**Методические указания по освоению дисциплины**

**«Программирование на языке Java для анализа данных»   
для специальностей 01.03.02**

**1. Целью** преподавания и изучения дисциплины является обучение студентов языку Java и разработке платформонезависимых программ для анализа данных.

**Задачами дисциплины** является приобретение знаний и умений по:

- основам объектно-ориентированного программирования на языке Java

- работе с библиотекой коллекций Java

- потоковой обработке данных (Stream API), чтение и запись данных из внешних файлов

- основам многопоточного и распределенного программирования

- разработке модульных (Unit) тестов

- основам работы с базами данных из Java

- работе с данными в распределенных отказоустойчивых системах на примере дистрибутива Cloudera Hadoop

- трансформации и анализу данных с помощью SQL-интерфейса Apache Hive

- визуализации данных с помощью библиотеки JFreeChart;

- обзору фреймворков MapReduce, Apache Spark – популярных компонентов современных архитектур для работы с большими данными

**Краткое содержание дисциплины**

Идеология и области применения языка Java. Среда разработки. Структура Java-приложения. Синтаксис языка Java. Типы данных, управляющие конструкции. Объекты и классы. Принципы разработки классов ООП. Инкапсуляция, наследование, полиморфизм, абстрактные классы. Интерфейсы. Обработка исключений. Стандартная библиотека классов java.lang, классы Object, Class, Syctem, Math, классы-оболочки, автоупаковка и автораспаковка, классы String, StringBuffer, StringBuilder, классы Exception, Error. Использование легковесных потоков. Обобщенное программирование. Шаблоны. Коллекции. Функциональные интерфейсы. Потоки ввода-вывода., сериализация объектов. Основы сетевого взаимодействия, интернационализация и локализация. Рефлексия. Взаимодействие с базами данных. Библиотека JFreeChart визуализации графиков. Проект Apache Hadoop для распределенных вычислений, платформа MapReduce. Применение фреймворка Apache Spark для анализа больших данных и машинного обучения.

**2. Компетенции обучающегося, формируемые в результате освоения дисциплины**

|  |  |
| --- | --- |
| Планируемые результаты освоения ОП ВО (компетенции) | Планируемые результаты  обучения по дисциплине |
| ПК-6 Способен применять основные алгоритмические и программные решения в области информационно-коммуникационных технологий и системах искусственного интеллекта, а также участвовать в их разработке | Знает: синтаксис, базовые классы, библиотеки языка Java  Умеет: создавать классы на языке Java для решения задач анализа данных по принципам объектно-ориентированного программирования  Имеет практический опыт: создания консольных и графических приложений и апплетов в IDE Eclipse |

**3. Место дисциплины в структуре ОП ВО**

|  |  |
| --- | --- |
| Перечень предшествующих дисциплин, видов работ учебного плана | Перечень последующих дисциплин, видов работ |
| Программирование на С# для разработки систем искусственного интеллекта | Мобильная и веб-разработка систем искусственного интеллекта, Визуальное программирование для систем искусственного интеллекта |

Требования к «входным» знаниям, умениям, навыкам студента, необходимым при освоении данной дисциплины и приобретенным в результате освоения предшествующих дисциплин:

|  |  |
| --- | --- |
| Дисциплина | Требования |
| Программирование на С# для разработки систем искусственного интеллекта | Знает: основные принципы, концепции и профессиональную лексику языка C# Умеет: применять конструкции, возможности и средства языка C# при разработке программного обеспечения Имеет практический опыт: создания прикладного программного обеспечения средствами объектно-ориентированного программирования языка C# |

**4. Объём и виды учебной работы**

Общая трудоемкость дисциплины составляет 3 з.е., 108 ч., 54,25 ч. контактной работы

|  |  |  |
| --- | --- | --- |
| Вид учебной работы | Всего часов | Распределение по семестрам в часах |
| Номер семестра |
| 6 |
| Общая трудоёмкость дисциплины | 108 | 108 |
| *Аудиторные занятия:* | 48 | 48 |
| Лекции (Л) | 32 | 32 |
| Практические занятия, семинары и (или) другие виды аудиторных занятий (ПЗ) | 0 | 0 |
| Лабораторные работы (ЛР) | 16 | 16 |
| *Самостоятельная работа (СРС)* | 53,75 | 53,75 |
| Подготовка к зачету | 8 | 8 |
| Подготовка к лабораторным работам и тестам | 45,75 | 45.75 |
| Консультации и промежуточная аттестация | 6,25 | 6,25 |
| Вид контроля (зачет, диф.зачет, экзамен) | - | зачет |

**5. Содержание дисциплины**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № раздела | Наименование разделов дисциплины | Объем аудиторных занятий по видам в часах | | | |
| Всего | Л | ПЗ | ЛР |
| 1 | Основы языка Java | 14 | 10 | 0 | 4 |
| 2 | Разработка приложений на Java | 18 | 12 | 0 | 6 |
| 3 | Проекты Java для анализа больших данных | 16 | 10 | 0 | 6 |

