

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гаскаев Сергей Валерьевич
Должность: Ректор
Дата подписания: 15.06.2026 12:22:45
Уникальный программный ключ:
04c19ed8bfb98f3b6b772486b9a87888352474



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)

Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю) «Объектно-ориентированное программирование на языке Java» по направлению подготовки 02.03.02 «Фундаментальная информатика и информационные технологии» направленности «Прикладное программирование и системы искусственного интеллекта» ФГБОУ ВО «ЧелГУ»

стр. 1

Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю)
«Объектно-ориентированное программирование на языке Java»

Направление подготовки (специальность)
02.03.02 «Фундаментальная информатика и информационные технологии»

Направленность (профиль)
«Прикладное программирование и системы искусственного интеллекта»

Присваиваемая квалификация
Бакалавр

Форма обучения
Очная

Год набора
2026

Челябинск, 2026 г.



Содержание

1. Паспорт фонда оценочных средств	3
2. Перечень формируемых компетенций	4
3. Содержание оценочных средств по дисциплине	6
3.1. Виды оценочных средств	6
3.2. Содержание оценочных средств	7
4. Порядок проведения и критерии оценивания промежуточной аттестации	11
4.1. Порядок проведения промежуточной аттестации	11
4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств	11
4.3. Результаты промежуточной аттестации и уровни сформированности компетенций.....	11



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)

Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю) «Объектно-ориентированное программирование на языке Java» по направлению подготовки 02.03.02 «Фундаментальная информатика и информационные технологии» направленности «Прикладное программирование и системы искусственного интеллекта» ФГБОУ ВО «ЧелГУ»

стр. 3

1. Паспорт фонда оценочных средств

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии.

Направленность: Прикладное программирование и системы искусственного интеллекта.

Дисциплина: Объектно-ориентированное программирование на языке Java.

Семестры: 3.

Форма промежуточной аттестации: дифференцированный зачёт в 3 семестре.

Для оценивания результатов обучения используется балльно-рейтинговая система.



2. Перечень формируемых компетенций

Изучение дисциплины «Объектно-ориентированное программирование на языке Java» направлено на формирование компетенций, приведённых в 1.

Таблица 1. Результаты обучения по дисциплине.

Код и наименование компетенции согласно ФГОС	Индикаторы достижения компетенций согласно ОПОП ВО	Перечень планируемых результатов обучения по дисциплине
1	2	3
ПК-2 Способен 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 о методах и средствах сборки модулей и компонент программного обеспечения, о разработке процедур для развертывания программного обеспечения, миграции и преобразования данных; основные понятия ООП; простейшие паттерны программирования (взаимосвязь классов). Уметь применять методы и средства сборки модулей и компонент программного обеспечения, разработки процедур для развертывания программного обеспечения, миграции и преобразования данных; проектировать и реализовывать простейшие классы; проектировать и реализовывать различные конструкции из связанных классов. Владеть навыками сборки модулей и компонент, использования средств настройки отдельных элементов в рамках платформы Android.
ПК-3 Способен к разработке требований и проектированию программного обеспечения на основе применения базовых математических знаний и информационных технологий при решении проектно-технических и прикладных задач	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 программного обеспечения, структур данных, баз данных, программных интерфейсов. ПК-3.3. Имеет практический опыт (навыки): проектирования программного	Знать о методах и средствах проектирования программного обеспечения; современные инструментальные и вычислительные средства ООП Уметь разрабатывать требования к программному продукту, применять методы и средства проектирования программного обеспечения; создавать простые приложения на основе современной ИТ-платформы с использованием библиотеки классов данной платформы и разработкой собственных классов.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)

Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю) «Объектно-ориентированное программирование на языке Java» по направлению подготовки 02.03.02 «Фундаментальная информатика и информационные технологии» направленности «Прикладное программирование и системы искусственного интеллекта» ФГБОУ ВО «ЧелГУ»

стр. 5

обеспечения, структур данных, баз данных, программных интерфейсов.

