

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Андрей Драгомирович Хлутков
Должность: директор
Дата подписания: 09.06.2026 20:14:41
Уникальный программный ключ:
880f7c07c583b07b775f6604a630281b13ca9fd2

Приложение 4
к образовательной программе

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.17. Программирование на языке *Java*

(индекс, наименование дисциплины в соответствии с учебным планом)

38.03.05 Бизнес-информатика

(код, наименование направления подготовки)

Бизнес-аналитика

(наименование образовательной программы)

очная форма обучения

(форма обучения)

Год набора – 2026

Санкт-Петербург

Автор(ы)-составитель(и) РПД:

Рассказов Владимир Александрович, кандидат технических наук, доцент кафедры бизнес-информатики

Заведующий кафедрой:

Наумов Владимир Николаевич, доктор военных наук, профессор, заведующий кафедрой бизнес-информатики

Рабочая программа дисциплины Б1.В.17. Программирование на языке *Java* одобрена на заседании кафедры бизнес-информатики факультета экономики и финансов СЗИУ РАНХиГС.

протокол № 6 от «26» марта 2026 г.

СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.
2. Объем и место дисциплины в структуре образовательной программы.
3. Содержание и структура дисциплины.
4. Типы оценочных материалов, показатели и критерии их оценивания.
5. Формы аттестации, типовые оценочные материалы для текущего контроля успеваемости обучающихся, критерии и шкалы оценивания по контрольным точкам.
6. Формы промежуточной аттестации, критерии и шкала оценивания, типовые оценочные материалы по дисциплине.
7. Методические материалы по освоению дисциплины.
8. Учебная литература и ресурсы информационно-телекоммуникационной сети «Интернет».
9. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы.

1 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Дисциплина Б1.В.17. Программирование на языке *Java* обеспечивает формирование у обучающихся следующих профессиональных компетенций:

ОТФ/ТФ и реквизиты ПС	Код компетенции	Наименование компетенции	Код индикатора достижения компетенций	Наименование индикатора достижения компетенций	Образовательный результат
06.015 СПЕЦИАЛИСТ ПО ИНФОРМАЦИОННЫМ СИСТЕМАМ С/14.6 Разработка архитектуры ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС	ПКС-4	Способен выполнять задачи проектирования и дизайна информационных систем, баз данных с использованием облачных, сетевых технологий	ПКС -4.1	Демонстрирует умение выполнять задачи проектирования и дизайна программных компонент и баз данных	ПКС-4.1. 3-13. Современные стандарты информационного взаимодействия систем ПКС-4.1. У-1. Проектировать архитектуру ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС

2 Объем и место дисциплины (модуля) в структуре образовательной программы

Общий объем дисциплины:

2,00 з.е., 72 ак.час

Контактная работа обучающихся с преподавателем по видам учебных занятий: 37 ак.час на контактную работу с преподавателем, из них 28 ак.часов на практические занятия, 9 ак.часов на аттестацию в период экзаменационной сессии, 35 ак.часа на самостоятельную работу обучающихся.

Дисциплина Б1.В.17 «Программирование на языке *Java*» относится к вариативной части учебного плана по направлению «Бизнес-информатика» 38.03.05. Преподавание дисциплины основано на дисциплине – Б1.О.11 «Программирование», Б1.В.14 «Введение в науку о данных. *SQL* и *Python*. В свою очередь дисциплина создаёт необходимые предпосылки для освоения программ таких дисциплин, как Б1.О.14 «Объектно-ориентированный анализ и программирование», ряда дисциплин по выбору студента, а также при выполнении научно-исследовательской и выпускной квалификационной работы.

3 Содержание и структура дисциплины (модуля)

3.1. Структура дисциплины (модуля)

Очная форма обучения

№ п/п	Наименование тем и (или) разделов	ВСЕГО	Объем дисциплины, ак.час										Форма текущего контроля успеваемости, промежуточной аттестации		
			Контактная работа обучающихся с преподавателем по видам учебных занятий							Самостоятельная работа					
			Период теоретического обучения				Период промежуточной аттестации (сессия)								
			Занятия лекционного типа		Занятия семинарского типа		ИК	КСР	КЭ	Кат.тэк	Контроль	СРкр		СРэк	СР
			Л	ВЛ	ЛР	ПЗ									
Тема 1.	Введение в технологию разработки программ на языке <i>Java</i>	16				8								8	Тестирование
Тема 2.	Рекурсия на языке <i>Java</i> . Алгоритмы поиска и сортировок на <i>Java</i> .	19				8								11	ПКЗ
Тема 3.	Объектно-ориентированное программирование на языке <i>Java</i>	16				8								8	Тестирование

Тема 4.	Использование классов из <i>java</i> -библиотек	12				4							8	ПКЗ
Промежуточная аттестация		9								9				Зачет с оценкой
Итого		72				28				9			35	

Используемые сокращения:

Л – лекции - занятия, предусматривающие преимущественную передачу учебной информации обучающимся педагогическими работниками организации и (или) лицами, привлекаемыми организацией к реализации образовательных программ на иных условиях,).

ВЛ – видео лекции.

ЛР – лабораторные работы.

ПЗ – практические занятия (за исключением лабораторных работ).

ИК – индивидуальные консультации.

КСР – контроль самостоятельной работы

КЭ – консультации перед экзаменом

Каттэк – контактная работа на аттестацию в период экзаменационных сессий

СРкр – самостоятельная работа на подготовку курсовой работы/ курсового проекта.

СРэк – самостоятельная работа на подготовку к экзамену.

СР – самостоятельная работа в семестре на подготовку к учебным занятиям

3.2. Содержание дисциплины

Тема 1. Введение в технологию разработки программ на языке *Java*. ПКС -4.1

Программная платформа *Java*. Программное обеспечение. *JDK* и *JRE*. Среды программирования. Основные характеристики языка *Java*. Структура программы на языке *Java*. Комментарии. Простые типы данных и литералы. Целые числа. Числа с плавающей точкой. Тип *char*. Тип *boolean*. Переменные. Область видимости переменных. Инициализация переменных. Константы. Операции. Операции инкрементирования и декрементирования. Операции отношений и логические операции. Битовые операции. Математические функции и константы. Арифметические выражения. Скобки и приоритет операций. Приведение типов. Перечислимые типы. Типы-оболочки. Программирование разветвляющихся процессов. Оператор условного перехода *if...else*. Оператор выбора *switch*. Программирование циклических процессов. Оператор цикла с параметром *for*. Оператор цикла с предусловием *while*. Оператор цикла с постусловием *do...while*. Операторы *break* и *continue*.

Одномерные и многомерны массивы данных. Копирование массивов.

Тема 2. Рекурсия на языке *Java*. Алгоритмы поиска и сортировок на *Java*. ПКС -4.1

Понятие рекурсии. Вспомогательные рекурсивные методы. Решение задач с помощью рекурсии (Числа Фибоначчи. Решение задачи: Ханойские башни.)