**5.1. Лекции**

|  |  |  |  |
| --- | --- | --- | --- |
| № лекции | № раздела | Наименование или краткое содержание лекционного занятия | Кол-во часов |
| 1 | 1 | Идеология и область применения Java. Сравнение с языками C#, C++, Python. Инструментальные средства JDK 11. Структура Java-программы, Apache Maven. Синтаксис языка Java. Типы данных и литералы. Область видимости переменных. | 2 |
| 2, 3 | 1 | Основные концепции ООП: объекты и классы, конструкторы, сообщения, наследование, инкапсуляция, полиморфизм, вложенные, локальные и анонимные классы, модификаторы доступа, сообщения. Принципы проектирования классов. UML-диаграмма классов. | 4 |
| 4,5 | 1 | Стандартная библиотека классов, подключение пакетов. Пакет java.lang: классы Object, Class, Syctem, Math, классы-оболочки, автоупаковка и автораспаковка, классы String, StringBuffer, StringBuilder, классы Exception, Error. | 4 |
| 6 | 2 | Тестирование JUnit. Обобщенное программирование. Рефлексия | 2 |
| 7 | 2 | Коллекции. Функциональные интерфейсы. | 2 |
| 8 | 2 | Ввод/вывод. Сериализация. Работа со строками | 2 |
| 9 | 2 | Легковесные процессы (потоки) | 2 |
| 10 | 2 | Базы данных. Потоковая обработка данных StreamIP | 2 |
| 11 | 2 | Основы сетевого взаимодействия. | 2 |
| 12 | 3 | Библиотека JFreeChart | 2 |
| 13 | 3 | Стек технологий Hadoop для работы с "большими данными". Файловая система NDFS. | 2 |
| 14 | 3 | Модель программирования MapReduce в Hadoop. Технология HаdoopStreaming | 2 |
| 15 | 3 | Обработка реляционных данных в система Hadoop. Технологии Hive, Pig | 2 |
| 16 | 3 | Анализ данных с помощью Spark | 2 |

**5.2. Практические занятия, семинары**

Не предусмотрены

**5.3. Лабораторные работы**

|  |  |  |  |
| --- | --- | --- | --- |
| № занятия | № раздела | Наименование или краткое содержание лабораторной работы | Кол-во часов |
| 1 | 1 | ЛР 1 Среда разработки Eclipse.. Классы. Наследование, полиморфизм. Обработка исключений. | 2 |
| 2 | 1 | ЛР 2 Тестирование JUnit. | 2 |
| 3 | 2 | ЛР 3 Обобщенные типы. Коллекции. Потоки ввода-вывода | 2 |
| 4 | 2 | ЛР 4 Многопоточное программирование | 2 |
| 5 | 2 | ЛР 5 Лямбда-выражения, StreamIP | 2 |
| 6 | 3 | ЛР 6 Работа с базами данных | 2 |
| 7 | 3 | ЛР 7 Распределенные вычисления с применением MapReduce | 2 |
| 8 | 3 | ЛР 8 Обработка реляционных данных с помощью Hive, Pig, MapReduce | 2 |

**6. Рекомендации по изучению дисциплины**

**Для преподавателя**

Обучение проводится последовательно путем чтения лекций с углублением и закреплением полученных знаний в ходе самостоятельной работы с последующим переводом знаний в умения в ходе лабораторных занятий. На лекциях излагаются лишь основные, имеющие принципиальное значение и наиболее трудные для понимания и усвоения вопросы. Теоретические знания, полученные обучающимися на лекциях и при самостоятельном изучении курса по литературным источникам, закрепляются на лабораторных занятиях.

Лабораторные занятия проводятся в форме практикума на компьютере с последующей защитой лабораторной работы с целью выявления полученных знаний, умений, навыков и компетенций.

С целью обеспечения успешного обучения обучающийся должен готовиться к лекции, поскольку она является важнейшей формой организации учебного процесса, поскольку:

* знакомит с новым учебным материалом;
* разъясняет учебные элементы, трудные для понимания;
* систематизирует учебный материал; ориентирует в учебном процессе.

Успешное изучение курса требует от студентов посещения лекций, активной работы на лабораторных занятиях, выполнения всех учебных заданий преподавателя, ознакомления с базовыми учебниками, основной и дополнительной литературой.

**Для студента**

*Лекции.*

Изучение дисциплины требует систематического и последовательного накопления знаний, следовательно, пропуски отдельных тем не позволяют глубоко освоить предмет. Именно поэтому посещение лекция обязательно.

