

<p>Документ подписан простой электронной подписью Информация о владельце: ФИО: Таскаев Сергей Валерьевич Должность: Ректор Дата подписания: 05.04.2025 14:49:02 Уникальный программный ключ: 054c0182970293149c21699f0009940292896884</p>	<p>МИНОБРАЗОВАНИЯ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)</p> <p>Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю) «Программирование на языке Java (научный семинар)» по направлению подготовки (специальности) 02.03.02 «Фундаментальная информатика и информационные технологии» направленности (профилю) «Математические и алгоритмические основы интеллектуальных систем» ФГБОУ ВО «ЧелГУ»</p>	<p>стр. 1</p>
---	--	---------------

**Фонд оценочных средств для промежуточной аттестации
по дисциплине (модулю)**

Программирование на языке Java (научный семинар)

Направление подготовки (специальность)

02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль)

Математические и алгоритмические основы интеллектуальных систем

Присваиваемая квалификация (степень)

бакалавр

Форма обучения

очная

Троицк, 2025 г.



Содержание

1. Паспорт фонда оценочных средств
2. Перечень формируемых компетенций
 - 2.1. Компетенции, закреплённые за дисциплиной
3. Содержание оценочных средств по дисциплине
 - 3.1. Виды оценочных средств
 - 3.2. Содержание оценочных средств
4. Порядок проведения и критерии оценивания промежуточной аттестации
 - 4.1. Порядок проведения промежуточной аттестации
 - 4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств
 - 4.3. Результаты промежуточной аттестации и уровни сформированности компетенций



1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

Направление подготовки: *02.03.02 «Фундаментальная информатика и информационные технологии»*

Направленность (профиль): *Математические и алгоритмические основы интеллектуальных систем*

Дисциплина: *Программирование на языке Java (научный семинар)*

Семестр изучения: *8*

Форма промежуточной аттестации: *экзамен*

Для оценивания результатов используется балльно-рейтинговая система

2. ПЕРЕЧЕНЬ ФОРМИРУЕМЫХ КОМПЕТЕНЦИЙ

2.1. Компетенции, закреплённые за дисциплиной

Изучение дисциплины «Программирование на языке Java (научный семинар)» направлено на формирование следующих компетенций:

Коды компетенции и согласно ФГОС (ОПОП ВО)	Содержание компетенций согласно ФГОС (ОПОП ВО)	Индикаторы достижения компетенции согласно ОПОП	Перечень планируемых результатов обучения по дисциплине
1	2	3	4
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1. Выполняет поиск информации, определяет критерии системного анализа поставленных задач УК-1.2. Использует критический анализ, систематизацию и обобщение информации для решения поставленных задач ПК-1.1. Обладает знаниями о методологии и этапах выполнения научно-исследовательской работы; о методах решения научных задач; о методике подготовки отчета, в том числе выпускной квалификационной работы	Знать: критерии системного анализа поставленных задач (для достижения УК-1.1) Уметь: систематизировать и обобщать информацию (для достижения УК-1.2) Владеть: навыками использования критического анализа проектов (для достижения УК-1.3)
ПК-1	Способность проводить под научным руководством локальные научно-исследовательские и опытно-конструкторские	ПК-1.1. Обладает знаниями о методологии и этапах выполнения научно-исследовательской работы; о методах решения научных задач; о методике подготовки отчета, в том числе выпускной квалификационной работы ПК-1.2. Демонстрирует умения: обрабатывать и анализировать научно-техническую информацию и результаты исследований; выполнять под научным	Знать: о методике подготовки отчета, в том числе выпускной квалификационной работы (для достижения ПК-1.1) Уметь: выполнять под научным руководством разработку проекта на языке Java (для достижения ПК-1.2)



