисления?
3. Какие классы платформы J2SE существуют для поддержки многопоточности?
4. Каким образом обеспечивается синхронизация при работе с объектами-коллекциями?

Лабораторная работа № 24. Распределенные приложения. Пакет java.net.*.

Цель работы: приобретение навыков разработки распределенных клиент-серверных приложений с поддержкой передачи данных по протоколу TCP/IP.

Контрольные вопросы:

1. Что произойдет в результате компиляции и выполнения следующего исходного кода?

```
public class Sample {
  public static void main(String[] args) {
    LinkedList lla = new LinkedList();
    LinkedList llb = new LinkedList();
    assert lla.size() == llb.size() : "пустой";
    la.add(”Hello”);
    assert lla.size() == 1 : "размер";
    llb.add("Hello");
    assert llb.contains("Hello") : "содежит";
    assert lla.get(0).equals(llb.get(0)) : "’элемент";
    assert lla.equals(llb) : "коллекции";
  }
}
```

2. Какой общий вид прототипа объявления метода класса?
3. Какой механизм используется для передачи параметров в методы класса?

Каким образом можно получить из объекта класса **java.lang.LinkedList** массив хранимых в нем объектов?

Лабораторная работа № 25. Распределенные приложения. Пакет java.net.*.

Цель работы: изучить принципы передачи данных в сети с использованием протокола UDP/IP, а также классов и интерфейсов платформы J2SE, предназначенных для поддержки этого протокола.

Контрольные вопросы:

1. Какое утверждение верно относительно следующего исходного кода программы?

```
public class SampleClass extends Thread{
  static Object lock1=new Object();
  static Object lock2=new Object();
  static volatile int i1,i2,j1,j2,k1,k2;
  public void run(){
    while (true) {
      doIt();
      check();
    }
  }
  void doIt(){
    synchronized(lock1){
      i1++;
    }
    j1++;
    synchronized(lock2){
      k1++;
      k2++;
    }
    j2++;
    synchronized(lock1){
```

```
      i2++;
    }
  }
  void check(){
    if (i1!=0)
       System.out.println("i");
    if (j1!!=j2)
       System.out.println("j");
    if (k1 !=k2)
       System.out.println("k");
  }
  public static void main(String[] args){
    new SampleClass().start();
    new SampleClass().start();
  }
}
```

Выберите правильный ответ:

а) программа не скомпилируется;

б) нельзя сказать определенно, будут ли напечатаны во время выполнения буквы i, j, k;

в) определенно можно сказать, что ни одна из букв i, j, k не будет напечатана во время выполнения;

г) определенно можно сказать, что буквы i и k никогда не будут напечатаны во время выполнения;

д) определенно можно сказать, что буква k никогда не будет напечатана во время выполнения;

2. Для чего предназначены и как реализуются статические секции инициализации?
3. Какие модификаторы доступа доступны при объявлении метода класса?
4. Как организовать перенаправление стандартного потока вывода в файл?

Лабораторная работа № 26. Распределенные приложения. Пакет java.net.*.

Целью работы: получение навыков разработки распределенных клиент-серверных приложений с поддержкой передачи данных по протоколу TCP/IP

Контрольные вопросы:

1. Что произойдет в результате компиляции и выполнения следующего исходного кода программы?

```
public class Nesting {
  public static void main(String[] args){
    B.C obj=new B().new C();
  }
}
class A{
  int val;
  A(int v){
    val=v;
  }
}
class B extends A{
  int val=1;
  B(){
    super(2);
  }
```

```
  class C extends A {
   int val=3;
   C(){
    super(4);
    System.out.println(B.this.val);
    System.out.println(C.this.val);
    System.out.println(super.val);
   }
 }
}
```