Подготовка студента к предстоящей к лекции включает:

* узнайте тему предстоящей лекции (по тематическому плану, по информации лектора);
* ознакомьтесь с учебным материалом по учебнику и учебным пособиям;
* на отдельные лекции приносить соответствующий материал на бумажных носителях, представленный лектором в электронном курсе (таблицы, графики, схемы). Это позволит сэкономить время на конспектирование лекции. Данный материал будет охарактеризован, прокомментирован, дополнен непосредственно на лекции;
* постарайтесь уяснить место изучаемой темы в своей профессиональной подготовке;
* перед очередной лекцией необходимо просмотреть по конспекту материал предыдущей лекции;
* запишите возможные вопросы, которые вы зададите лектору на лекции.

При затруднениях в восприятии материала следует обратиться к основным литературным источникам. Если разобраться в материале опять не удалось, то обратитесь к лектору (по графику его консультаций) или к преподавателю на практических занятиях

Конспектирование лекции – одна из форм активной самостоятельной работы студентов, требующая навыков и умения кратко, схематично, последовательно и логично фиксировать основные положения, выводы, обобщения, формулировки. Каждая учебная дисциплина как наука использует свою терминологию, категориальный, графический материал, которыми студент должен научиться пользоваться и применять по ходу записи лекции. Последующая работа над текстом лекции воскрешает в памяти ее содержание, позволяет развивать мышление.

Основная задача при слушании лекции – учиться мыслить, понимать идеи, излагаемые лектором. Большую помощь при этом может оказать *конспект*. Передача мыслей лектора *своими словами* помогает сосредоточить внимание, не дает перейти на механическое конспектирование. Механическая запись лекции приносит мало пользы. Ведение конспекта создает благоприятные условия для запоминания услышанного, т.к. в этом процессе принимают участие слух, зрение и рука. Конспектирование способствует запоминанию только в том случае, если студент понимает излагаемый материал.

Известный отечественный педагог В.А. Сухомлинский, рекомендовал учиться думать над конспектом уже на лекции и работать над записями ежедневно хотя бы в течение 2 часов. Он советовал также делить конспект на две графы: в первой кратко записывать изложенные лекции, а во второй – то, над чем надо подумать; сюда же следовало заносить узловые, главные вопросы, над которыми надо подумать постоянно, связывая с этим повседневное чтение. Он подчеркивал, что узловые вопросы предмета будут программой, на основе которой припоминается весь материал.

Важно помнить, что ни одна дисциплина не может быть изучена в необходимом объеме только по конспектам. Для хорошего усвоения курса нужна систематическая работа с учебной и научной литературой, а конспект может лишь облегчить понимание и усвоение материала.

*Практические занятия (семинары) и лабораторные работы.*

Лабораторные занятия завершают изучение наиболее важных тем учебной дисциплины. Они служат для закрепления изученного материала, получения правктических навыков разработки программ на Java. а также для контроля преподавателем степени подготовленности студентов по изучаемой дисциплине.

При подготовке к практическим занятиям и лабораторным работам следует:

* ознакомиться с темой и планом занятия, чтобы выяснить круг вопросов, которые будут обсуждаться на занятии;
* внимательно прочитать материал лекций, относящихся к данному семинарскому занятию, ознакомиться с учебным материалом по учебнику и учебным пособиям;
* выписать основные термины;
* ответить на контрольные вопросы, готовиться дать развернутый ответ на каждый из вопросов;
* уясните, какие учебные элементы остались для вас неясными и постарайтесь получить на них ответ заранее (до лабораторного занятия) во время текущих консультаций преподавателя.

Подготовка к лабораторноц работе (ЛР) включает в себя текущую работу над учебными материалами с использованием конспектов и рекомендуемой основной и дополнительной литературы; групповые и индивидуальные консультации; самостоятельное решение ситуационных задач, изучение литературы.

Работу с литературой рекомендуется делать в следующей последовательности: беглый просмотр (для выбора глав, статей, которые необходимы по изучаемой теме); беглый просмотр содержания и выбор конкретных страниц, отрезков текста с пометкой их расположения по перечню литературы, номеру страницы и номеру абзаца; конспектирование прочитанного. Если самостоятельно не удалось разобраться в материале, необходимо сформулировать вопросы и обратиться за помощью к преподавателю на консультации или ближайшей лекции. Рекомендуется регулярно отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.

Качество учебной работы студентов преподаватель оценивает в зашиты лабораторных работ, выставляя баллы.. Обучающийся имеет право ознакомиться с ними.

Обучающиеся, не отчитавшиеся по каждой не проработанной ими на занятиях теме к началу зачетной сессии, упускают возможность получить положенные баллы за соответствующее контрольное мероприятие.