	е разработки на основе существующих методов в конкретной области профессиональной деятельности	руководством научно-исследовательскую или опытно-конструкторскую разработку в конкретной области профессиональной деятельности. ПК-1.3. Имеет практический опыт (навыки): научной аргументации при анализе объекта научной и профессиональной деятельности; подготовки научных обзоров, публикаций, рефератов и библиографий по тематике проводимых исследований.	Владеть: навыками написания рефератов по тематике программирования на Java (для достижения ПК-1.3)
ПК-2	Способность к осуществлению интеграции программных модулей и компонент и проверки работоспособности программного продукта на основе международных и профессиональных стандартов информационных технологий, современных парадигм и методологий, инструментальных и вычислительных средств, методов и механизмов оценки и анализа функционирования средств и систем информационных технологий	ПК-2.1. Обладает знаниями о методах и средствах сборки модулей и компонент программного обеспечения, о разработке процедур для развертывания программного обеспечения, миграции и преобразования данных, о создании программных интерфейсов; о методах и механизмах оценки и анализа функционирования средств и систем информационных технологий; о международных и профессиональных стандартах информационных технологий, о современных парадигмах и методологиях, инструментальных и вычислительных средствах. ПК-2.2. Демонстрирует умения: применять методы и средства сборки модулей и компонент программного обеспечения, разработки процедур для развертывания программного обеспечения, миграции и преобразования данных, создания программных интерфейсов; проводить проверку и оценку работоспособности программного продукта. ПК-2.3. Имеет практический опыт (навыки): сборки модулей и компонент программного обеспечения, разработки процедур для развертывания программного обеспечения, миграции и преобразования данных, создания программных интерфейсов; оценки работоспособности программного продукта.	Знать: методы и средства разработки на языке Java (для достижения ПК-2.1) Уметь: применять методы и средства создания программных интерфейсов на языке Java (для достижения ПК-2.2) Владеть: навыками оценки работоспособности программного продукта (для достижения ПК-2.3)
ПК-3	Способность к разработке требований и проектированию программного обеспечения на основе	ПК-3.1. Обладает знаниями о методах и средствах проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов. ПК-3.2. Демонстрирует умения: разрабатывать требования к программному продукту, применять методы и средства проектирования	Знать: методы и средства проектирования программного обеспечения на языке Java (для достижения ПК-3.1) Уметь: применять методы и средства проектирования программного обеспечения



	применения базовых математических знаний и информационных технологий при решении проектно-технических и прикладных задач	программного обеспечения, структур данных, баз данных, программных интерфейсов. ПК-3.3. Имеет практический опыт (навыки): проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.	на Java (для достижения ПК-3.2) Владеть: навыками проектирования программных интерфейсов на языке Java (для достижения ПК-3.3)
--	--	---	---

3. СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

3.1 Виды оценочных средств

№ п/п	Контролируемые темы/разделы	Код компетенции/планируемые результаты обучения	Наименование оценочного средства для текущего контроля	Наименование оценочного средства на промежуточной аттестации
1	Введение	УК-1, ПК-1, ПК-2, ПК-3 (знания, умения, навыки)	Лабораторная работа, домашнее задание	Экзамен: Устный опрос (вопросы 1-4) Практическое задание
2	Типы данных и лексика языка	УК-1, ПК-1, ПК-2, ПК-3 (знания, умения, навыки)	Лабораторная работа, домашнее задание	Экзамен: Устный опрос (вопросы 5-7) Практическое задание
3	Операторы и структура кода	УК-1, ПК-1, ПК-2, ПК-3 (знания, умения, навыки)	Лабораторная работа, домашнее задание	Экзамен: Устный опрос (вопросы 8-11) Практическое задание
4	Объектная модель в Java	УК-1, ПК-1, ПК-2, ПК-3 (знания, умения, навыки)	Лабораторная работа, домашнее задание	Экзамен: Устный опрос (вопросы 12-19) Практическое задание
5	Параметризация. Пакеты	УК-1, ПК-1, ПК-2, ПК-3 (знания, умения, навыки)	Лабораторная работа, домашнее задание	Экзамен: Устный опрос (вопросы 20-22) Практическое задание
6	Коллекции объектов	УК-1, ПК-1, ПК-2, ПК-3 (знания, умения, навыки)	Лабораторная работа, домашнее задание	Экзамен: Устный опрос (вопросы 23-24) Практическое задание
7	Обработка ошибок и исключения	УК-1, ПК-1, ПК-2, ПК-3 (знания, умения, навыки)	Лабораторная работа, домашнее задание	Экзамен: Устный опрос