Владеть навыками проектирования программного обеспечения; использования среды разработки Java; платформы для разработки мобильных приложений Android на базовом уровне.



3. Содержание оценочных средств по дисциплине

3.1. Виды оценочных средств

Таблица 2. Виды оценочных средств.

Код, наименование компетенции согласно ФГОС	Перечень планируемых результатов обучения по дисциплине	Контролируемые темы/разделы (номер и название раздела из РПД п.2.2)	Семестр	Номер задания	Наименование оценочного средства
ПК-2 Способен к осуществлению интеграции программных модулей и компонент и проверки работоспособности и 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	Знать о методах и средствах сборки модулей и компонент программного обеспечения, о разработке процедур для развертывания программного обеспечения, миграции и преобразования данных; основные понятия ООП; простейшие паттерны программирования (взаимосвязь классов).	Введение в ООП Объектно – ориентированный язык программирования Java Использование методов ООП на примере платформы Android	3	1-28	Вопросы к зачету
	Уметь применять методы и средства сборки модулей и компонент программного обеспечения, разработки процедур для развертывания программного обеспечения, миграции и преобразования данных; проектировать и реализовывать простейшие классы; проектировать и реализовывать различные конструкции из связанных классов.				
	Владеть навыками сборки модулей и компонент, использования средств настройки отдельных элементов в рамках платформы Android.				
ПК-3 Способен к разработке требований и проектированию программного обеспечения на основе	Знать о методах и средствах проектирования программного обеспечения; современные инструментальные и			1-40	Практическое задание



применения базовых математических знаний и информационных технологий при решении проектно-технических и прикладных задач	вычислительные средства ООП.				
	Уметь разрабатывать требования к программному продукту, применять методы и средства проектирования программного обеспечения; создавать простые приложения на основе современной ИТ-платформы с использованием библиотеки классов данной платформы и разработкой собственных классов.				
	Владеть навыками проектирования программного обеспечения; использования среды разработки Java; платформы для разработки мобильных приложений Android на базовом уровне.				

Типовые задания, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе дисциплины (модуля). Полные комплекты оценочных средств и контрольно-измерительных материалов хранятся на кафедре.

3.2. Содержание оценочных средств

Промежуточная аттестация проводится в виде дифференцированного зачёта в 3 семестре.

Вопросы для зачёта:

1. Что такое сущность? Что такое объект?
2. Что такое свойства объекта? Что такое методы объекта?
3. Что такое инкапсуляция? Что такое класс? Что такое экземпляр класса? Как выглядит общая форма определения класса?
4. Массивы в Java. Что такое массив? Как объявить массив в Java? Как заполнить массив в Java? Как создать двумерный массив в Java? Что такое свойство length у массивов в Java? Как при помощи цикла for each обработать массив?
5. Строки в Java. Как создать строку в Java? Какие методы есть в Java для работы со строками? Как осуществить конкатенацию строк в Java?
6. Как написать метод, принимающий в качестве параметра объект? Чем отличается передача методу простых типов данных от передачи методу объектов?
7. Как перегрузить конструктор? Зачем это нужно?
8. Что такое статические свойства и методы?
9. Как создать метод с переменным числом аргументов? Как его вызвать?
10. Что такое наследование? подкласс? суперкласс?