*Самостоятельная работа студентов*

Важную роль в изучении курса играет самостоятельная работа студента. Она предполагет подготовку к лекциям, лабораторным занятиям путем изучения литературы по теме предстоящего занятия. Оценка сформированности компетенций по дисциплине осуществляется с помощью выполнения самостоятельных заданий к ЛР и тестов. Все задания снабжены методическими указаниями и примерами их выполнения. Поэтому рекомендуется вынимательно изучить эти рекомендации. Всего предусмотрено 8 лабораторных работ и 2 тестов. Полученные в текущем контроле оценки учитываются в общей оценке курса согласно БРС. При необходимости добор баллов может осуществляться путем выполнения дополнительных заданий или рефератов по указанию преподавателя.

*Методика написания рефератов и докладов.*

**Целью** написания рефератов является:

* привитие студентам навыков библиографического поиска необходимой литературы (на бумажных носителях, в электронном виде);
* привитие студентам навыков компактного изложения мнения авторов и своего суждения по выбранному вопросу в письменной форме, научно грамотным языком и в хорошем стиле;
* приобретение навыка грамотного оформления ссылок на используемые источники, правильного цитирования авторского текста;
* выявление и развитие у студента интереса к определенной научной и практической проблематике с тем, чтобы исследование ее в дальнейшем продолжалось в подготовке и написании курсовых и дипломной работы и дальнейших научных трудах.

Основные **задачи** студента при написании реферата:

* с максимальной полнотой использовать литературу по выбранной теме (как рекомендуемую, так и самостоятельно подобранную) для правильного понимания авторской позиции;
* верно (без искажения смысла) передать авторскую позицию в своей работе;
* уяснить для себя и изложить причины своего согласия (несогласия) с тем или иным автором по данной проблеме.

**Требования к содержанию:**

* материал, использованный в реферате, должен относиться строго к выбранной теме;
* необходимо изложить основные аспекты проблемы не только грамотно, но и в соответствии с той или иной логикой (хронологической, тематической, событийной и др.)
* при изложении следует сгруппировать идеи разных авторов по общности точек зрения или по научным школам;
* реферат должен заканчиваться подведением итогов проведенной исследовательской работы: содержать краткий анализ-обоснование преимуществ той точки зрения по рассматриваемому вопросу, с которой Вы солидарны.

*Структура реферата*

* + 1. Начинается реферат с титульного листа.
    2. За титульным листом следует Оглавление. Оглавление - это план реферата, в котором каждому разделу должен соответствовать номер страницы, на которой он находится.
    3. Текст реферата. Он делится на три части: введение, основная часть и заключение.

а) Введение – раздел реферата, посвященный постановке проблемы, которая будет рассматриваться и обоснованию выбора темы.

б) Основная часть – это звено работы, в котором последовательно раскрывается выбранная тема. Основная часть может быть представлена как цельным текстом, так и разделена на главы. При необходимости текст реферата может дополняться иллюстрациями, таблицами, графиками, но ими не следует «перегружать» текст.

в) Заключение – данный раздел реферата должен быть представлен в виде выводов, которые готовятся на основе подготовленного текста. Выводы должны быть краткими и четкими. Также в заключении можно обозначить проблемы, которые "высветились" в ходе работы над рефератом, но не были раскрыты в работе.

Список источников и литературы. В данном списке называются как те источники, на которые ссылается студент при подготовке реферата, так и все иные, изученные им в связи с его подготовкой. Работа, выполненная с использованием материала, содержащегося в одном научном источнике, является явным плагиатом и не принимается. Оформление Списка источников и литературы должно соответствовать требованиям библиографических стандартов. Список использованных источников представляет собой перечень использованных книг, статей, фамилии авторов приводятся в алфавитном порядке, при этом все источники даются под общей нумерацией литературы. В исходных данных источника указываются фамилия и инициалы автора, название работы, место и год издания.

*Объем и технические требования, предъявляемые к выполнению реферата.*

Объем работы должен быть, как правило, не менее 12 и не более 20 страниц. Работа должна выполняться через полуторный интервал 14 шрифтом, размеры оставляемых полей: левое -30 мм, правое -10 мм, нижнее - 20 мм, верхнее - 20 мм. Страницы должны быть пронумерованы. Расстояние между названием части реферата или главы и последующим текстом должно быть равно трем интервалам. Фразы, начинающиеся с «красной» строки, печатаются с абзацным отступом от начала строки, равным 1 см.

При цитировании необходимо соблюдать следующие правила:

* текст цитаты заключается в кавычки и приводится без изменений, без произвольного сокращения цитируемого фрагмента (пропуск слов, предложений или абзацев допускается, если не влечет искажения всего фрагмента, и обозначается многоточием, которое ставится на месте пропуска) и без искажения смысла;
* каждая цитата должна сопровождаться ссылкой на источник, библиографическое описание которого должно приводиться в соответствии с требованиями библиографических стандартов.