		навыки)	задание	(вопросы 25-28) Практическое задание
8	Массивы	УК-1, ПК-1, ПК-2, ПК-3 (знания, умения, навыки)	Лабораторная работа, домашнее задание	Экзамен: Устный опрос (вопросы 29-30) Практическое задание
9	Система ввода\вывода в Java	УК-1, ПК-1, ПК-2, ПК-3 (знания, умения, навыки)	Лабораторная работа, домашнее задание	Экзамен: Устный опрос (вопросы 31-33) Практическое задание
10	Информация о типах RTTI	УК-1, ПК-1, ПК-2, ПК-3 (знания, умения, навыки)	Лабораторная работа, домашнее задание	Экзамен: Устный опрос (вопросы 34-36) Практическое задание
11	Параллельное выполнение	УК-1, ПК-1, ПК-2, ПК-3 (знания, умения, навыки)	Лабораторная работа, домашнее задание	Экзамен: Устный опрос (вопросы 37-41) Практическое задание

Типовые задания, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе по дисциплине. Полные комплекты оценочных средств и контрольно-измерительных материалов хранятся на кафедре и являются учебно-методическими материалами ограниченного (конфиденциального) пользования.

3.2 Содержание оценочных средств

Оценочные средства для промежуточной аттестации представлены базой контрольных вопросов, и практических заданий к экзамену.

3.2.1. База контрольных вопросов к экзамену

1. Отличие интерфейса от реализации.
2. Наследование, полиморфизм, инкапсуляция.
3. Программа "Hello, world!" на Java.
4. Ссылки и объекты. Ключевое слово static
5. Примитивные типы данных: целочисленный, дробный, булевский.
6. Ссылочный тип данных Null. класс String.



7. Идентификаторы, ключевые слова, литералы
8. Операторы if, switch, while, do, for.
9. Управление ходом программы.
10. Операторы break, continue.
11. Оператор return
12. Создание новых классов.
13. Синтаксисы композиции, наследования, делегирования.
14. Разработка с наследованием.
15. Абстрактные классы и методы.
16. Интерфейсы. Отделение интерфейса от реализации.
17. Ключевые слова this и super.
18. Разработка с интерфейсами.
19. Внутренние классы
20. Параметризация объектов в Java 5.
21. Параметризованные методы и и интерфейсы.
22. Пакеты. Основные пакеты в Java
23. Основные концепции. List.
24. Итераторы. Синтаксис foreach. Collection. Map. Set. Queue
25. Основные исключения. Перехват исключений.
26. Создание собственных исключений.
27. Отличия Exception, RuntimeException, Error и Throwable.
28. Ключевое слово finally
29. Особенности массивов. Массив как объект.
30. Многомерные массивы
31. Класс File.
32. Классы Reader и Writer.
33. Потоки данных: InputStream и OutputStream и их классы реализации
34. Необходимость в динамическом определении типов (RTTI).
35. Динамическая информация о классе (reflection).
36. Классы как объекты. ClassLoader'ы
37. Многопоточная архитектура java. Класс Thread.
38. Интерфейс Runnable.
39. Ключевое слово synchronized и совместное использование ресурсов.
40. Ключевое слово synchronized как модификатор методов.
41. Потокбезопасное программирование.



3.2.2. Тематика практических заданий экзамена

1. Заполните массив случайными числами и выведете максимальное, минимальное и среднее значение. Для генерации случайного числа используйте метод `Math.random()`, который возвращает значение в промежутке $[0, 1]$.

2. Реализуйте алгоритм сортировки пузырьком для сортировки массива.

3. Напишите программу, которая выводит на консоль простые числа в промежутке от $[2, 100]$. Используйте для решения этой задачи оператор "%" (остаток от деления) и циклы.

4. Дан массив целых чисел и ещё одно целое число. Удалите все вхождения этого числа из массива (пропусков быть не должно).

5. 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()`

метод, вычисляющий скалярное произведение

метод, вычисляющий векторное произведение с другим вектором

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

методы для суммы и разности

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".

6. 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».

7. Напишите метод, который проверяет, входит ли в массив заданный элемент или нет. Используйте перебор и двоичный поиск для решения этой



задачи. Сравните время выполнения обоих решений для больших массивов (например, 100000000 элементов).