Определение методов и их использование. Передача значений через параметры. Перегрузка методов. Передача массива методу. Возвращение массива из метода. Списки параметров переменной длины

Алгоритмы сортировок на *Java*. Алгоритмы глупой сортировки и их асимптотика. Пузырьковая сортировка. Сортировка выбором. Сортировка вставками. Алгоритмы эффективной сортировки и их асимптотика. Сортировка слиянием. Быстрая сортировка. Пирамидальная сортировка. *TimSort*. Сравнение скоростей сортировок.

Алгоритмы поиска на *Java*. Линейный поиск. Бинарный поиск. Левый и правый бинарные поиски. Задача поиска двух элементов с фиксированной суммой. *JumpSearch*. Поиск Фибоначчи. Экспоненциальный поиск. Интерполяционный поиск.

Тема 3. Объектно-ориентированное программирование на языке Java. ПКС -4.1

Объекты и объектные переменные. Различия между переменными примитивных и ссылочных типов. Создание классов и объектов. Модификатор доступа. Создание методов. Статические поля и методы. Доступ к членам класса. Конструкторы. Подклассы. Ключевое слово *this*. Внутренние классы. Анонимные объекты. Абстрактные классы и методы. Пакеты и интерфейсы. Пакеты в *Java*. Интерфейсы. Свойства интерфейсов. Интерфейсные ссылки. Расширение интерфейсов. Интерфейсы и абстрактные классы. Клонирование объектов. Суперклассы и подклассы, особенности наследования в *Java*. Приведение объектов и оператор *instanceof*. Класс *ArrayList*. Полиморфизм. Отношения между классами. Агрегация и композиция. Автоматическое преобразование между примитивными типами и классами-обертками.

Тема 4. Использование классов из java-библиотек. ПКС -4.1

Использование классов из *java*-библиотеки. Класс *Date*. Класс *Random*. Класс *String*. Неизменяемые и интернированные строки. Замена и разделение строк. Поиск, замена и разделение с помощью шаблонов. Преобразования между строками и массивами. Преобразования символов и чисел в строки. Класс *Object* и его метод *toString()*.

4 Типы оценочных материалов, показатели и критерии оценивания

4.1. Оценочные материалы по дисциплине Б1.О.07.01 Математический анализ входят в состав оценочных материалов по образовательной программе. Совокупность оценочных материалов по всем дисциплинам (модулям) образовательной программы составляют фонд оценочных средств (далее – ФОС). ФОС используется при проведении текущего контроля успеваемости и промежуточной аттестации обучающихся с целью оценивания достижения обучающимися планируемых результатов обучения.

4.2. ФОС разработан как комплекс проверочных заданий различного типа и уровня сложности, включает критерии и шкалы оценивания, а также «ключи» правильных ответов. ФОС формируется как отдельный документ и хранится в электронном виде, доступ к ФОС предоставлен ограниченному кругу лиц.

4.3. Для самостоятельной работы обучающихся при подготовке к текущему контролю успеваемости и промежуточной аттестации в рабочих программах дисциплин размещены типовые проверочные задания закрытого типов.

Задания закрытого типа — это тестовые задания, в которых каждый вопрос

сопровождается готовыми вариантами ответов, из которых необходимо выбрать один или несколько правильных.

В зависимости от типа задания рекомендованы определенная последовательность выполнения и система оценивания выполнения заданий.

4.4. Типы заданий, сценарии выполнения, критерии оценивания

ТИП ЗАДАНИЯ	ИНСТРУКЦИЯ	СЦЕНАРИИ ВЫПОЛНЕНИЯ	КРИТЕРИИ ОЦЕНИВАНИЯ
Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных	Прочитайте текст, выберите правильный ответ	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа (например, 3 или В). 	Ответ считается верным, если правильно указана цифра или буква
Задание закрытого типа на установление соответствия	Прочитайте текст и установите соответствие	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидаются пары элементов. 2. Внимательно прочитать оба списка: список 1 – вопросы, утверждения, факты, понятия и т.д.; список 2 – утверждения, свойства объектов и т.д. 3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов. 4. Записать попарно буквы и цифры (в зависимости от задания) вариантов ответа (например, А1 или Б4). 	Ответ считается верным, если правильно указаны цифры или буквы
Задание закрытого типа с выбором нескольких правильных ответов из нескольких вариантов предложенных	Прочитайте текст, выберите правильные ответы	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать несколько правильных ответов. 4. Записать только номера (или буквы) выбранных вариантов ответа (например, 1 4 или А Г). 	Ответ считается верным, если правильно установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого)
Задание закрытого типа на установление последовательности	Прочитайте текст и установите последовательность	<ol style="list-style-type: none"> 1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается последовательность элементов. 2. Внимательно прочитать предложенные варианты 	Ответ считается верным, если правильно указана вся последовательность цифр

		<p>ответа.</p> <p>3. Построить верную последовательность из предложенных элементов.</p> <p>4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности (например, БВА или 135).</p>	
<p>Задание комбинированного типа с выбором одного правильного ответа из предложенных и обоснованием выбора</p>	<p>Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа</p>	<p>1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов.</p> <p>2. Внимательно прочитать предложенные варианты ответа.</p> <p>3. Выбрать один верный ответ.</p> <p>4. Записать только номер (или букву) выбранного варианта ответа.</p> <p>5. Записать аргументы, обосновывающие выбор ответа (например, 4 текст обоснования).</p>	<p>Ответ считается верным, если правильно указана цифра или буква и приведены корректные аргументы, используемые при выборе ответа</p>
<p>Задание открытого типа с развернутым ответом</p>	<p>Прочитайте текст и запишите развернутый обоснованный ответ</p>	<p>1. Внимательно прочитать текст задания и понять суть вопроса.</p> <p>2. Продумать логику и полноту ответа.</p> <p>3. Записать ответ, используя четкие компактные формулировки.</p> <p>4. В случае расчетной задачи, записать решение и ответ</p>	<p>Ответ считается верным:</p> <p>1. Отсутствие фактических ошибок.</p> <p>2. Раскрытие объема используемых понятий (полнота ответа).</p> <p>3. Обоснованность ответа (наличие аргументов).</p> <p>4. Логическая последовательность излагаемого материала.</p>

Общая шкала оценивания результатов текущего контроля успеваемости и промежуточной аттестации обучающихся с применением БРС

Итоговая балльная оценка	Традиционная система	Бинарная система	ECTS	
			Для традиционной системы	Для бинарной системы
95-100	Отлично	Зачтено	A	P/ Passed
85-94			B	P/ Passed
75-84	Хорошо		C	P/ Passed
65-74			D	P/ Passed
55-64			E	P/ Passed
0-54	Неудовлетворительно		Не зачтено	F

Соотношение баллов за текущий контроль успеваемости и промежуточную аттестацию, а также повторную промежуточную аттестацию:

Максимальная сумма баллов за текущий контроль успеваемости	Максимальная сумма баллов за промежуточную аттестацию	Максимальная итоговая балльная оценка	Максимальная сумма баллов за повторную промежуточную аттестацию
60 баллов	40 баллов	100 баллов	100 баллов

5 Формы аттестации, типовые оценочные материалы для текущего контроля успеваемости обучающихся, критерии и шкалы оценивания по контрольным точкам

5.1. В ходе реализации дисциплины Б1.В.18. Программирование на языке *Java* используются следующие формы текущего контроля успеваемости обучающихся (в том числе, задания к контрольным точкам):

тестирование, практическое контрольное задание.