# Структура и содержание доклада

Введение – это вступительная часть. Автор должен приложить все усилия, чтобы в этом небольшом по объему разделе показать актуальность темы, раскрыть практическую значимость ее, определить цели и задачи проделанной работы.

Основная часть. В ней раскрывается содержание доклада. Как правило, основная часть состоит из теоретического и практического разделов.

В теоретическом разделе раскрываются история и теория исследуемой проблемы, дается критический анализ литературы и показывается позиция автора.

В практическом разделе излагаются методы, ход, и результаты самостоятельно проведенного исследования (в данном курсе – необязательная часть).

В основной части могут быть также представлены схемы, диаграммы, таблицы, рисунки и т.д.

В заключении содержатся итоги работы, выводы, к которым пришел автор, и рекомендации. Заключение должно быть кратким, обязательным и соответствовать поставленным задачам.

*Требования к оформлению доклада*

Объем доклада может колебаться в пределах 5-15 слайдов в формате Power Point;

Доклад должен быть выполнен грамотно, с соблюдением культуры изложения.

Обязательно должны иметься ссылки на используемую литературу.

*Критерии оценки доклада*

* актуальность темы исследования;
* соответствие содержания теме;
* глубина проработки материала;
* правильность и полнота использования источников;
* соответствие оформления доклада стандартам.

**7. Текущий контроль успеваемости, промежуточная аттестация**

Контроль качества освоения образовательной программы осуществляется в соответствии с Положением о балльно-рейтинговой системе оценивания результатов учебной деятельности обучающихся.

