

| | | | |
|--|---|---|--------|
| Документ подписан простой электронной подписью Информация о владельце: ФИО: Таскаев Сергей Валерьевич Должность: Ректор Дата подписания: 07.04.2025 16:06:07 Уникальный программный ключ: 04c19ed8bf09857b6cb771486b9a8788b8722727 | МИНОВ НАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») | Рабочая программа дисциплины "Объектно-ориентированное программирование" по направлению подготовки (специальности) 02.03.02 "Фундаментальная информатика и информационные технологии" направленности (профиль) Математические и алгоритмические основы интеллектуальных систем ФГБОУ ВО «ЧелГУ» | стр. 1 |
|--|---|---|--------|

Рабочая программа дисциплины (модуля)*

Объектно-ориентированное программирование

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

Год(ы) набора

*Рабочая программа дисциплины (модуля) адаптирована для инклюзивного обучения инвалидов и лиц с ограниченными возможностями здоровья

Челябинск 2023 г.



Содержание

1. Цели освоения дисциплины
2. Место дисциплины в структуре ОПОП
3. Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля)
4. Объем дисциплины (модуля)
5. Структура и содержание дисциплины (модуля)
6. Фонд оценочных средств
 - 6.1. Перечень видов оценочных средств
 - 6.2. Типовые контрольные задания и иные материалы для текущей аттестации
 - 6.3. Типовые контрольные вопросы и задания для промежуточной аттестации
 - 6.4. Критерии оценивания
7. Учебно-методическое и информационное обеспечение дисциплины (модуля)
 - 7.1. Рекомендуемая литература
 - 7.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"
 - 7.3. Перечень информационных технологий
8. Материально-техническое обеспечение дисциплины (модуля)
9. Методические указания для обучающихся по освоению дисциплины (модуля)
10. Специальные условия освоения дисциплины обучающимися с инвалидностью и ограниченными возможностями здоровья



1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В дисциплине изучаются основные понятия объектно-ориентированного программирования и их реализация на примере языка Java.

Цель дисциплины - сформировать у студентов фундаментальное понимание принципов и методов объектно-ориентированного программирования.

Задачи - в процессе обучения студенты должны приобрести знания о правилах проектирования и реализации классов в языке программирования. Их взаимосвязях и применении в современных информационных технологиях.

Результаты изучения дисциплины направлены на достижение следующих индикаторов:

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.

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 обеспечения, структур данных, баз данных, программных интерфейсов.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Цикл (раздел) ОПОП: Б1.В.1.02

2.1 Требования к предварительной подготовке обучающегося:

Изучению дисциплины должно предшествовать получение студентами знаний по технологиям программирования и информатике: синтаксис языков программирования, создание и реализациях алгоритмов.

Технология программирования

Информатика

2.2 Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:

Дисциплина является одной из основных для последующего изучения дисциплин, связанных использованием современных информационных технологий и разработкой пользовательских приложений.

Интернет-технологии

Компьютерная графика

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

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

Знать:

основные понятия ООП; простейшие паттерны программирования (взаимосвязь классов).



Рабочая программа дисциплины "Объектно-ориентированное программирование" по направлению подготовки (специальности) 02.03.02 "Фундаментальная информатика и информационные технологии" направленности (профилю) Математические и алгоритмические основы интеллектуальных систем ФГБОУ ВО «ЧелГУ»

стр. 4

Уметь:

проектировать и реализовывать простейшие классы; проектировать и реализовывать различные конструкции из связанных классов.

Владеть:

навыками использования средств настройки отдельных элементов в рамках платформы Android.

ПК-3: Способность к разработке требований и проектированию программного обеспечения на основе применения базовых математических знаний и информационных технологий при решении проектно-технических и прикладных задач

Знать:

современные инструментальные и вычислительные средства ООП

Уметь:

создавать простые приложения на основе современной ИТ-платформы с использованием библиотеки классов данной платформы и разработкой собственных классов.

Владеть:

навыками использования среды разработки Java; платформы для разработки мобильных приложений Android на базовом уровне.