11. Чем отличает переопределение метода от перегрузки?
12. Что такое динамическая диспетчеризация методов?
13. Что такое абстрактный метод? абстрактный класс? какое ограничение при компиляции связано с использованием ключевого слова `abstract`?
14. Для чего нужно ключевое слово `final`? Как оно используется?
15. Что такое класс `Object`? Какие его методы мы можем переопределить? Какие - не можем? Почему?
16. Что такое пакет? Как его создать? Права доступа в пакетах, определяемые модификаторами `public`, `private`, `protected` и модификатором по умолчанию.
17. Что такое интерфейс? как создать? как использовать? Что такое интерфейсная ссылка?
18. Наследование интерфейсов.
19. Методы интерфейсов, используемые по умолчанию.
20. Множественное наследование интерфейсов. Статические методы интерфейсов.
21. Как создать проект `Android`? В чем практический смысл выбора версии `API`? Какие основные файлы содержит структура приложения?
22. Как создать обработчик нажатия на кнопку? Как получить значение из элемента, расположенного в макете?
23. Что такое файл манифеста `Android`? Что такое `интент`? Как из одной активности запустить другую? Как передать через `интент` информацию из одной активности в другую?
24. Как при помощи `интента` запустить активность из другого приложения? Как настроить фильтр `интентов` для активностей в своем приложении?
25. Как настроить в приложении запуск кода через определенные промежутки времени? Объекты `Handler` и `Runnable`.
26. Иерархия классов активностей. Кто, у кого и какие методы наследует?
27. Как сохранить данные при уничтожении активности? Объект `Bundle`.
28. Жизненный цикл активности. В каком порядке могут запускаться методы `onCreate()`, `onStart()`, `onRestart()`, `onResume()`, `onPause()`, `onStop()`, `onDestroy()`?

Список практических заданий:

1. Опишите объект "комплексное число" в виде класса `Complex`. Свойства, важные для абстракции, определите самостоятельно.
2. Создайте экземпляр класса. Заполните все свойства данными. Выведите на экран строку, содержащую комплексное число в виде `a+bi`.
3. Создайте два разных экземпляра класса. Заполните их разными данными. Выведите оба числа на экран.
4. Создайте две переменные одного класса. В первую переменную создайте экземпляр класса. А во вторую переменную просто присвойте первую. Убедитесь, что обе переменных "содержат" одну и ту же информацию.
5. Напишите метод, выводящий комплексное число на экран. Вызовите для двух разных чисел.
6. Напишите метод, возвращающий модуль комплексного числа. Выведите на экран модули для двух разных чисел.
7. Напишите метод, принимающий целое число. Сравните модуль вашего комплексного числа с введенным параметром. Если модуль меньше, метод должен вернуть `true`, иначе - `false`.
8. Напишите конструктор класса, присваивающий комплексному числу нули.
9. Напишите конструктор класса, принимающий в качестве параметров два числа и присваивающий их соответственно действительной и мнимой части числа.