2. Какое утверждение относительного приведенного ниже исходного кода верно?

```
public class Joining{
  static Thread createThread(final int i
     , final Thread t1){
  Thread t2=new Thread(){
    public void run(){
      System.out.println(i+1);
      try{
        t1.join();
      } catch (InterruptedException e){          }
      System.out.println(i+2);
    }
   };
    System.out.println(i+3);
    t2.start();
    System.out.println(i+4);
    return t2;
  }
  public static void main(String[] args){
    createThread(10,createThread(20,
    Thread.currentThread()));
  }
}
```

Выберите правильные ответы:

а) Первое число, которое будет напечатано, это 13;
б) Число 14 будет напечатано прежде, чем число 22;
в) Число 24 будет напечатано прежде, чем число 21;
г) Последнее число, которое будет напечатано, это 12;
д) Число 11 будет напечатано прежде, чем число 23;

3. Для каких целей предназначены классы java.io.DataInput- Stream, java.io.DataOutputStream?
4. Каким способом можно считать из стандартного потока чтения строку данных?
5. Какие состояния потока вы знаете?

Лабораторная работа № 27. Основы разработки распределенных приложений на базе RMI

Цель работы: ознакомиться с технологией RMI, изучить определение и реализацию удаленного интерфейса, приобрести навыки вызова удаленного объекта; изучить методы обеспечения безопасности в распределенных приложениях; приобрести навыки использования файловых потоков ввода/вывода.

Контрольные вопросы:

1. Что произойдет при попытке компиляции и выполнения следующего

исходного кода

```
abstract class AbstractClass{
  private int value;
  public abstract void doAction();
  public int getValue(){
    return this.value;
  }
  public void setValue(final int value){
    if (value>=0)
      this.value=value;
  }
}
class AbstractClassRealization extends AbstractClass{
  public void doAction() {
    value=new Integer((new Double(Math.random()))
      .intValue());
  }
}
```

2. Какая из отмеченных строк в следующем коде может быть раскомментированна, так что код будет по-прежнему компилироваться без ошибок?

```
class SampleClass{
// int width =14;        /* Строка А*/
{
// area=width*height; /*Строка Б*/
}
int width=37;
{
//height=11           /* Строка В*/
}
int height,area;
//area =width*height;    /* Строка Г */
{
//int width=15;         /* Строка Д*/
area=100;
}
};
```

3. В чем заключается сущность технологии RMI?
4. В каких случаях необходимо применять синхронизацию?
5. Каким образом происходит сравнение объектов, реализующих интерфейс Comparable?

Критерии и шкала оценивания по оценочному средству «Лабораторная работа»

Шкала оценивания (интервал баллов)[2]	Критерий оценивания
5	Лабораторная работа выполнена самостоятельно на высоком уровне и в полном объеме, отчет оформлен в соответствии с требованиями, сделаны правильные выводы по проведенным экспериментам.
4	Лабораторная работа выполнена самостоятельно на среднем уровне и в полном объеме, отчет оформлен с незначительными

	отклонениями от требований, допущены незначительные неточности в выводах по проведенным экспериментам
3	Лабораторная работа выполнена на низком уровне и не полностью, отчет оформлен с отклонениями от требований, выводы по экспериментам сделаны не в полном объеме.
2	Лабораторная работа не выполнена, отчет не оформлен, или представленный отчет не соответствует варианту задания.

Оценочные средства для промежуточной аттестации (зачет)