Тема 1. Введение в технологию разработки программ на языке *Java*. ПКС-4.1.

Тест 1.1

Задание закрытого типа на установление соответствия

- Внимательно прочитать текст задания и понять, что в качестве ответа ожидаются пары элементов.
- Внимательно прочитать оба списка: список 1 – вопросы, утверждения, факты, понятия и т.д.;

список 2 – утверждения, свойства объектов и т.д.

3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов.

4. Записать попарно буквы и цифры (в зависимости от задания) вариантов ответа (например, А1 или Б4).

1) Спецификация языка <i>Java</i>	а) Техническое определение языка, включающее синтаксис и семантику языка программирования <i>Java</i> .
2) <i>JavaAPI</i>	б) Содержит predefined классы и интерфейсы для разработки <i>Java</i> -программ.
3) <i>JavaJDK</i>	с) Состоит из нескольких отдельных программ для разработки и тестирования <i>Java</i> -программ, каждая из которых вызывается из командной строки.
4) <i>Java IDE</i>	д) Программное обеспечение для быстрой разработки <i>Java</i> -программ, в котором редактирование, компиляция, создание, отладка программы и интерактивная справка интегрированы в единый графический интерфейс пользователя

Задание закрытого типа с выбором нескольких правильных ответов из нескольких вариантов предложенных

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.
2. Внимательно прочитать предложенные варианты-ты ответа.
3. Выбрать несколько правильных ответов.
4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).

- 1) Какие из следующих идентификаторов являются допустимыми?
 - а) $\$343$
 - б) *class*
 - в) *9X*
 - г) $8+9$
 - д) *Radius*

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных.

1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.
2. Внимательно прочитайте предложенные варианты-ты ответа.
3. Выбрать несколько правильных ответов.
4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).

- 2) Чтобы объявить внутри метода константу *MAX_LENGTH* со значением 99.98, необходимо написать ...
- a) *final MAX_LENGTH = 99.98;*
 - b) *final float MAX_LENGTH = 99.98;*
 - c) *double MAX_LENGTH = 99.98;*
 - d) *final double MAX_LENGTH = 99.98;*

Задание закрытого типа с выбором нескольких правильных ответов из нескольких вариантов предложенных

1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.
2. Внимательно прочитайте предложенные варианты-ты ответа.
3. Выбрать несколько правильных ответов.
4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).

- 3) Чтобы обновить значение переменной *sum*, увеличив его на *number*, необходимо написать ...
- a) *number += sum;*
 - b) *number = sum + number;*
 - c) *sum = Number + sum;*
 - d) *sum += number;*
 - e) *sum = sum + number;*

Задание закрытого типа на установление последовательности

1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидается последовательность элементов.
2. Внимательно прочитайте предложенные варианты ответа.

3. Построить верную последовательность из предложенных элементов.
4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности (например, БАВ или 135).

- 4) Расставьте в правильной последовательности операторы для создания массива из 10 целых чисел. Ключевые слова:
- a) `int`
 - b) `[]`
 - c) `nums`
 - d) `=`
 - e) `new`
 - f) `int[10];`

Задание комбинированного типа с выбором одного правильного ответа из предложенных и обоснованием выбора

1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов.
 2. Внимательно прочитайте предложенные варианты ответа.
 3. Выбрать один верный ответ.
 4. Записать только номер (или букву) выбранного варианта ответа.
 5. Записать аргументы, обосновывающие выбор ответа (например, 4 текст обоснования).
- 5) Проанализируйте следующую программу:

```
public class Test1 {
    public static void main(String[] args) {
        xMethod(new double[]{3, 3});
        xMethod(new double[5]);
        xMethod(new double[3]{1, 2, 3});
    }

    public static void xMethod(double[] a) {
        System.out.println(a.length);
    }
}
```

- a) Произойдет ошибка компиляции.
- b) Программа выполнится нормально.

Тема 2. Рекурсия на языке *Java*. Алгоритмы поиска и сортировок на *Java*. ПКС 4.1

Практическое контрольное задание 2.1

Задание 1 – Вычисление выражений.

Вычислите значение функции, значения аргументов вводятся с клавиатуры пользователем: $y = \frac{4}{3(r+34)} - 9(a+bc) + \frac{3+d(2+a)}{a+bd}$

Задание 2 – Условия.

Налоги и чаевые. Программа должна начинаться с запроса у пользователя суммы заказа в ресторане. После этого должен быть произведен расчет налога 13% и чаевых официанту 18%. На выходе программа должна отобразить отдельно налог, сумму чаевых и итоговую сумму, включая обе составляющие. Форматируйте вывод таким образом, чтобы все числа отображались с двумя знаками после запятой.

Задание 3 – Методы.

Создайте новую функцию (метод) *CountRange*, которая будет подсчитывать количество элементов в массиве, значения которых больше или равны заданному минимальному порогу и меньше максимального. Функция должна принимать три параметра: массив, минимальную границу и максимальную границу. Возвращать она будет целочисленное значение, большее или равное нулю, соответствующее количеству найденных элементов в массиве. В основной программе реализуйте демонстрацию вашей функции для нескольких массивов с разными минимальными и максимальными границами.

Задание 4 – Рекурсия. Напишите рекурсивный метод, который вычисляет $1 + 2 + 3 + \dots + n$ для положительного целого числа n

Тема 3. Объектно-ориентированное программирование на *Java*. ПКС 4.1

Тестовые задания 3.1 - Введение в ООП на *Java*

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных.

1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.
2. Внимательно прочитайте предложенные варианты-ты ответа.
3. Выбрать несколько правильных ответов.
4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).

1) Для объявления класса требуется ключевое слово ...

- a) *public*
- b) *private*
- c) *class*

d) все перечисленные

Задание открытого типа с развернутым ответом

1. Внимательно прочитайте текст задания и понять суть вопроса.
2. Продумать логику и полноту ответа.
3. Записать ответ, используя четкие компактные формулировки.
4. В случае расчетной задачи, записать решение и ответ

2) Проанализируйте следующую программу. Какой будет результат после ее выполнения

```
public class Test {
    int x;

    public Test(String t) {
        System.out.println("Test");
    }

    public static void main(String[] args) {
        Test = null;
        System.out.println(test.x);
    }
}
```

Задание закрытого типа с выбором нескольких правильных ответов из нескольких вариантов предложенных

1. Внимательно прочитайте текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.
2. Внимательно прочитайте предложенные вариант-ты ответа.
3. Выбрать несколько правильных ответов.
4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).