10. Перепишите конструктор из предыдущего задания так, чтобы названия принимаемых им параметров СОВПАДАЛИ с названиями свойствами класса. Выполните корректное присвоение принятых значений свойствам при помощи служебного слова `this`.
11. Создайте массив из 20 элементов.
12. Заполните его случайными числами. (Как сгенерировать случайное число в Java, найдите самостоятельно -Google вам в помощь!)
13. Выведите массив на экран.
14. Создайте двумерный массив 4x5 и заполните случайными числами.
15. Транспонируйте двумерный массив.
16. Выведите результат на экран в виде таблицы 5x4.
17. Создайте три строки, содержащие ваши фамилию, имя и отчество.
18. Создайте ещё одну строку и добавьте в нее три предыдущие строки через пробелы.
19. Выведите результат на экран.
20. Выведите на экран 15 символ из строки.
21. Выведите на экран отрезок строки с 5 по 13 символы.
22. Напишите класс `MyMatrix`. Ограничьте доступ к переменным, задающим высоту, ширину и массив, в котором будут храниться элементы матрицы.
23. Напишите конструктор, который принимает число, создает квадратную матрицу соответствующего размера, и заполняет ее нулями.
24. Создайте методы, позволяющие узнать ширину и высоту матрицы.
25. Добавьте методы, позволяющие задать значение отдельному элементу матрицы и получить значение отдельного элемента матрицы.
26. Напишите метод, который выводит матрицу на экран.
27. Перегрузите конструктор матрицы - создайте версии конструктора, которые:
Будет создавать прямоугольную матрицу с заданными высотой и шириной.
Будет создавать прямоугольную матрицу и заполнять ее заданным числом.
Будет создавать копию уже имеющейся матрицы.
28. Напишите метод, который будет добавлять к матрице число. Перегрузите его, чтобы можно было добавить матрице другую матрицу.
29. Напишите методы для транспонирования матрицы, перемножения матриц и умножения матрицы самой на себя.
30. Создайте класс, описывающий точку на числовой прямой. У точки должна быть одна координата. Создайте конструктор по умолчанию, создающий точку с нулевыми координатами, и конструктор, принимающий координаты точки. Напишите метод, выводящий координаты точки на экран. Напишите метод, позволяющий вычислить расстояние от точки до начала координат. Напишите метод, возвращающий расстояние между двумя точками.
31. Создайте подкласс, унаследованный от класса точки, описывающий точку на координатной плоскости. Переопределите для него все методы и конструкторы.
32. Создайте подкласс, унаследованный от класса двумерной точки, описывающий точку в трехмерном пространстве. Переопределите для него все методы и конструкторы.
33. Сделайте начальный класс точки абстрактным, а в двумерных и трехмерных точках - реализуйте все методы. Попробуйте методы двумерной точки пометить модификатором `final`. Обратите внимание на то, какие проблемы это вызовет в методах трехмерных точек.
34. Вынесите классы точек в отдельный пакет.
35. Создайте интерфейс `Comparable`, в котором определите функции `lessThan` и `moreThan`. Реализуйте этот интерфейс в классах точек. Функции должны сравнивать текущую точку с другой и возвращать значения `true` или `false`. Меньшей считается та точка, которая ближе к началу



координат. При помощи реализованных функций `lessThan` и `moreThan` сравните несколько точек между собой.

36. Создайте приложение, которое отображает раскрывающийся список, кнопку "Найти студентов!" и поле для отображения текста. В списке нужно задать выбор из учебных групп (например, "МП-201, МП-202, МТ-201 и т.д.) При нажатии на кнопку должен отображаться список студентов соответствующей группы.

37. Измените приложение из предыдущей темы так, чтобы список студентов выбранной группы отображался в новой активности.

38. *Добавьте на макет второй активности кнопку "Переслать список". Сделайте так, чтобы при нажатии на эту кнопку, запускалась активность для отправки сообщений через какой-нибудь мессенджер (ВК, ФБ, Вайбер и т.п.)

39. Напишите программу-таймер.

40. Напишите приложение, состоящее из одной активности. Переопределите все методы изменения состояния активности так, чтобы при каждом изменении состояния на экране отображались разные картинки.



4. Порядок проведения и критерии оценивания промежуточной аттестации

4.1. Порядок проведения промежуточной аттестации

На дифференцированном зачёте будет предложен один теоретический вопрос и одно практическое задание.

4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств

Критерии оценивания теоретического вопроса на зачете:

Отлично/зачтено - Обучающийся отлично знает материал, может привести примеры, реализующие теоретические концепции, из любого языка программирования. Обучающийся практически не допускает ошибок.

Хорошо/зачтено - Обучающийся хорошо знает материал, частично может проиллюстрировать теоретические концепции примерами из языка программирования. Обучающийся допускает незначительные ошибки.

Удовлетворительно/зачтено - Обучающийся знаком с материалом, но упускает важные детали. Не может составить работающий пример программы.

Неудовлетворительно/не зачтено - Обучающийся не знает основных положений вопроса, не ориентируется в основных понятиях, излагает материал с трудом. Не может составить работающий пример программы.

Критерии оценивания практической задачи на зачете:

Отлично/зачтено - Программа написана самостоятельно, компилируется и запускается. Программа полностью выполняет задачу. Студент уверенно отвечает на вопросы об особенностях работы программы.