1. Какая разница между объектом и экземпляром класса?
2. Что входит в класс Java?
3. Что такое конструктор класса?
4. Какая операция выделяет оперативную память для объекта?
5. Что такое суперкласс и подкласс?
6. Как реализуется полиморфизм в Java?
7. Для чего нужны статические поля и методы класса?
8. Какую роль играют абстрактные методы и классы?
9. Можно ли записать конструктор в абстрактном классе?
10. Почему метод main ( ) должен быть статическим?
11. Почему метод main ( ) должен быть открытым?
12. Синтаксис описания класса в языке Java. Модификаторы оператора описания класса
13. Описание полей в классе языка Java. Модификаторы описания поля.
14. Особенности наследования классов в языке Java.
15. Интерфейсы в языке Java. Назначение интерфейсов. Синтаксис описания интерфейсов
16. Реализация интерфейсов классами привести пример.
17. Интерфейс Comparable. Его назначение и характеристики
18. Класс Object. Его назначение, характеристики. Основные методы.
19. Класс Class. Его назначение и характеристики.
20. Что такое класс Object? Какие в нем есть методы?
21. Что такое метод equals(). Чем он отличается от операции ==.
22. Какие модификаторы доступа к членам класса в Java Вы знаете?
23. Какой из модификаторов более строгий: protected или package-private?
24. Если у класса-родителя есть метод, объявленный как private, может ли наследник расширить его видимость? А если protected?
25. Что означает ключевое слово final?
26. Имеет ли смысл объявлять метод private final?
27. Какие особенности инициализации final переменных?
28. Что означает ключевое поле static?
29. Что такое статический класс, какие особенности его использования?
30. Какие виды исключений в Java вы знаете, чем они отличаются?

Критерии и шкала оценивания по оценочному средству промежуточный контроль (зачет)

Характеристика знания предмета и ответов	Зачеты

Студент глубоко и в полном объёме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.	зачтено
Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.	
Студент знает только основной программный материал, допускает неточности, недостаточно чѐткие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.	
Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы.	не зачтено

Оценочные средства для итоговой аттестации (экзамен)

1. Какая конструкция используется в Java для обработки исключений?
2. Возможно ли использование блока try-finally (без catch)?
3. Ключевое слово throws. Для каких целей оно используется в языке Java?
4. Чем отличается абстрактный класс от интерфейса
5. В чем отличие конструкторов и обычных методов класса? Если класс имеет несколько конструкторов, можно ли из одного конструктора вызвать другой?
6. Объясните использование пакетов в Java.
7. Как записывается заголовок метода main?
8. Что такое классы-обертки? Для каких примитивных типов Java существуют классы-обертки? Дайте характеристику понятиям “boxing” и “unboxing”.
9. Зачем кроме примитивных типов в язык Java введены еще соответствующие классы-оболочки?
10. Можно ли использовать объекты числовых классов-оболочек в арифметических выражениях?
11. Какое наибольшее целое значение можно занести в объект класса BigInteger?
12. Какое наибольшее вещественное значение можно занести в объект класса BigDecimal?

13. Можно ли использовать в одном выражений значения примитивных типов и распакованные значения числовых классов-оболочек?
14. Объектно-ориентированное программирование. Классы, объекты, отношения.
15. Лексика и синтаксис языка – основные элементы.
16. Примитивные типы данных, преобразование примитивных типов.
17. Ссылочные типы данных (классы, интерфейсы, массивы, классы Object и Class).
18. Имена и пакеты, область видимости имен.
19. Классы – разграничение доступа, синтаксис описания класса.
20. Объектная модель в языке Java, интерфейсы, полиморфизм.
21. Массивы – статические массивы и классы динамических массивов.
22. Обработка ошибок – исключения.
23. Базовые инструментальные средства разработки программ на Java (javac, java, javadoc). Переменная ClassPath
24. Виртуальная машина. Структура программ. Типы переменных в Java.
25. Потоковый ввод-вывод в Java. Опишите иерархию классов для ввода вывода в Java. Байтовые потоки, символьные потоки.
26. Что такое перечисления (ENUM). Правила создания. Привести примеры.
27. Дать характеристику аннотациям в Java. Время жизни (хранения) аннотаций. Привести примеры.
28. Обобщенное программирование в Java. Основные синтаксические конструкции. Привести примеры.
29. Обработка коллекций в Java. Охарактеризовать основные типы коллекций.
30. Списки. Интерфейсы и классы реализации списков.
31. Множества (Set). Интерфейсы и классы реализации множеств.
32. Структуры даннях типа «ключ-значение». Особенности их реализации в Java.
33. Дайте характеристику классам StringBuilder и StringBuffer в Java.