В результате освоения дисциплины обучающийся должен

| | |
|---------------------|---|
| 3.1 Знать: | |
| 3.1.1 | - основные понятия ООП; простейшие паттерны программирования (взаимосвязь классов); |
| 3.1.2 | - современные инструментальные и вычислительные средства ООП. |
| 3.2 Уметь: | |
| 3.2.1 | - проектировать и реализовывать простейшие классы; проектировать и реализовывать различные конструкции из связанных классов; |
| 3.2.2 | - создавать простые приложения на основе современной ИТ-платформы с использованием библиотеки классов данной платформы и разработкой собственных классов. |
| 3.3 Владеть: | |
| 3.3.1 | - использования средств настройки отдельных элементов в рамках платформы Android; |
| 3.3.2 | - использования среды разработки Java; платформы для разработки мобильных приложений Android на базовом уровне. |

4. ОБЪЕМ ДИСЦИПЛИНЫ (МОДУЛЯ)

| | |
|---|--|
| Общая трудоемкость | 6 ЗЕТ |
| Часов по учебному плану : 216 в том числе : аудиторные занятия : 68 самостоятельная работа : 141,1 : контактная работа: 74,9 ИКР: 6,9 | Виды контроля в семестрах: зачеты с оценкой 3 |

5. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

| Код занятия | Наименование разделов и тем /вид занятия/ | Семестр / Курс | Часов | Литература |
|-------------|---|----------------|-------|-------------------------|
| | Раздел 1. Введение в ООП | | | |
| 1.1 | Введение в ООП. /Лек/ | 3 | 2 | Л1.1 Л1.2 Л1.3Л2.1 Л2.2 |
| 1.2 | Инструменты ООП /Лаб/ | 3 | 2 | Л1.1 Л1.2 Л1.3Л2.1 Л2.2 |



| | | | | |
|---|--|---|-----|--|
| Рабочая программа дисциплины "Объектно-ориентированное программирование" по направлению подготовки (специальности) 02.03.02 "Фундаментальная информатика и информационные технологии" направленности (профилю) Математические и алгоритмические основы интеллектуальных систем ФГБОУ ВО «ЧелГУ» | | | | стр. 5 |
| 1.3 | Изучение теоретического материала. Установка необходимых программ на свой компьютер. Создание простейшего тестового приложения. /Ср/ | 3 | 5,1 | Л1.1 Л1.2 Л1.3Л2.1 Л2.2 |
| Раздел 2. Объектно-ориентированный язык программирования Java | | | | |
| 2.1 | Основы проектирования и разработки классов на примере Java /Лек/ | 3 | 6 | Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э2 Э3 Э4 |
| 2.2 | Наследование в ООП на примере Java /Лек/ | 3 | 6 | Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э2 Э3 Э4 |
| 2.3 | Интерфейсы и другие паттерны программирования /Лек/ | 3 | 4 | Л1.1 Л1.2 Л1.3Л2.1 Л2.2 |
| 2.4 | Введение в классы, объекты, методы /Лаб/ | 3 | 4 | Л1.1 Л1.2 Л1.3Л2.1 Л2.2 |
| 2.5 | Массивы и строки в Java /Лаб/ | 3 | 4 | Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э1 Э2 Э3 |
| 2.6 | Управление классом в Java /Лаб/ | 3 | 4 | Л1.1 Л1.2 Л1.3Л2.1 Л2.2 |
| 2.7 | Наследование в Java /Лаб/ | 3 | 4 | Л1.1 Л1.2 Л1.3Л2.1 Л2.2 |
| 2.8 | Пакеты и интерфейсы в Java /Лаб/ | 3 | 4 | Л1.1 Л1.2 Л1.3Л2.1 Л2.2 |
| 2.9 | Разработка классов. Применение технологий наследования, интерфейсов, пакетов. Изучение особенностей реализации массивов и строк в Java. /Ср/ | 3 | 78 | Л1.1 Л1.2 Л1.3Л2.1 Л2.2 |
| Раздел 3. Использование методов ООП на примере платформы Android | | | | |
| 3.1 | Применение средств ООП на примере платформы Android /Лек/ | 3 | 16 | Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э1 |
| 3.2 | Разработка простого приложения на Android /Лаб/ | 3 | 4 | Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э1 |
| 3.3 | Разработка приложения с несколькими активностями /Лаб/ | 3 | 4 | Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э1 |
| 3.4 | Изучение жизненного цикла активности в приложении на Android /Лаб/ | 3 | 4 | Л1.1 Л1.2 Л1.3Л2.1 Л2.2 Э1 |
| 3.5 | Изучение технологий создания приложения в Android. Создание интерфейса приложения. Обработка нажатия на кнопку и других событий. Применение и настройка активностей. Использование методов жизненного цикла активности. /Ср/ | 3 | 58 | Л1.1 Л1.2 Л1.3Л2.1 Л2.2 |
| Раздел 4. Иная контактная работа | | | | |
| 4.1 | Индивидуальные консультации, текущий контроль /ИКР/ | 3 | 6,9 | |