3) С какими из следующих утверждений вы согласны?

- a) При создании объекта типа *Random* необходимо указать начальное (случайное) число или использовать начальное (случайное) число, заданное по умолчанию.
- b) Если два объекта типа *Random* имеют одно и то же начальное (случайное) число, то последовательность случайных чисел, полученных из этих двух объектов, эквивалентна.

- c) Метод *nextInt()* класса *Random* возвращает следующее случайное значение типа *int*.
- d) Метод *nextDouble()* класса *Random* возвращает следующее случайное значение типа *double*.

Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных.

1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.
2. Внимательно прочитать предложенные вариант-ты ответа.
3. Выбрать несколько правильных ответов.
4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).

- 4) Что будет отображено в консоли вторым предложением *println()* метода *main()*?

```
public class Foo {
    int i;
    static int s;

    public static void main(String[] args) {
        Foo f1 = new Foo();
        System.out.println("f1.i равно " + f1.i + " f1.s равно " + f1.s);
        Foo f2 = new Foo();
        System.out.println("f2.i равно " + f2.i + " f2.s равно " + f2.s);
        Foo f3 = new Foo();
        System.out.println("f3.i равно " + f3.i + " f3.s равно " + f3.s);
    }

    public Foo() {
        i++;
        s++;
    }
}
```

- a) *f2.i* равно 1 *f2.s* равно 1
- b) *f2.i* равно 1 *f2.s* равно 2
- c) *f2.i* равно 2 *f2.s* равно 2
- d) *f2.i* равно 2 *f2.s* равно 1

Задание открытого типа с развернутым ответом

1. Внимательно прочитать текст задания и понять суть вопроса.
2. Продумать логику и полноту ответа.

3. Записать ответ, используя четкие компактные формулировки.

4. В случае расчетной задачи, записать решение и ответ

5) Проанализируйте следующую программу. Что будет в результате ее выполнения.

```
public class Test {
    public static void main(String args[]) {
        NClass nc = new NClass();
        nc.t = nc.t++;
    }
}

class NClass {
    int t;
    private NClass() {
    }
}
```

Тема 4. Использование классов из *java*-библиотек. ПКС 4.1

Практическое контрольное задание 4.1

Задание 1 – Простой класс.

Создайте класс *Stock* для представления акций компании. Класс *Stock* должен содержать:

- Строковое поле данных с именем *symbol* для обозначения акций.
- Строковое поле данных с именем *name* для наименования акций.
- Поле данных *previousClosingPrice* типа *double*, в котором хранится стоимость акций на момент закрытия предыдущего дня.
- Поле данных *currentPrice* типа *double*, в котором хранится стоимость акций в настоящий момент.
- Конструктор, создающий акции с указанными обозначением и наименованием.
- Метод с именем *getChangePercent()*, который возвращает процент изменения стоимости акций с *previousClosingPrice* на *currentPrice*.

Нарисуйте *UML*-диаграмму класса *Stock*, а затем реализуйте этот класс. Напишите клиент этого класса — программу, которая создает объект типа *Stock* с обозначением *SBER*, именем ПАО Сбербанк, стоимостью акций на момент закрытия предыдущего дня, равной 281.50. Задайте новую стоимость

акций в настоящий момент, равную 282.87, и отобразите процент изменения стоимости акций.

Задание 2— Наследование, конструкторы, методы.

Класс *Rectangle*. Создайте класс *Rectangle* для представления треугольников, который порождается от класса *GeometricObject*. Класс *Rectangle* должен содержать:

- Два поля данных типа *double* с именами *width*, *height* с заданными по умолчанию значениями, равными 1.0 и 2.0, для обозначения сторон прямоугольника.
- Безаргументный конструктор, который создает треугольник с заданными по умолчанию значениями.
- Конструктор, который создает треугольник с *width*, *height*.
- *Getter*-методы для всех трех полей данных.
- Метод с именем *getArea()*, который возвращает площадь этого прямоугольника.
- Метод с именем *getPerimeter()*, который возвращает периметр этого прямоугольника.
- Метод с именем *toString()*, который возвращает строковое описание прямоугольника.

Формулы для вычисления площади треугольника:

$$s = width + height,$$
$$area = s \cdot width - 2 \cdot width \cdot height,$$

где *s* — периметр, а *area* — площадь треугольника.

Метод *toString()* должен быть реализован следующим образом:

```
return "Прямоугольник: ширина = " + width + " высота = " + height;
```

Нарисуйте *UML*-диаграммы классов *Rectangle* и *GeometricObject*, а затем реализуйте эти классы. Напишите клиент этих классов — программу, которая запрашивает у пользователя ввести три стороны треугольника, цвет и логическое значение для указания закрашен ли треугольник. Программа должна создавать объект типа *Rectangle* с указанными сторонами и присваивать значения свойствам цвет (*color*) и заливка (*isFilled*) с помощью этих входных данных. Программа должна отображать площадь (*Area*), периметр (*Perimeter*), цвет, а также *true* или *false* для указания, закрашен треугольник или нет.

Задание 3 - Строки.

Java-библиотека предоставляет класс *String*. Предоставьте собственную реализацию следующих методов (назовите новый класс *MyString1*):

```
public MyString1(char[] chars);
public char charAt(int index);
public int length();
public MyString1 substring(int begin, int end);
public MyString1 toLowerCase();
public static MyString1 valueOf(int i);
public char[] toChars();
public boolean equals(MyString1 obj);
```

5.2. Типовые оценочные материалы для текущего контроля успеваемости обучающихся (вне контрольных точек):

приведены в п.6.2.

5.3. Один или несколько тематических блоков дисциплины завершаются контрольной точкой (далее – КТ). Текущий контроль успеваемости по дисциплине предусматривает не менее 2 (двух) и не более 10 (десяти) КТ в течение периода освоения дисциплины.

Максимальное количество баллов за любой тип работ в рамках КТ составляет 100 (сто) баллов.

Распределение весовых коэффициентов по КТ в рамках текущего контроля успеваемости по дисциплине и формулы расчета:

Наименование контрольной точки	Максимальное количество баллов за работу в рамках КТ, которое может набрать студент	Коэффициент веса контрольной точки	Результат контрольной точки, участвующий в формировании итоговой балльной оценки по дисциплине (отражается в журнале БРС в СДО)
1 семестр			
Тема 1. Введение в технологию разработки программ на языке <i>Java</i> .			
КТ - 1 Тест 1.1	100	0,1	10
Тема 2. Рекурсия на языке <i>Java</i> . Алгоритмы поиска и сортировок на <i>Java</i> .			
КТ - 2 ПКЗ 2.1	100	0,2	20
Тема 3. Объектно-ориентированное программирование на языке <i>Java</i> .			
КТ - 3 Тест 3.1	100	0,1	10
Тема 4. Использование классов из <i>java</i> -библиотек			
КТ – 4 ПКЗ 4.1	100	0,2	26
Итого:	x	0,6	60

Формула расчета результата контрольной точки:

Результат контрольной точки = Количество баллов за работу в рамках КТ x Коэффициент веса контрольной точки.