8. Найдите корень заданного уравнения на заданном отрезке с точностью по x не хуже, чем 0.001. Известно, что на этом промежутке корень единственный. Используйте для этого метод деления отрезка пополам (и рекурсию).

9. Реализовать иерархию классов, описывающую трёхмерные фигуры (см. приложение 1). Класс `Box` является контейнером, он может содержать в себе другие фигуры. Метод `add()` принимает на вход `Shape`. Нужно добавлять новые фигуры до тех пор, пока для них хватает места в `Box` (будем считать только объём, игнорируя форму. Допустим, мы переливаем жидкость). Если места для добавления новой фигуры не хватает, то метод должен вернуть `false`.

10. Реализуйте ту же иерархию классов, но сделав некоторые классы абстрактными.

11. Реализуйте ту же иерархию классов, но используя интерфейсы.

12. Добавьте в иерархию классов фигуру вращения для произвольной функции. Вычислять объём можно приближенно с помощью определённого интеграла. Реализовать метод прямоугольников.

13. Напишите метод для поиска самой длинной строки в массиве.

14. Напишите метод, который проверяет является ли слово палиндромом.

15. Напишите метод, заменяющий в тексте все вхождения слова «бьяка» на «[вырезано цензурой]».

16. Имеются две строки. Напишите метод, возвращающий количество вхождений одной строки в другую.

17. Постройте частотный словарь букв русского (или английского) алфавита. Опустим проблему выбора и анализа корпуса языка, достаточно будет взять текст небольшой длины).

18. Напишите класс `BaseConverter` для конвертации из градусов по Цельсию в Кельвины, Фаренгейты, и так далее. У метода должен быть метод `convert`, который и делает конвертацию.

19. Напишите свой класс `StringBuilder` с поддержкой операции `undo`. Для этого делегируйте все методы стандартному `StringBuilder`, а в собственном классе храните список всех операций для выполнения `undo()`. Это будет реализацией шаблона «Команда».



20. Напишите свой класс `StringBuilder`, с возможностью оповещения других объектов об изменении своего состояния. Для этого делегируйте все методы стандартному `StringBuilder`, а в собственном классе реализуйте шаблон проектирования «Наблюдатель».

21. Напишите метод `filter`, который принимает на вход массив (любого типа) и реализацию интерфейса `Filter` с методом `apply(Object o)`, чтобы убрать из массива лишнее. Проверьте, как он работает на строках или других объектах.

22. Напишите метод `fill`, который принимает массив объектов, и реализацию интерфейса `Function` (или своего). Метод `fill` должен заполнить массив, получая новое значение по индексу с помощью реализации интерфейса `Function`.

23. Напишите метод, который на вход получает коллекцию объектов, а возвращает коллекцию уже без дубликатов.

24. Напишите метод, который добавляет 1000000 элементов в `ArrayList` и `LinkedList`. Напишите еще один метод, который выбирает из заполненного списка элемент наугад 100000 раз. Замерьте время, которое потрачено на это. Сравните результаты и предположите, почему они именно такие.

25. Написать итератор по массиву.

26. Напишите итератор по двумерному массиву.

27. Дан итератор. Метод `next()` возвращает либо `String`, либо итератор такой же структуры (то есть который опять возвращает или `String`, или такой же итератор). Напишите поверх этого итератора другой, уже «плоский».

28. Напишите итератор, который проходит по двум итераторам.

29. Напишите метод, который получает на вход `Map<K, V>` и возвращает `Map`, где ключи и значения поменяны местами. Так как значения могут совпадать, то тип значения в `Map` будет уже не `K`, а `Collection<K>`.

30. Напишите метод, который получает на вход массив элементов типа `K` (`Generic`) и возвращает `Map<K, Integer>`, где `K` — значение из массива, а `Integer` — количество вхождений в массив.

31. Выведите состояние потока перед его запуском, после запуска и во время выполнения.

32. Напишите программу, в которой создаются два потока, которые выводят на консоль своё имя по очереди.

33. Дано два потока — производитель и потребитель. Производитель генерирует некоторые данные (в примере — числа). Производитель «потребляет» их.