6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

6.1. Перечень видов оценочных средств

Контрольная работа
Зачет

6.2. Типовые контрольные задания и иные материалы для текущей аттестации

Сгруппированные по темам вопросы для самоподготовки и практические упражнения в электронном курсе Moodle moodle.uio.csu.ru/course/view.php?id=1548



Список вопросов:

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.

6.3. Типовые контрольные вопросы и задания для промежуточной аттестации

На зачёте вам будет предложен один теоретический вопрос и одно практическое задание. Ответ на вопрос должен состоять из двух частей: вы должны объяснить, зачем существует то или иное понятие (какую проблему позволяет решить) и привести пример использования. Пользоваться учебником, тетрадями, Интернетом, телефонами и



другими вспомогательными материалами будет нельзя. Оцениваться будет сухой остаток в вашей голове. Если вы неправильно отвечаете на вопрос или не можете выполнить задание, получаете дополнительный вопрос или дополнительное задание.

"Отлично" - без дополнительных заданий.

"Хорошо" - одно дополнительное задание.

"Удовлетворительно" - два дополнительных задания.

Список вопросов:

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 использование 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(), 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 MuMatrix. 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, выводющий 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.

6.4. Критерии оценивания

«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.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1. Рекомендуемая литература

7.1.1. Основная литература

| | Авторы, составители | Заглавие | Издательство, год | Ресурс |
|------|---|--|--|--------|
| Л1.1 | Березовская Ю. В., Юфрякова О. А., Вологодина В. Г., Озерова О. В., Куликов Э. Е. | Введение в разработку приложений для ОС Android (https://biblioclub.ru/index.php?page=book&id=428937) | Москва : Национальный Открытый Университет «ИНТУИТ», 2016 | ЭБС |
| Л1.2 | Семакова А. | Введение в разработку приложений для смартфонов на ОС Android: учебное пособие (https://biblioclub.ru/index.php?page=book&id=429181) | Москва : Национальный Открытый Университет «ИНТУИТ», 2016 | ЭБС |
| Л1.3 | Гуськова О. И. | Объектно ориентированное программирование в Java: учебное пособие (https://biblioclub.ru/index.php?page=book&id=500355) | Москва : Московский педагогический государственный университет (МПГУ), 2018 | ЭБС |

7.1.2. Дополнительная литература

| | Авторы, составители | Заглавие | Издательство, год | Ресурс |
|------|---------------------|--|---------------------------------|--------|
| Л2.1 | Баженова И. Ю. | Язык программирования Java: практическое пособие (https://biblioclub.ru/index.php?page=book&id=54745) | Москва : Диалог- МИФИ, 2008 | ЭБС |
| Л2.2 | Киркор М. А. | Технология разработки игровых приложений для операционной системы Android с использованием инструмента UNITY3D: выпускная квалификационная работа: студенческая научная работа (https://biblioclub.ru/index.php?page=book&id=462620) | Екатеринбург : [б. и.], 2016 | ЭБС |

7.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

| | |
|----|--|
| Э1 | Официальный портал для разработчиков Android https://developer.android.com |
| Э2 | Онлайн компилятор Java https://www.onlinegdb.com/online_java_compiler |