5.4. Формы текущего контроля успеваемости обучающихся в рамках КТ и типовые оценочные материалы:

КТ-1.Тема 1

Тест 1.1- Основы Java программирования.

КТ-2. Тема 2

ПКЗ 2.1 - Рекурсия на языке Java. Алгоритмы поиска и сортировок на Java.

КТ-3. Тема 3

Тест 3.1 – Объектно-ориентированное программирование на языке Java.

КТ-4. Тема 4

ПКЗ 4.1 - Использование классов из java-библиотек.

Для каждой формы текущего контроля успеваемости обучающихся в рамках КТ определены критерии оценивания результатов выполнения задания.

1. Критерии оценивания тестирования:

Критерии оценки	Диапазон баллов	Описание критерия
<i>Количество правильных ответов</i>	<i>0</i>	<i>Количество правильных ответов менее 55%</i>
	<i>25</i>	<i>Количество правильных ответов от 55% до 64%</i>
	<i>50</i>	<i>Количество правильных ответов от 65% до 74%</i>
	<i>75</i>	<i>Количество правильных ответов от 75% до 84%</i>
	<i>100</i>	<i>Количество правильных ответов от 85% до 100%</i>
Итого максимально:	100	

2. Критерии оценивания практического контрольного задания:

Диапазон баллов	Описание критерия
85-100	Обучающийся полно излагает материал (отвечает на вопрос), дает правильное определение основных понятий; обнаруживает понимание материала, может обосновать свои суждения, применить знания на

	практике, привести необходимые примеры не только из учебника, но и самостоятельно составленные; излагает материал последовательно и правильно с точки зрения норм литературного языка.
65-84	Обучающийся дает ответ, удовлетворяющий тем же требованиям, что и для оценки «отлично», но допускает 1–2 ошибки, которые сам же исправляет, и 1–2 недочета в последовательности и языковом оформлении излагаемого.
55-64	Обучающийся обнаруживает знание и понимание основных положений данной темы, но излагает материал неполно и допускает неточности в определении понятий или формулировке правил; не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.
0-54	Обучающийся обнаруживает незнание вопроса, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал.

5.5. Описание дополнительных материалов и оборудования, необходимых для выполнения проверочных заданий (*при необходимости*).

Для решения тестовых заданий студенту разрешается использование калькулятора; программ для работы с электронными таблицами для обработки, анализа и визуализации данных.

6 Формы промежуточной аттестации, критерии и шкала оценивания, типовые оценочные материалы по дисциплине

6.1. Промежуточная аттестация проводится в форме:

3 семестр – **зачет с оценкой**.

Зачет с оценкой проводится в письменной форме. Обучающийся получает экзаменационный билет с двумя теоретическими и одним практическим вопросами. На выполнение заданий дается 45 минут. По завершении подготовки необходимо представить ответы в письменном виде, подробно изложив ход выполнения задания, сделать выводы (при необходимости).

При реализации промежуточной аттестации в ЭО/ДОТ могут быть использованы следующие формы: устно в ДОТ - в форме обоснованных ответов на задания различного типа; письменно в СДО - в форме письменного решения заданий различного типа; тестирование в СДО.

6.2. Типовые оценочные материалы промежуточной аттестации.

Вопросы для подготовки к зачету.

1. Среды программирования для *Java*. Пакет *JDK* и среда *JRE*.
2. *Java*-компилятор и *Java*-интерпретатор. Байт-код.
3. Структура программы на языке *Java*.
4. Включение комментариев в программу на *Java*. Виды комментариев.
5. Парадигмы программирования.
6. Среды программирования *IntelliJ Idea* и *Giga IDE*.
7. Целочисленные типы данных.
8. Стандартные математические функции и арифметические выражения.
9. Строковый и символьный литералы. Строковые типы данных.
10. Область видимости переменных.
11. Локальные переменные. Объявление и область видимости. Создание переменных, объявленных внутри блока. Механизм удаление переменной.
12. Структурированные типы данных.
13. Линейные алгоритмы. Операции ввода и вывода данных. Составление арифметических выражений. примеры.
14. Операторы условного перехода. Программирование разветвляющихся процессов. Синтаксис оператора *if-then-else*. Оператор выбора *switch*. Синтаксис. Примеры.
15. Программирование циклических процессов. Оператор цикла с предусловием. Оператор цикла с постусловием. Оператор цикла с параметром. Операторы *break* и *continue*.
16. ООП. Понятие объекта, класса, метода. Структура класса. Основные принципы ООП: инкапсуляция, наследование и полиморфизм. Создание классов и методов Статические методы.
17. Понятие конструктора класса. Создание конструктора класса. Конструктор по умолчанию.
18. Наследование класса. Создание подклассов. Интерфейсы классов. Создание интерфейсов и методов. Доступ к полям суперкласса. ключевое слово *this*.
19. Интерфейсы. Свойства интерфейсов Интерфейсные ссылки. Расширение интерфейсов. Интерфейсы и абстрактные классы. Примеры.
20. Реализация полиморфизма в *Java*. Статические поля и методы класса.
21. Строки символов. Работа со строковыми методами. Поиск и замена подстрок.
22. Коллекции и перечисления. Итераторы. Работа с итераторами.

Типовые задания для зачета:

1. Двумерные массивы. Поиск суммы наибольших элементов строк.
2. Двумерные массивы. Поиск произведения наименьших элементов строк.
3. Обход массива, оператор цикла "for each". Перебор всех элементы массива в порядке возрастания индексов.
4. Напишите метод, который удаляет повторяющиеся элементы из целочисленного списка типа *ArrayList*, используя следующий заголовок:

```
public static void removeDuplicate(ArrayList<Integer> list)
```

Напишите тестовую программу, которая запрашивает у пользователя 10 целых чисел для списка, отображает несовпадающие целые числа в порядке их ввода и отделяет их друг от друга только одним пробелом.

5. Создайте класс с именем *MyDate* для представления даты. Класс *MyDate* должен содержать:

- Поля данных *year*, *month* и *day*, которые представляют дату. Поле *month* должно быть основано на числе, т.е. 0 — для января, 1 — для февраля и т.д.
- Безаргументный конструктор, который создает объект типа *MyDate* для текущей даты.
- Конструктор, который создает объект типа *MyDate* с указанным временем в миллисекундах, прошедших с 00:00, 1 января 1970 г.
- Конструктор, который создает объект типа *MyDate* с указанными годом, месяцем и днем.
- Три *getter*-метода для полей данных *year*, *month* и *day*, соответственно.
- Метод с именем *setDate(long elapsedTime)*, который присваивает новую дату объекту, используя прошедшее время.