4. ПОРЯДОК ПРОВЕДЕНИЯ И КРИТЕРИИ ОЦЕНИВАНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

4.1. Порядок проведения промежуточной аттестации

Промежуточная аттестация проводится в форме экзамена в два этапа. На первом этапе студенту выдаётся теоретический вопрос по одному из разделов дисциплины из базы контрольных вопросов. Время выполнения – до 20 минут. На втором этапе выдается практическое задание по написанию программы на компьютере. Время выполнения – до 60 минут.

При дистанционном обучении устный опрос, в том числе защита курсовых работ, проводятся в Microsoft Teams. Практические задания и письменные ответы размещаются в системе Moodle. Тестирование осуществляется в системе Moodle.

4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств

4.2.1. Критерии оценивания на экзамене

Лабораторная работа 0-5

Домашняя работа 0-5

Экзамен 0-40

Суммарно по дисциплине можно получить 100 баллов, из них текущая работа оценивается в 60 баллов, итоговая форма контроля - в 40 баллов. Минимальное количество для допуска к экзамену 28 баллов.

90 баллов и более - "отлично" (отл.) - ответ на теоретический вопрос и решение задачи билета;

71-89 баллов - "хорошо" (хор.) - неполный ответ на теоретический вопрос или неполное решение задачи билета;

66-70 баллов - "удовлетворительно" (удов.) - отсутствие ответа на теоретический вопрос, либо отсутствие решение задачи билета;

65 баллов и менее - "неудовлетворительно" (неуд.) - полное или практически полное отсутствие ответа на теоретический вопрос и решения задачи билета.

4.3. Результаты промежуточной аттестации и уровни

**сформированности компетенций**

Код компетенции	Планируемые результаты обучения по дисциплине	Критерии оценивания			
		Отлично Высокий уровень освоения проверяемых компетенций	Хорошо Средний уровень освоения проверяемых компетенций	Удовлетворительно Базовый уровень освоения проверяемых компетенций	Неудовлетворительно Недостаточный уровень освоения проверяемых компетенций
УК-1	<i>Знать:</i> критерии системного анализа поставленных задач <i>Уметь:</i> систематизировать и обобщать информацию <i>Владеть:</i> навыками использования критического анализа проектов	<i>Знает:</i> критерии системного анализа поставленных задач <i>Умеет:</i> систематизировать и обобщать информацию <i>Владеет:</i> навыками использования критического анализа проектов	<i>Знает:</i> критерии системного анализа поставленных задач, но допускает несущественные ошибки <i>Умеет:</i> систематизировать и обобщать информацию, но допускает несущественные ошибки <i>Владеет:</i> навыками использования критического анализа проектов, но допускает несущественные ошибки	<i>Знает:</i> в ограниченном объеме критерии системного анализа поставленных задач <i>Умеет:</i> в ограниченном объеме систематизировать и обобщать информацию <i>Владеет:</i> в ограниченном объеме навыками использования критического анализа проектов	<i>Не знает:</i> критерии системного анализа поставленных задач <i>Не умеет:</i> систематизировать и обобщать информацию <i>Не владеет:</i> навыками использования критического анализа проектов
ПК-1	<i>Знать:</i> о методике подготовки отчета, в том числе выпускной квалификационной работы <i>Уметь:</i> выполнять под научным руководством разработку проекта на языке Java <i>Владеть:</i> навыками написания рефератов по тематике программирования на Java	<i>Знает:</i> о методике подготовки отчета, в том числе выпускной квалификационной работы <i>Умеет:</i> выполнять под научным руководством разработку проекта на языке Java <i>Владеет:</i> навыками написания рефератов по тематике программирования на Java	<i>Знает:</i> о методике подготовки отчета, в том числе выпускной квалификационной работы, но допускает несущественные ошибки <i>Умеет:</i> выполнять под научным руководством разработку проекта на языке Java, но допускает несущественные ошибки <i>Владеет:</i> навыками написания рефератов по тематике программирования	<i>Знает:</i> в ограниченном объеме о методике подготовки отчета, в том числе выпускной квалификационной работы <i>Умеет:</i> в ограниченном объеме выполнять под научным руководством разработку проекта на языке Java <i>Владеет:</i> в ограниченном объеме навыками написания рефератов по тематике программирования	<i>Не знает:</i> о методике подготовки отчета, в том числе выпускной квалификационной работы <i>Не умеет:</i> выполнять под научным руководством разработку проекта на языке Java <i>Не владеет:</i> навыками написания рефератов по тематике программирования на Java