| | | |
|--|---|---------|
| Рабочая программа дисциплины "Объектно-ориентированное программирование" по направлению подготовки (специальности) 02.03.02 "Фундаментальная информатика и информационные технологии" направленности (профилю) Математические и алгоритмические основы интеллектуальных систем ФГБОУ ВО «ЧелГУ» | | стр. 11 |
| Э3 | Простые упражнения на Java https://www.w3schools.com/java/default.asp | |
| Э4 | Официальная документация Java https://docs.oracle.com/en/java/javase/16/ | |
| 7.3 Перечень информационных технологий | | |
| 7.3.1 Программное обеспечение | | |
| LMS Moodle | | |
| MS Office365 | | |
| Android Studio | | |
| Java Development Kit | | |
| 7.3.2 Профессиональные базы данных и информационно-справочные системы | | |
| Электронный каталог научной библиотеки ЧелГУ [Электронный ресурс] : база данных / Челяб. гос. ун-т. – Челябинск, 1992 | | |
| eLIBRARY.RU [Электронный ресурс] : научная электронная библиотека [научной периодики на русском языке]. — Москва, [1999-]. - Доступ к полным текстам после регистрации из сети ЧелГУ. – URL: http://elibrary.ru/defaultx.asp . | | |
| Moodle [Электронный ресурс]: система дистанционного обучения : [база данных] / Челяб. гос. ун-т. – Челябинск, [б.г.]. – Доступ из сети ЧелГУ или, после регистрации из сети ун-та, из любой точки, имеющей доступ в интернет. – URL: http://moodle.uio.csu.ru/login/index.php . | | |
| Научная библиотека Челябинского государственного университета [Электронный ресурс] : [сайт] / Челяб. гос. ун-т. – Челябинск, [2001-]. – Режим доступа: http://www.lib.csu.ru/ , свободный. – Загл. с экрана. | | |
| Интернет университет информационных технологий [Электронный ресурс]. – Электрон. дан. – Режим доступа : http://www.intuit.ru/ | | |

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

| |
|---|
| Для реализации дисциплины используются учебные аудитории для проведения занятий лекционного типа, лабораторных занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы. |
| Учебные аудитории укомплектованы специализированной мебелью. |
| Для проведения занятий лекционного типа используется переносное и / или стационарное мультимедийное оборудование (экран, ноутбук или десктоп, проектор). Для обеспечения тематической иллюстрации занятий лекционного типа в образовательном процессе используются цифровые образовательные ресурсы (мультимедийные презентации по всем темам программы). |
| Для проведения практических работ и самостоятельной работы используется компьютерный класс из рабочих станций, объединённых в локальную компьютерную сеть с выходом в Интернет и доступом в электронную информационно-образовательную среду университета, с установленным программным обеспечением. |
| Для самостоятельной работы обучающихся используется также читальный зал научной библиотеки ЧелГУ (первый корпус ЧелГУ) с доступом к различной справочной литературе, энциклопедиям, библиографическим и полнотекстовым базам данных, информационным Интернет-ресурсам. |

9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

| |
|--|
| При проведении лабораторных работ студентам необходимо обращать внимание на тот теоретический (лекционный) материал, который используется в конкретном задании. |
| При проведении лекций и лабораторных работ студенты должны использовать следующие активные и интерактивные формы: |
| - обсуждение вариантов решения в диалоговом режиме, |
| - компьютерное моделирование и практический анализ результатов, |
| - научные дискуссии о современных достижениях в изучаемой области, |
| - взаимопомощь по принципу "сделал сам - помоги товарищу". |
| В каждом семестре студенты выполняют на лабораторных работах индивидуальные задания по решению задач на применение изученных на лекциях методов, результаты проверки которых служат основой для сдачи зачета или |



допуска к экзамену.

При обучении лиц с ограниченными возможностями здоровья электронное обучение, дистанционные образовательные технологии предусматривают возможность приема-передачи информации в доступных для них формах.

Реализация дисциплины с применением электронного обучения, дистанционных образовательных технологий (далее – ЭО, ДОТ) осуществляется на основании «Положения о реализации основных и дополнительных образовательных программ с применением электронного обучения и дистанционных образовательных технологий в федеральном государственном бюджетном образовательном учреждении высшего образования «Челябинский государственный университет», «Положения о порядке зачета обучающимися по основным профессиональным образовательным программам высшего образования в ФГБОУ ВО «ЧелГУ» результатов освоения в организациях, осуществляющих образовательную деятельность, учебных предметов, курсов, дисциплин (модулей), практик, дополнительных образовательных программ» посредством электронной информационно-образовательной среды ФГБОУ ВО «ЧелГУ». В исключительных случаях (форс-мажор и т.п.) при реализации образовательной деятельности с применением ЭО, ДОТ могут применять компоненты, не входящие в перечень электронной информационно-образовательной среды.