**7.1. Контрольные мероприятия (КМ)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № КМ | Се- местр | Вид контроля | Название контрольного мероприятия | Вес | Макс. балл | Порядок начисления баллов | Учи- тыва- ется в ПА |
| 1 | 6 | Текущий контроль | ЛР 1 Исключения, Классы, Наследование, полиморфизм | 1 | 7 | Оценка суммируется из следующих оценок: 1. Создан проект в Maven – 1 балл 2. Классы создан верно в соответствии с принципами ООП – 1 балл 3. Реализован интерфейс toString() для отображения состояния класса – 1 балл; 4. Реализована обработка исключений – 1 балл 5. Реализованы наследование и полиморфизм классов – 1 балл 6. Программа протестирована. Отчет оформлен в соответствии с требованиями – 1 балл 7. Полные и верные ответы на дополнительные вопросы – 1 балл Максимальная оценка 7 баллов | зачет |
| 2 | 6 | Текущий контроль | ЛР 2 Тестирование JUnit. | 1 | 4 | Оценка суммируется из следующих оценок: 1. Задание выполнено полностью. Созданы все необходимые тесты – 2 балла; созданы не все тесты – 1 балл; Тесты не созданы – 0 баллов 2. Выполнен контрольный пример на тестирование. Отчет оформлен в соответствии с требованиями – 1 балл 3. Полные и верные ответы на дополнительные вопросы – 1 балл Максимальная оценка 4 балла Отчет по заданию высылается в виде документа формата Word или PDF | зачет |
| 3 | 6 | Текущий контроль | ЛР 3 Обобщенные типы. Коллекции. | 1 | 8 | В работе 2 задания. Оценивается отдельно каждое задание. За решение каждой задачи вы можете получить до 2 баллов: задание выполнено полностью и правильно -2 балла; имеются незначительные ошибки -1 балл; приложение не работоспособно – 0 баллов; Ответы на контрольные вопросы должны продемонстрировать понимание механизмов Java, за ответ вы можете получить до 2 баллов: ответ полный и правильный -2 балла; ответ не полный -1 балл; ответ не по существу – 0 баллов. Итого, максимальная оценка - 8 баллов Отчет по заданию высылается в виде документа формата Word или PDF | зачет |
| 4 | 6 | Текущий контроль | ЛР 4 Многопоточное программирование | 1 | 4 | Оценка суммируется из следующих оценок: За решение задачи можно получить до 2 баллов: задание выполнено полностью и правильно -2 балла; имеются незначительные ошибки -1 балл; приложение не работоспособно – 0 баллов; Ответ на контрольный вопрос должен продемонстрировать понимание механизмов Java, за ответ можно получить до 2 баллов: ответ полный и правильный -2 балла; ответ не полный -1 балл; ответ не по существу – 0 баллов. Итого, максимальная оценка - 4 балла  Отчет по заданию высылается в виде документа формата Word или PDF | зачет |
| 5 | 6 | Текущий контроль | ЛР 5 Лямбда-выражения, StreamIP | 1 | 6 | В задании 2 задачи и контрольный вопрос. Оценка за задачу: задание выполнено полностью и правильно -2 балла; имеются незначительные ошибки -1 балл; приложение не работоспособно – 0 баллов; Оценка за ответ на вопрос; ответ полный и правильный -2 балла; ответ не полный – 1 балл, ответ не удовлетворительный – 0 баллов. Максимальная оценка - 6 баллов Отчет по заданию высылается в виде документа формата Word или PDF | зачет |
| 6 | 6 | Текущий контроль | ЛР 6 Работа с базами данных | 1 | 3 | За решение задачи можно получить до 2 баллов: задание выполнено полностью и правильно -2 балла; имеются незначительные ошибки -1 балл; приложение не работоспособно – 0 баллов; Ответ на контрольный вопрос должен продемонстрировать понимание механизмов Java, за ответ можно получить оценку: ответ полный и правильный -1 балл; ответ не удовлетворительный – 0 баллов. Итого, максимальная оценка - 3 балла | зачет |
| 7 | 6 | Текущий контроль | ЛР 7 Распределенные вычисления с применением MapReduce | 1 | 3 | За решение задачи можно получить до 2 баллов: задание выполнено полностью и правильно – 2 балла; имеются незначительные ошибки – 1 балл; приложение не работоспособно – 0 баллов; Ответ на контрольный вопрос должен продемонстрировать понимание архитектуры Hadoop, за ответ можно получить оценку: ответ полный и правильный – 1 балл; ответ не удовлетворительный – 0 баллов. Итого, максимальная оценка – 3 балла   Отчет по заданию высылается в виде документа формата Word или PDF | зачет |
| 8 | 6 | Текущий контроль | ЛР 8 Обработка реляционных данных с помощью Hive, Pig, MapReduce | 1 | 8 | В задании 3 задачи: а) программа на Hive; б) Pig; в) MapReduce За решение каждой задачи можно получить до 2 баллов: задание выполнено полностью и правильно – 2 балла; имеются незначительные ошибки – 1 балл; приложение не работоспособно – 0 баллов; Ответ на контрольный вопрос должен продемонстрировать понимание архитектуры Hadoop, за ответ можно получить оценку: ответ полный и правильный – 2 балла; ответ не полный – 1 балл; ответ не удовлетворительный – 0 баллов. Итого, максимальная оценка – 8 баллов  Отчет по заданию высылается в виде документа формата Word или PDF | зачет |
| 11 | 6 | Текущий контроль | Тест Основы Java | 1 | 20 | В тесте 20 вопросов. За каждый правильный ответ начисляется 1 балл | зачет |
| 12 | 6 | Текущий контроль | Тест по JDBC | 1 | 10 | В тесте 10 вопросов. За каждый правильный ответ добавляется 1 балл | зачет |
| 20 | 6 | Бонус | бонусы за призовое место в олимпиаде по программированию | - | 15 | Бонусные баллы студент может получить за призовое место в олимпиадах по программированию, а также за активность на занятиях и за решение дополнительных задач повышенной сложности. Студент представляет копии документов, подтверждающие победу или участие в предметных олимпиадах по данной дисциплине. Максимально возможная величина бонус-рейтинга 15 | зачет |
| 21 | 6 | Проме- жуточная аттестация | зачет | - | 6 | Билет на зачет содержит два теоретических вопроса. Суммируются оценки за каждый вопрос. Критерии оценки: полные и правильные ответы на вопрос билета и дополнительные вопросы - 3 балла; не полные или не совсем правильные ответы на вопрос билета и дополнительные вопросы - 2 балла; неправильный ответ на вопрос билета, правильные ответы на дополнительные вопросы - 1 балл; неудовлетворительные ответы на вопрос билета и дополнительные вопросы - 0 баллов Итого, максимальная оценка - 6 баллов | зачет |

**7.2. Процедура проведения, критерии оценивания**

|  |  |  |
| --- | --- | --- |
| Вид промежуточной аттестации | Процедура проведения | Критерии оценивания |
| зачет | Зачет проводится в очной форме по билетам. Процедура прохождения зачета не является обязательной если по результатам текущего контроля БРС у студента положительная оценка и он с ней согласен. В каждом билете 2 теоретических вопроса. Студент устно отвечает на вопросы билета. Студент должен находиться в аудитории на протяжении всей процедуры зачета. Число студентов, одновременно находящихся в аудитории, где сдается зачет, не более 8 человек. На подготовку к ответу студенту отводится не более 30 мин. Когда обучающийся будет готов к ответу, ему задаются контрольные вопросы по содержанию билета. Студент должен УСТНО ответить на вопросы билета и дополнительные вопросы по теме билета в течение 5 мин. На этом основании преподаватель выставляет баллы за зачетную работу. Окончательная оценка за курс выставляется согласно БРС. Добор баллов осуществляется путем выполнения дополнительных заданий из КМ1-КМ8. | В соответствии с пп. 2.5, 2.6 Положения |

Фонды оценочных средств по каждому контрольному мероприятию находятся в приложениях к РПД.