Типовые проверочные задания для самоподготовки обучающегося к промежуточной аттестации:

ТИП ЗАДАНИЯ	СЦЕНАРИИ ВЫПОЛНЕНИЯ	ТИПОВЫЕ ЗАДАНИЯ
Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа (например, А или В).	1. Какое логическое выражение истинно, если $x \in [-10, 10]$? a) $(x > 10) \text{ И } (x < -10)$ b) $(x > 10) \text{ ИЛИ } (x < -10)$ c) $(x < 10) \text{ ИЛИ } (x > = -10)$ d) $(x > = -10) \text{ И } (x < = 10)$
		2. Все <i>Java</i> -программы должны иметь метод ... a) <code>public static Main(String[] args)</code> b) <code>public static Main(String args[])</code> c) <code>public static void main(String[] args)</code> d) <code>public void main(String[] args)</code>

		<p>e) <code>public static main(String[] args)</code></p> <p>3. Дан следующий метод:</p> <pre>static void nPrint(String message, int n) { while (n > 0) { System.out.print(message); n--; } }</pre> <p>Чему будет равно k после вызова метода <code>nPrint("Сообщение ", k)</code> в следующем фрагменте кода?</p> <pre>int k = 2; nPrint("Сообщение ", k);</pre> <p>a) 0 b) 1 c) 2 d) 3 e) 0,7</p>																						
<p>Задание закрытого типа на установление соответствия</p>	<p>1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидаются пары элементов.</p> <p>2. Внимательно прочитать оба списка: список 1 – вопросы, утверждения, факты, понятия и т.д.; список 2 – утверждения, свойства объектов и т.д.</p> <p>3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов.</p> <p>4. Записать попарно буквы и цифры (в зависимости от задания) вариантов ответа (например, А1 или Б4).</p>	<p>1. Установите соответствие между оператором и выполняемым действием.</p> <table border="1" data-bbox="890 846 1481 1064"> <tr> <td>1) %</td> <td>a) Тернарная операция, аналог условной конструкции <code>if</code></td> </tr> <tr> <td>2) ?</td> <td>b) Остаток от деления</td> </tr> <tr> <td>3) !=</td> <td>c) Логическое И</td> </tr> <tr> <td>4) &&</td> <td>d) Не равно</td> </tr> </table> <p>2. Установите соответствие между оператором и выполняемым действием.</p> <table border="1" data-bbox="890 1137 1481 1473"> <thead> <tr> <th>Операция</th> <th>Приоритет</th> </tr> </thead> <tbody> <tr> <td>1. <code>-i</code></td> <td>a) инкремент</td> </tr> <tr> <td>2.</td> <td>b) выделение памяти</td> </tr> <tr> <td>3. <code>new</code></td> <td>c) инверсия</td> </tr> <tr> <td>4. <code>i = i</code></td> <td>d) умножение</td> </tr> <tr> <td>5. <code>i</code></td> <td>e) проверка на эквивалентность</td> </tr> <tr> <td>6. <code>++i</code></td> <td>f) вычитание с присвоением</td> </tr> </tbody> </table>	1) %	a) Тернарная операция, аналог условной конструкции <code>if</code>	2) ?	b) Остаток от деления	3) !=	c) Логическое И	4) &&	d) Не равно	Операция	Приоритет	1. <code>-i</code>	a) инкремент	2.	b) выделение памяти	3. <code>new</code>	c) инверсия	4. <code>i = i</code>	d) умножение	5. <code>i</code>	e) проверка на эквивалентность	6. <code>++i</code>	f) вычитание с присвоением
1) %	a) Тернарная операция, аналог условной конструкции <code>if</code>																							
2) ?	b) Остаток от деления																							
3) !=	c) Логическое И																							
4) &&	d) Не равно																							
Операция	Приоритет																							
1. <code>-i</code>	a) инкремент																							
2.	b) выделение памяти																							
3. <code>new</code>	c) инверсия																							
4. <code>i = i</code>	d) умножение																							
5. <code>i</code>	e) проверка на эквивалентность																							
6. <code>++i</code>	f) вычитание с присвоением																							
<p>Задание закрытого типа с выбором нескольких правильных ответов из нескольких вариантов предложенных</p>	<p>1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов.</p> <p>2. Внимательно прочитать предложенные варианты ответа.</p> <p>3. Выбрать несколько правильных ответов.</p> <p>4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).</p>	<p>1. Закончите код следующего метода для вычисления факториала.</p> <pre>/** * Возвращает n! */ public static long factorial(int n) { if (n == 0) // простой случай return 1; else return ...; // рекурсивный вызов }</pre> <p>a) $n * (n - 1)$ b) n c) $n * factorial(n - 1)$ d) $factorial(n - 1) * n$</p> <p>2. С какими из следующих утверждений вы согласны?</p> <p>a) Рекурсивные методы работают быстрее,</p>																						

		<p>чем нерекурсивные.</p> <p>b) Рекурсивные методы обычно занимают больше памяти, чем нерекурсивные.</p> <p>c) Рекурсивный метод всегда может быть заменен на нерекурсивный.</p> <p>d) В некоторых случаях использование рекурсии дает нам естественное и простое решение задачи, которую было бы трудно решить иначе.</p>
Задание закрытого типа на установление последовательности	<p>1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается последовательность элементов.</p> <p>2. Внимательно прочитать предложенные варианты ответа.</p> <p>3. Построить верную последовательность из предложенных элементов.</p> <p>4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности (например, БВА или 135).</p>	<p>1. Расставьте в правильной последовательности основные этапы создания обобщенного класса <i>MI</i> с одним параметром:</p> <p>a) <i>class</i></p> <p>b) <i>{</i></p> <p>c) <i>}</i></p> <p>d) <i>public</i></p> <p>e) <i><T></i></p> <p>f) <i>MI</i></p> <p>g) <i>value</i></p>
Задание комбинированного типа с выбором одного правильного ответа из предложенных и обоснованием выбора	<p>1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов.</p> <p>2. Внимательно прочитать предложенные варианты ответа.</p> <p>3. Выбрать один верный ответ.</p> <p>4. Записать только номер (или букву) выбранного варианта ответа.</p> <p>5. Записать аргументы, обосновывающие выбор ответа.</p>	<p>1. В каком случае требуется применить явное преобразование типов.</p> <p>a) Из <i>byte</i> в <i>int</i></p> <p>b) Из <i>short</i> в <i>int</i></p> <p>c) Из <i>int</i> в <i>float</i></p> <p>d) Из <i>double</i> в <i>float</i>;</p> <p>e) Из <i>float</i> в <i>double</i>.</p> <p>2. Какой результат выполнения следующего кода?</p> <pre>public class A {public int num;} A obj1 = new A{num = 3}; A obj2 = new A(num = 5); A obj3 = new A(num = 6); obj2 = obj1 obj2.A += 3; Console.WriteLine(obj1.A==obj3.A)</pre> <p>a) <i>True</i>;</p> <p>b) <i>False</i>;</p> <p>c) Ошибка на этапе компиляции;</p> <p>d) Ошибка на этапе выполнения.</p>
Задание открытого типа с развернутым ответом	<p>1. Внимательно прочитать текст задания и понять суть вопроса.</p> <p>2. Продумать логику и полноту ответа.</p> <p>3. Записать ответ, используя четкие компактные формулировки.</p> <p>4. В случае расчетной задачи, записать решение и ответ</p>	<p>Пусть <i>s1</i> равно " <i>Welcome</i> " и <i>s2</i> равно " <i>welcome</i> ". Напишите код для следующих предложений:</p> <p>1. Проверьте, равны ли <i>s1</i> и <i>s2</i>, и присвойте результат логической переменной <i>isEqual</i>.</p> <p>2. Проверьте, равны ли <i>s1</i> и <i>s2</i>, игнорируя регистр, и присвойте результат логической переменной <i>isEqual</i>.</p> <p>3. Сравните <i>s1</i> и <i>s2</i> и присвойте результат переменной <i>x</i> типа <i>int</i>.</p> <p>4. Сравните <i>s1</i> и <i>s2</i>, игнорируя регистр, и присвойте результат переменной <i>x</i> типа <i>int</i>.</p> <p>5. Проверьте, имеет ли <i>s1</i> префикс <i>AAA</i>, и присвойте результат логической переменной <i>b</i>.</p> <p>1) Пусть <i>s1</i> равно " <i>Welcome</i> " и <i>s2</i> равно " <i>welcome</i> ".</p>