Хорошо/зачтено - Программа написана с помарками. Студент может самостоятельно исправить, скомпилировать и запустить программу после указания на помарки. Студент отвечает на простые вопросы об особенностях работы программы.

Удовлетворительно/зачтено - Программа написана с грубыми ошибками. Студент может скомпилировать и запустить программу только после указания на конкретные допущенные ошибки и их места в коде. Студент не может ответить на вопросы об особенностях работы программы.

Неудовлетворительно/не зачтено - Программа написана с грубыми ошибками. Студент не в состоянии исправить ошибки, даже после указания на причину возникновения ошибки и конкретное место в коде.

4.3. Результаты промежуточной аттестации и уровни сформированности компетенций

При подведении итогов учитываются результаты зачета.

«Отлично» («5») – студент глубоко и полно владеет содержанием учебного материала и понятийным аппаратом; умеет связывать теорию с практикой, иллюстрировать примерами, фактами, данными научных исследований; обозначает межпредметные связи, делает выводы логично, четко. Ясно и кратко излагает ответы на поставленные вопросы; умеет обосновывать свои суждения и профессионально-личностную позицию по излагаемому вопросу. Ответ носит самостоятельный характер.

«Хорошо» («4») – ответ студента соответствует указанным выше критериям, но содержание ответа имеет отдельные неточности (несущественные ошибки) в изложении теоретического и практического материала, отличается меньшей обстоятельностью, глубиной и полнотой; допущенные ошибки исправляются студентом после дополнительных вопросов экзаменатора.



«Удовлетворительно» («3») – студент обнаруживает знание и понимание основных положений учебного материала, но излагает его неполно, непоследовательно, допускает неточности и существенные ошибки в определении понятий, формулировке положений, не привлекает для аргументации ответа основные положения исследовательских, концептуальных и нормативных документов, не умеет обосновать свои суждения; наблюдается нарушение логики изложения. Ответ отличается низким уровнем самостоятельности.

«Неудовлетворительно» («2») – студент имеет разрозненные, бессистемные знания: не умеет выделять главное и второстепенное; допускает ошибки в определении понятий, формулировке теоретических положений, искажающие их смысл; не ориентируется в нормативно-концептуальных, программно-методических, исследовательских материалах, беспорядочно и неуверенно излагает материал; не умеет соединять теоретические положения с педагогической практикой; не умеет применять знания для обоснования и объяснения фактов.

Особенности проведения процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья обозначены в рабочей программе дисциплины (модуля).

Уровни сформированности компетенций определяется следующим образом:

1. Продвинутый уровень сформированности компетенций соответствует оценке «отлично/зачтено» (85-100 баллов):

Обучающийся владеет знаниями предмета в полном объеме учебной программы, достаточно глубоко осмысливает дисциплину; самостоятельно, в логической последовательности и исчерпывающе отвечает на все вопросы, подчеркивает при этом самое существенное, умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделять в нем главное: устанавливать причинно-следственные связи; четко формирует ответы.

2. Базовый уровень соответствует оценке «хорошо/зачтено» (75-84 баллов):

Обучающийся владеет знаниями дисциплины почти в полном объеме программы (имеются пробелы знаний только в некоторых, особенно сложных разделах); самостоятельно и отчасти при наводящих вопросах дает полноценные ответы на вопросы; не всегда выделяет наиболее существенное, не допускает вместе с тем серьезных ошибок в ответах.

3. Пороговый уровень соответствует оценке «удовлетворительно/зачтено» (60-74 баллов):

Обучающийся владеет основным объемом знаний по дисциплине; проявляет затруднения в самостоятельных ответах, оперирует неточными формулировками; в процессе ответов допускает ошибки по существу вопросов.

4. Низкий уровень соответствует оценке «неудовлетворительно/не зачтено»:

Обучающийся не освоил обязательного минимума знаний предмета, не способен ответить на вопросы даже при дополнительных наводящих вопросах экзаменатора.