**8. Учебно-методическое и информационное обеспечение дисциплины**

**Печатная учебно-методическая документация**

*а) основная литература:*

Не предусмотрена

*б) дополнительная литература:*

Не предусмотрена

*в) отечественные и зарубежные журналы по дисциплине, имеющиеся в библиотеке:*

Не предусмотрены

**Электронная учебно-методическая документация**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Вид  литературы | Наименование ресурса в электронной форме | Библиографическое описание |
| 1 | Основная литература | Электронно-библиотечная система издательства Лань | Программирование на языке Java. Конспект лекций : учебно-методическое пособие / А. В. Гаврилов, С. В. Клименков, Ю. А. Королёва [и др.]. — Санкт-Петербург : НИУ ИТМО, 2019. — 127 с.  https://e.lanbook.com/book/136549 |
| 2 | Дополнительная литература | Электронно-библиотечная система издательства Лань | Пономарчук, Ю. В. Программирование на языке Java : учебное пособие / Ю. В. Пономарчук, И. В. Кузнецов. — Хабаровск : ДВГУПС, 2021. — 103 с.  https://e.lanbook.com/book/259451 |
| 3 | Методические пособия для самостоятельной работы студента | Учебно-методические материалы кафедры | Оленчикова Т.Ю. Методические указания и задания к выполнению лабораторных работ по Java-программированию https://prm.susu.ru/documents/dop/SRS\_po\_Java-2024.zip |
| 4 | Основная литература | Электронно-библиотечная система издательства Лань | Лебедев, А. С. Методы Big Data : учебно-методическое пособие / А. С. Лебедев, Ш. Г. Магомедов. — Москва : РТУ МИРЭА, 2021. — 91 с. https://e.lanbook.com/book/182452 |
| 5 | Дополнительная литература | Электронно-библиотечная система издательства Лань | Коузен, К. Современный Java: рецепты программирования / К. Коузен. — Москва : ДМК Пресс, 2018. — 275 с. https://e.lanbook.com/book/116121 |
| 6 | Дополнительная литература | Электронный архив ЮУрГУ | Цымблер, М.Л. Обзор методов интеграции интеллектуального анализа данных в СУБД // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2019. Т. 8, № 2. С. 32–62. DOI: 10.14529/cmse190203. https://dspace.susu.ru/xmlui/handle/0001.74/34926 |