		<p>Напишите код для следующих предложений:</p> <ol style="list-style-type: none"> 1. Проверьте, имеет ли <i>s1</i> префикс <i>AAA</i>, и присвойте результат логической переменной <i>b</i>. 2. Проверьте, имеет ли <i>s1</i> суффикс <i>AAA</i>, и присвойте результат логической переменной <i>b</i>. 3. Присвойте длину <i>s1</i> переменной <i>x</i> типа <i>int</i>. 4. Присвойте первый символ <i>s1</i> переменной <i>x</i> типа <i>char</i>. 5. Создайте новую строку <i>s3</i>, соединяющую <i>s1</i> и <i>s2</i>. 6. Создайте подстроку <i>s1</i>, начиная с индекса, равного 1. 7. Создайте подстроку <i>s1</i>, начиная с индекса, равного 1, и заканчивая индексом, равным 4. 8. Создайте новую строку <i>s3</i>, которая преобразует <i>s1</i> в строчные буквы. 9. Создайте новую строку <i>s3</i>, которая преобразует <i>s1</i> в заглавные буквы. 10. Создайте новую строку <i>s3</i>, которая усекает пробелы с обоих концов <i>s1</i>. 11. Присвойте переменной <i>x</i> типа <i>int</i> индекс первого вхождения символа <i>e</i> в <i>s1</i>. 12. Присвойте переменной <i>x</i> типа <i>int</i> индекс последнего вхождения строки <i>abc</i> в <i>s1</i>. <p>2) Запишите код программы, которая конвертирует температуру из градусов Цельсия в Кельвины.</p>
--	--	---

6.3. Критерии и шкала оценивания на основе БРС.

Критерии и балльная шкала определяются преподавателем

Зачет с оценкой

КРИТЕРИИ ОЦЕНИВАНИЯ	РЕЗУЛЬТАТ В БАЛЛАХ
<i>Дан полный, в логической последовательности развернутый ответ на поставленный вопрос, где он продемонстрировал знания предмета в полном объеме учебной программы, достаточно глубоко осмысливает дисциплину, самостоятельно, и исчерпывающе отвечает на дополнительные вопросы, приводит собственные примеры по проблематике поставленного вопроса, решил предложенные практические задания без ошибок</i>	40
<i>Дан развернутый ответ на поставленный вопрос, где студент демонстрирует знания, приобретенные на лекционных и семинарских занятиях, а также полученные посредством изучения обязательных учебных материалов по курсу, дает аргументированные ответы, приводит примеры, в ответе присутствует свободное владение монологической речью, логичность и последовательность ответа. Однако допускается неточность в ответе. Решил предложенные практические задания с небольшими неточностями.</i>	30-39
<i>Дан ответ, свидетельствующий в основном о знании процессов изучаемой дисциплины, отличающийся недостаточной глубиной и полнотой раскрытия темы, знанием основных вопросов теории, слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры, недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа и решении практических заданий.</i>	20-29

<p><i>Дан ответ, который содержит ряд серьезных неточностей, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы, незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов, неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Выводы поверхностны. Решение практических заданий не выполнено, т.е. студент не способен ответить на вопросы даже при дополнительных наводящих вопросах преподавателя.</i></p>	<p>0-19</p>
--	-------------

7 Методические материалы по освоению дисциплины

Для изучения основных вопросов образовательной программы необходимо конспектировать материалы лекций, работать с рекомендованной преподавателем литературой, а также ресурсами информационно-телекоммуникационной сети «Интернет». Для приобретения навыков активного использования знаний полезно обсуждать плановые и возникающие вопросы, а также решаемые задачи на практических занятиях. Чтобы легче и прочнее усвоить материал следует постоянно использовать конкретные примеры, сравнения из уже полученных областей наук.

Для закрепления изученного материала даны вопросы по каждой теме дисциплины, на которые следует самостоятельно найти ответы.

Важной составной частью учебного процесса в вузе являются практические занятия. Практические занятия проводятся главным образом по дисциплинам, требующим закрепления навыков решения задач. Все практические занятия проводятся в компьютерных классах с использованием ИСР *VisualStudio* среды *MSOffice*. Каждая работа должна быть защищена, т.е. студент должен ответить на вопросы преподавателя о ходе выполнения работы, а также на вопросы теоретического характера.

При подготовке к практическим занятиям необходимо проанализировать конспект лекции, ознакомиться с рекомендованной литературой по соответствующей теме, осуществить подготовку по рекомендованным в рабочей программе вопросам для обсуждения темы, выполнить домашнее задание (при необходимости).

Необходимо помнить, что на лекции обычно рассматривается не весь материал, а только его часть. Остальная его часть восполняется в процессе самостоятельной работы. В связи с этим работа с рекомендованной литературой обязательна. Особое внимание при этом необходимо обратить на содержание основных положений и выводов, объяснение явлений и фактов, уяснение практического приложения рассматриваемых теоретических вопросов. В процессе этой работы студент должен стремиться понять и

запомнить основные положения рассматриваемого материала, примеры, поясняющие его, а также разобраться в иллюстративном материале.

В процессе подготовки к занятиям рекомендуется взаимное обсуждение материала, во время которого закрепляются знания, а также приобретается практика в изложении и разъяснении полученных знаний, развивается речь. При необходимости следует обращаться за консультацией к преподавателю (в том числе по электронной почте).