1. Создайте класс, который описывает вектор (в трёхмерном пространстве).

У него должны быть:

- конструктор с параметрами в виде списка координат *x, y, z*
- метод, вычисляющий длину вектора. Корень можно посчитать с помощью *Math.sqrt()*:

$$\sqrt{x^2 + y^2 + z^2}$$

- метод, вычисляющий скалярное произведение:

$$x_1 x_2 + y_1 y_2 + z_1 z_2$$

- метод, вычисляющий векторное произведение с другим вектором:

$$(y_1 z_2 - z_1 y_2, z_1 x_2 - x_1 z_2, x_1 y_2 - y_1 x_2)$$

- метод, вычисляющий угол между векторами (или косинус угла): косинус угла между векторами равен скалярному произведению векторов, деленному на произведение модулей (длин) векторов:

$$\frac{(a, b)}{|a| \cdot |b|}$$

- методы для суммы и разности:

$$(x_1 + x_2, y_1 + y_2, z_1 + z_2)$$

$$(x_1 - x_2, y_1 - y_2, z_1 - z_2)$$

- статический метод, который принимает целое число *N*, и возвращает массив случайных векторов размером *N*.

Если метод возвращает вектор, то он должен возвращать новый объект, а не менять базовый. То есть, нужно реализовать шаблон "Неизменяемый объект"

2. Напишите класс, конструктор которого принимает два массива: массив значений и массив весов значений.

Класс должен содержать метод, который будет возвращать элемент из первого массива случайным образом, с учётом его веса.

Пример:

Дан массив [1, 2, 3], и массив весов [1, 2, 10].

В среднем, значение «1» должно возвращаться в 2 раза реже, чем значение «2» и в десять раз реже, чем значение «3».

Типовой экзаменационный билет

ФБГОУ ВО «ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ ВЛАДИМИРА ДАЛЯ»

Кафедра Информационные и управляющие системы

Экзаменационный контроль **Семестр** 3

Дисциплина «Кроссплатформенное программирование»

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Примитивные типы данных, преобразование примитивных типов.
2. Можно ли использовать объекты числовых классов-оболочек в арифметических выражениях?
3. Напишите класс, конструктор которого принимает два массива: массив значений и массив весов значений. Класс должен содержать метод, который будет возвращать элемент из первого массива случайным образом, с учётом его веса.

Утверждено на заседании кафедры _____

Протокол № ___

Зав. кафедрой _____ Преподаватель _____

Критерии и шкала оценивания

Критерии и шкала оценивания по оценочному средству итоговый контроль (экзамен)

Шкала оценивания	Характеристика знания предмета и ответов
отлично (5)	Студент глубоко и в полном объеме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.

хорошо (4)	Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.
удовлетворительно (3)	Студент знает только основной программный материал, допускает неточности, недостаточно четкие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.
неудовлетворительно (2)	Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы.

Методические материалы, определяющие процедуры оценивания знаний, умений, навыков

Требования к выполнению контрольных заданий определены Методическими указаниями для лабораторных работ по дисциплине «Кроссплатформенное программирование», (электронное издание) (для студентов по специальности 09.03.02 Информационные системы и технологии).

Контрольные сроки защиты лабораторных работ определены графиком учебного процесса, в частности, лабораторные работы 1-5 должны быть защищены в первой половине семестра, 6-10 до начала экзаменационной сессии.

Итоговая аттестация проводится в виде экзамена по билетам, содержащим 2 теоретических вопроса и одно практическое задание.

4. Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр[1]), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)

Примечание:

[1] **-** для ФОС по государственной итоговой аттестации указываются реквизиты протоколов заседания кафедр и подписи заведующих кафедрами, деканов/директоров, совместно реализующих ОП