			я на Java, но допускает несущественные ошибки	я на Java	
ПК-2	<i>Знать:</i> методы и средства разработки на языке Java <i>Уметь:</i> применять методы и средства создания программных интерфейсов на языке Java <i>Владеть:</i> навыками оценки работоспособности программного продукта	<i>Знает:</i> методы и средства разработки на языке Java <i>Умеет:</i> применять методы и средства создания программных интерфейсов на языке Java <i>Владеет:</i> навыками оценки работоспособности и программного продукта	<i>Знает:</i> методы и средства разработки на языке Java, но допускает несущественные ошибки <i>Умеет:</i> применять методы и средства создания программных интерфейсов на языке Java, но допускает несущественные ошибки <i>Владеет:</i> навыками оценки работоспособности и программного продукта, но допускает несущественные ошибки	<i>Знает:</i> в ограниченном объеме методы и средства разработки на языке Java <i>Умеет:</i> в ограниченном объеме применять методы и средства создания программных интерфейсов на языке Java <i>Владеет:</i> в ограниченном объеме навыками оценки работоспособности и программного продукта	<i>Не знает:</i> методы и средства разработки на языке Java <i>Не умеет:</i> применять методы и средства создания программных интерфейсов на языке Java <i>Не владеет:</i> навыками оценки работоспособности и программного продукта
ПК-3	<i>Знать:</i> методы и средства проектирования программного обеспечения на языке Java <i>Уметь:</i> применять методы и средства проектирования программного обеспечения на языке Java <i>Владеть:</i> навыками проектирования программных интерфейсов на языке Java	<i>Знает:</i> методы и средства проектирования программного обеспечения на языке Java <i>Умеет:</i> применять методы и средства проектирования программного обеспечения на Java <i>Владеет:</i> навыками проектирования программных интерфейсов на языке Java	<i>Знает:</i> методы и средства проектирования программного обеспечения на языке Java, но допускает несущественные ошибки <i>Умеет:</i> применять методы и средства проектирования программного обеспечения на Java, но допускает несущественные ошибки <i>Владеет:</i> навыками проектирования программных интерфейсов на языке Java, но допускает несущественные ошибки	<i>Знает:</i> в ограниченном объеме методы и средства проектирования программного обеспечения на языке Java <i>Умеет:</i> в ограниченном объеме применять методы и средства проектирования программного обеспечения на Java <i>Владеет:</i> в ограниченном объеме навыками проектирования программных интерфейсов на языке Java	<i>Не знает:</i> методы и средства проектирования программного обеспечения на языке Java <i>Не умеет:</i> применять методы и средства проектирования программного обеспечения на Java <i>Не владеет:</i> навыками проектирования программных интерфейсов на языке Java



Уровни сформированности компетенций определяются следующим образом:

1. Высокий уровень соответствует оценке “отлично” и предполагает:
 - готовность к самостоятельной профессиональной деятельности;
 - глубокое и правильное усвоение программного материала, последовательное, грамотное и логически стройное его изложение;
 - владение основными методами и алгоритмами решения задач;
 - умение строить математические модели, увязывать теорию с практикой, применять знания.
2. Средний уровень соответствует оценке “хорошо” и предполагает:
 - твердое знание программного материала, его изложение грамотное и по существу;
 - владение основными методами;
 - отсутствие существенных ошибок, но затруднения в выводах и доказательствах;
 - умение применять основные положения для решения задач.
3. Базовый уровень соответствует оценке “удовлетворительно” и предполагает:
 - знания только основного материала, неумение делать выводы и проводить доказательства;
 - ошибки, недостаточно правильные формулировки;
 - трудное увязывание основных положений с практикой.
4. Низкий уровень соответствует оценке “неудовлетворительно” и предполагает:
 - незнание основополагающих вопросов изучаемого курса или значительной части программного материала;
 - ошибки, неумение их исправлять;
 - не умение увязать теорию с практикой.