Планируя консультацию, необходимо хорошо продумать вопросы, которые требуют разъяснения. Заканчивать подготовку следует составлением плана (конспекта) по изучаемому материалу (вопросу). Это позволяет составить концентрированное, сжатое представление по изучаемым вопросам. Записи имеют первостепенное значение для самостоятельной работы студентов. Они помогают понять построение изучаемого материала, выделить основные положения, проследить их логику. Кроме того, ведение записей способствует превращению чтения в активный процесс, мобилизует, наряду со зрительной, и моторную память. Следует помнить: у студента, систематически ведущего записи, создается свой индивидуальный фонд методических материалов для быстрого повторения изученных вопросов, для мобилизации накопленных знаний. Особенно важны и полезны записи тогда, когда в них находят отражение мысли, возникшие при самостоятельной работе.

После изучения базовых тем курса проводится текущий контроль знаний студентов в виде опроса или письменного тестирования. Типовые тесты и задания по темам дисциплины приведены в специальном разделе данной рабочей программы.

Подготовка к текущему и промежуточному контролю предполагает изучение представленных вопросов к зачету, работу над тестами, представленными в данной рабочей программе, выполнение семестровой проектной работы по применению системного подхода и методов системного анализа к выбранной системе.

Работа в малых группах – это одна из самых популярных форм проведения занятий, так как она дает всем обучающимся (в том числе и стеснительным) возможность участвовать в работе, практиковать навыки сотрудничества, межличностного общения (в частности, умение активно слушать, вырабатывать общее мнение, разрешать возникающие разногласия). Цель данной формы проведения занятий: продемонстрировать сходство или различия определенных явлений, выработать стратегию или разработать план, выяснить отношение различных групп участников к одному и тому же вопросу. В ходе этой работы дополнительно решаются следующие задачи: развитие навыков общения и взаимодействия в группе, формирование ценностно-

ориентационного единства группы, поощрение к гибкой смене социальных ролей в зависимости от ситуации.

Группа студентов делится на несколько малых групп. Количество групп определяется числом творческих заданий, которые будут обсуждаться в процессе занятия. Малые группы формируются либо по желанию студентов, либо по родственной тематике для обсуждения. Каждая малая группа обсуждает творческое задание в течение отведенного времени. Основной этап – проведение обсуждения творческого задания. Заслушиваются суждения, предлагаемые каждой малой группой по творческому заданию. Преподаватель дает оценочное суждение и работе малых групп, по решению творческих заданий, и эффективности предложенных путей решения.

С целью активизации самостоятельной работы студентов в системе дистанционного обучения *Moodle* разработан учебный курс «Программирование», включающий набор файлов с текстами лекций, заданиями для выполнения практических и лабораторных работ. Для активизации работы студентов во время контактной работы с преподавателем отдельные занятия проводятся в интерактивной форме. В основном, интерактивная форма занятий обеспечивается при проведении занятий в компьютерном классе. Интерактивная форма обеспечивается наличием разработанных файлов с заданиями, наличием контрольных вопросов, возможностью доступа к системе дистанционного обучения, а также к тестеру.

В качестве самостоятельной работы студентами выполняется семестровая работа по всем темам. При выполнении заданий по темам могут использоваться представленные студентом материалы по предыдущим темам. Выполненная семестровая работа представляется студентом на открытой защите на промежуточной аттестации.

8 Учебная литература и ресурсы информационно-телекоммуникационной сети Интернет

8.1. Основная литература

1. Свистунов, А. Н. Построение распределенных систем на Java: учебное пособие / А. Н. Свистунов. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 316 с. — ISBN 978-5-4497-0940-0. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/102045.html>. — Режим доступа: для авторизир. пользователей.

2. Ермаков, А. В. Объектно-ориентированное программирование в задачах на языке *Java*: учебное пособие / А. В. Ермаков. — Саратов: Саратовский государственный технический университет имени Ю.А. Гагарина, ЭБС АСВ, 2022. — 156 с. — ISBN 978-57433-3478-0. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/128034.html>. — Режим доступа: для авторизир. пользователей.

3. Брюс, Эккель *Философия Java*. 4-е полное издание /изд. Питер-Санкт-Петербург, 2022. – 1168 с. - ISBN 978-5-4461-1107-7

8.2. Дополнительная литература

1. Вязовик, Н. А. Программирование на Java: учебное пособие / Н. А. Вязовик. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 601 с. — ISBN 978-5-4497-0852-6. — Текст: электронный // Цифровой образовательный ресурс IPR SMART:[сайт].—URL: <https://www.iprbookshop.ru/102048.html>. — Режим доступа: для авторизир. Пользователей

2. Гуськова, О. И. Объектно-ориентированное программирование в *Java* : учебное пособие : [16+] / О. И. Гуськова. – Москва : Московский педагогический государственный университет (МПГУ), 2020. – 240 с. : ил. – Режим доступа: по подписке. –URL:<https://biblioclub.ru/index.php?page=book&id=500355>.– Библиогр. в кн. – ISBN 978-5- 4263-0648-6. – Текст : электронный.

8.3. Нормативные правовые документы и иная правовая информация

Не используются

8.4 Интернет-ресурсы

Обучающимся обеспечен доступ к материалам курса в СДО Академии <http://lms.ranepa.ru>, а также через сайт научной библиотеки к следующим подписным электронным ресурсам:

Русскоязычные ресурсы

- Электронные учебники электронно-библиотечной системы (ЭБС) «Айбукс»
- Электронные учебники электронно-библиотечной системы (ЭБС) «Юрайт»
- Электронные учебники электронно-библиотечной системы (ЭБС) «Лань»
- Электронные учебники электронно-библиотечной системы (ЭБС) «ZNANIUM.COM»
- Электронные учебники электронно-библиотечной системы (ЭБС) «BOOK.RU»
- Электронные учебники электронно-библиотечной системы (ЭБС) «IPRSMART»

9. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

№ п/п	Наименование
1.	Специализированные залы для проведения лекций, оснащенные персональным компьютером/ноутбуком и мультимедийным проектором
2.	Аудитории и компьютерные классы, оборудованные посадочными местами и персональными компьютерами с выходом в Интернет для проведения практических занятий. Программное обеспечение: <i>MSVisualStudio</i> , <i>MSVisualStudioCode</i>
3.	«МТС Линк» — российская платформа для онлайн-коммуникаций и совместной работы команд ; «Яндекс Телемост» — сервис для видеоконференций от Яндекса; Я-мессенджер
4.	Технические средства обучения: персональные компьютеры; программные средства, обеспечивающие просмотр видеофайлов в форматах <i>AVI</i> , <i>MPEG-4</i> , <i>DivX</i> , <i>RMVB</i> , <i>WMV</i> ; программы для работы с электронными таблицами для обработки, анализа и визуализации данных; соответствующие онлайн-инструменты для построения интеллект-карты и моделей в различных нотациях
5.	Научная библиотека (в т.ч. электронные информационные ресурсы научной библиотеки)
6.	СДО Академии https://lms.ranepa.ru/