**9. Вопросы для самоконтроля**

* + - 1. Какие есть типы данных в Java?
      2. Чем отличается объект от примитивных типов данных?
      3. В чем разница передачи параметров по ссылке и по значению?
      4. Что такое JVM, JDK, JRE?
      5. Зачем используют JVM?
      6. Что такое bytecode?
      7. В чем отличие между try-with-resources и try-catch-finally при работе с ресурсами?
      8. Что такое конструкторы? Какие типы знаете?
      9. Что такое конструктор по умолчанию, когда он генерируется автоматически?
      10. Что такое побитовые операции?
      11. Что такое ООП? Назовите принципы с примерами.
      12. В чем преимущества ООП перед процедурным программированием?
      13. Расскажите, какие есть недостатки в ООП?
      14. Расскажите о принципе наследования в ООП? Зачем он нужен?
      15. Дайте определение принципа полиморфизма в ООП? Как работает полиморфизм?
      16. Что такое статический и динамический полиморфизм?
      17. . Дайте определение принципа абстракции в ООП.
      18. Какие элементы языка отвечают за инкапсуляцию?
      19. Какие элементы языка отвечают за наследование?
      20. Какие элементы языка отвечают за полиморфизм?
      21. Что такое SOLID? Приведите примеры.
      22. Что такое перегрузки (overloading) метода?
      23. Что такое переопределение (override) метода?
      24. Что такое класс, объект, интерфейс?
      25. Какие элементы могут содержать класс?
      26. Дайте определение объекта.
      27. Расскажите о наследовании в Java. Какие особенности использования ключевого слова super?
      28. Что такое сигнатура метода? Приведите примеры правильных и неправильных сигнатур.
      29. Можно ли в конструкторе применять return?
      30. Можно ли в конструкторе выкинуть исключение (exception)?
      31. Из каких элементов состоит заголовок класса? Напишите пример.
      32. Из каких элементов состоит заголовок метода? Напишите пример.
      33. Создайте в объекте-наследнике конструктор по умолчанию, если в базовом классе он не определен (но определен другой конструктор).
      34. Когда применяется ключевое слово this?
      35. Что такое инициализатор?
      36. Для наследования класса public class Child extends Parent напишите порядок инициализации объекта.
      37. Как из класса-потомка вызвать метод класса-родителя?
      38. Какие знаете отношения между классами (объектами)?
      39. Какие ассоциативные связи между объектами вы знаете?
      40. Что такое модификаторы доступа в Java? Назовите их. Для чего используются?
      41. Назовите основную особенность статических переменных и методов.
      42. Какие основные ограничения действуют на статические переменные и методы?
      43. Что означает ключевое слово static? Может ли статический метод быть переопределенным или перегруженным?
      44. Может ли метод быть статическим и абстрактным одновременно?
      45. Можно ли использовать статические методы внутри обычных? Наоборот? Почему?
      46. Что означает ключевое слово final?
      47. Что такое abstract? Абстрактный класс? Абстрактный метод?
      48. Что такое interface? Может ли быть final interface?
      49. В чем разница между абстрактным классом и интерфейсом в Java?
      50. Где можно инициализировать статические поля?
      51. Что такое анонимные классы?
      52. . Какие модификаторы доступа могут быть у класса?
      53. Назначение модульного тестирования.
      54. Принципы модульного тестирования.
      55. Отличительные особенности библиотеки модульного тестирования JUnit.
      56. Понятия модульного теста и тестируемого модуля. Последовательность действий для организации unit-тестирования.
      57. Что такое коллекции? Перечислите типы коллекции.
      58. Чем отличается ArrayList от LinkedList? Приведите достоинства и недостатки каждой из них.
      59. Как устроена HashMap?
      60. Роль equals и hashCode?
      61. Расскажите об итераторах и об их применении.
      62. Какая иерархия коллекций в Java Collection Framework?
      63. Какое внутреннее строение ArrayList?
      64. Какое внутреннее строение LinkedList?
      65. Какое внутреннее строение HashMap?
      66. Чем отличается ArrayList от LinkedList?
      67. Чем отличается ArrayList от HashSet?
      68. Как отсортировать коллекцию элементов?
      69. Зачем в Java такое многообразие имплементации динамического массива?
      70. Какие средства для работы с многопоточностью знаете?
      71. Что такое процесс и поток? Чем отличается процесс от потока?
      72. Расскажите о синхронизации между потоками. Для чего используют методы wait(), notify() - notifyAll(), join()?
      73. Как остановить поток?
      74. Как между потоками обмениваться данными?
      75. В чем заключается отличие класса Thread от интерфейса Runnable?
      76. Есть потоки Т1, Т2 и Т3. Как реализовать их последовательное выполнение?
      77. Опишите жизненный цикл потока.
      78. Если объявить метод synchronized, то какой эффект будет этим достигнут?
      79. Опишите механизмы синхронизации потоков
      80. Как работают static synchronized методы?
      81. Если один поток начал исполнение synchronized-блока, указав ссылку на некий объект, может ли другой поток обратиться к полю этого объекта? К методу?
      82. Почему метод wait требует обработки InterruptedException, а методы notify и notifyAll– нет?
      83. Может ли поток никогда не выйти из метода wait, даже если будет вызван метод notify? notifyAll?
      84. Какие средства для работы с многопоточностью знаете?
      85. Что такое процесс и поток? Чем отличается процесс от потока?
      86. Расскажите о синхронизации между потоками. Для чего используют методы wait(), notify() - notifyAll(), join()?
      87. Как остановить поток?
      88. Как между потоками обмениваться данными?
      89. В чем заключается отличие класса Thread от интерфейса Runnable?
      90. Есть потоки Т1, Т2 и Т3. Как реализовать их последовательное выполнение?
      91. Опишите жизненный цикл потока.
      92. Если объявить метод synchronized, то какой эффект будет этим достигнут?
      93. Опишите механизмы синхронизации потоков
      94. Как работают static synchronized методы?
      95. Если один поток начал исполнение synchronized-блока, указав ссылку на некий объект, может ли другой поток обратиться к полю этого объекта? К методу?
      96. Почему метод wait требует обработки InterruptedException, а методы notify и notifyAll– нет?
      97. Может ли поток никогда не выйти из метода wait, даже если будет вызван метод notify? notifyAll?
      98. Что такое JDBC?
      99. Как осуществляется соединение?
      100. Что передается в стороке URL?
      101. Что делает класс Statement?
      102. Какие есть типы запросов и классы для их выполнения?
      103. Что такое большие данные?
      104. Объясните пять признаков больших данных
      105. Сколько форматов ввода существует в Hadoop?
      106. Каковы основные функции Hadoop?
      107. В каких трех режимах может работать Hadoop?
      108. В каких операционных системах работает Hadoop?
      109. Каковы компоненты HDFS?
      110. Что такое Hadoop Map Reduce?
      111. Как работает Hadoop MapReduce?
      112. Объясните, что такое спекулятивное исполнение?
      113. Объясните, в чем разница между входным разделением и блоком HDFS?
      114. Укажите, в чем разница между СУБД и Hadoop?
      115. Укажите, какие компоненты данных использует Hadoop?
      116. Укажите, какие наиболее распространенные форматы ввода определены в Hadoop?
      117. Перечислите три файла конфигурации Hadoop.