10. СПЕЦИАЛЬНЫЕ УСЛОВИЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ ОБУЧАЮЩИМИСЯ С ИНВАЛИДНОСТЬЮ И ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Освоение дисциплины инвалидами и лицами с ограниченными возможностями здоровья осуществляется с использованием специальных технических средств и голо информационных технологий, предоставляемых Ресурсным учебно-методическим центром по обучению инвалидов и лиц с ограниченными возможностями здоровья ЧелГУ по запросу обучающегося.

1. Мобильные специальные технические средства для лиц с нарушениями зрения: портативный компьютер с вводом/выводом шрифтом Брайля с синтезатором речи «EiBraille-W14J G2»; ноутбуки с программной экранного доступа NVDA; электронные увеличители для удаленного просмотра; видеувеличители портативные; тифлоплеер; цифровые диктофоны.

2. Мобильные специальные технические средства для лиц с нарушениями слуха: система свободного звукового поля со встроенной совместимостью с FM-устройствами; радиоклассы «Сонет-PCM» с передатчиком, заушным индуктором и индукционной петлей; система информационная для слабослышащих переносная «Исток» А2 со встроенным плеером – звуковым информатором; документ-камера; программируемые слуховые аппараты индивидуального пользования.

3. Ассистивные информационные технологии: программное обеспечение экранного доступа с синтезом речи NVDA; программы экранного увеличения; программы речевого синтеза для компьютеров и ноутбуков; программы речевого синтеза для мобильных устройств; экранная клавиатура; экранная лупа.

При необходимости для обучающихся с нарушениями зрения на рабочих местах для проведения практических или лабораторных занятий устанавливается специальное программное обеспечение (программа речевой навигации NVDA, речевые синтезаторы, экранные лупы).

В учебные аудитории обеспечивается беспрепятственный доступ для обучающихся инвалидов и обучающихся с ограниченными возможностями здоровья. В каждой аудитории, где обучаются инвалиды и лица с ограниченными возможностями здоровья, предусматривается соответствующее количество мест для обучающихся с учетом нарушений их здоровья.

Для освоения дисциплины инвалидам и лицам с ограниченными возможностями здоровья предоставляется доступ к печатным источникам, имеющимся в научной библиотеке ЧелГУ, с помощью специальных технических средств; доступ к электронным источникам, представленным в форме электронного документа в фонде научной библиотеки ЧелГУ или электронно-библиотечных системах, с помощью специальных технических и программных средств (рабочее место для незрячего пользователя с программным обеспечением экранного доступа с синтезом речи NVDA, рабочее место с компьютерным роллером и клавиатурой Clevy с большими кнопками и с разделяющей клавиши накладкой).

Учебно-методические материалы для обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла,
- в печатной форме шрифтом Брайля.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:



- в печатной форме,
- в форме электронного документа,
- в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

Для инвалидов и лиц с ограниченными возможностями здоровья освоение дисциплины может быть частично или полностью осуществлено с использованием дистанционных образовательных технологий (Moodle, Adobe Connect Pro и пр.).

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья используется индивидуальная работа. Под индивидуальной работой подразумевается две формы взаимодействия с преподавателем: индивидуальная учебная работа (консультации), т.е. дополнительное разъяснение учебного материала и углубленное изучение материала с теми обучающимися, которые в этом заинтересованы, и индивидуальная воспитательная работа. Индивидуальные консультации направлены на индивидуализацию обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или обучающимся с ограниченными возможностями здоровья.

При проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине обеспечивается выполнение следующих дополнительных требований в зависимости от индивидуальных особенностей, обучающихся:

- а) инструкция по порядку проведения процедуры оценивания предоставляется в доступной форме (устно, в письменной форме, в письменной форме шрифтом Брайля, устно с использованием услуг сурдопереводчика);
- б) доступная форма предоставления заданий оценочных средств (в печатной форме, в печатной форме увеличенным шрифтом, в печатной форме шрифтом Брайля, в форме электронного документа, задания зачитываются ассистентом, задания предоставляются с использованием сурдоперевода);
- в) доступная форма предоставления ответов на задания (письменно на бумаге, набор ответов на компьютере, письменно шрифтом Брайля, с использованием услуг ассистента, устно).

При проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями. Эти средства могут быть предоставлены ЧелГУ или могут использоваться собственные технические средства. При необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на задания, процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Проведение процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья допускается с использованием дистанционных образовательных технологий.

