

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Баламирзоев Назим Лиодинович
Должность: Ректор
Дата подписания: 25.03.2026 16:00:30
Уникальный программный ключ:
5cf0d6f89e80f49a334f6a4ba58e91f3326b9926



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

Институт кибербезопасности и цифровых технологий

Региональный партнёр

ФГБОУ ВО

«Дагестанский государственный технический университет»



1. .06

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Направленность (профиль подготовки): «Прикладной искусственный интеллект»

Квалификация выпускника бакалавр

Форма обучения очная

Махачкала 2023

ПАСПОРТ фонда оценочных средств

по дисциплине Б1.В.06 Разработка кроссплатформенных приложений

1. Результаты обучения по дисциплине:

Код	Наименование компетенции	Индикатор достижения компетенции (закрепленный за дисциплиной)	В результате освоения дисциплины обучающийся должен:	Другая дисциплина (дисциплины)/практика, участвующая в формировании компетенции
ПК-1	Способен проектировать программное обеспечение вычислительных машин и сетевого оборудования	ПК-1.2. Создает варианты реализации компонент ПО на основе анализа предъявляемых требований	<p style="text-align: center;">Знать: типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения</p> <p style="text-align: center;">Уметь: использовать существующие типовые решения и шаблоны проектирования программного обеспечения</p> <p style="text-align: center;">Владеть: инструментами разработки компонент ПО</p>	<p>Базы данных и знаний программно-аппаратных комплексов вычислительной техники</p> <p>Web-программирование</p> <p>Технологии сети Internet</p> <p>Высокопроизводительные вычисления</p> <p>Разработка интеллектуальных компонентов программного обеспечения</p> <p>Технологии программирования</p> <p>Средства отладки сетевых приложений</p> <p>Учебная (ознакомительная) практика</p> <p>Производственная (проектно-технологическая) практика</p>

2. Программа оценивания контролируемой компетенции:

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции/ индикатора	Наименование оценочного средства
1	Тема 1.1. Вводная лекция	ПК-1.2	Защита лабораторных работ Вопросы экзамена
2	Тема 2.1. Операции и операторы в языке Java.	ПК-1.2	Защита лабораторных работ Защита курсовой работы Вопросы экзамена
3	Тема 2.2. Реализация принципов объектно-ориентированного подхода в Java	ПК-1.2	Защита курсовой работы Защита лабораторных работ Вопросы экзамена
4	Тема 2.3. Элементы хранения наборов данных	ПК-1.2	Защита лабораторных работ Защита курсовой работы Вопросы экзамена
5	Тема 2.4. Обработка исключительных ситуаций	ПК-1.2	Защита лабораторных работ Защита курсовой работы Вопросы экзамена
6	Тема 2.5. Ввод/вывод в Java	ПК-1.2	Защита лабораторных работ Защита курсовой работы Вопросы экзамена
7	Тема 2.6. Многопоточность в Java	ПК-1.2	Защита лабораторных работ Защита курсовой работы Вопросы экзамена
8	Тема 3.1. Перспективы развития языка Java	ПК-1.2	Защита курсовой работы Вопросы экзамена
Форма промежуточной аттестации в 6 семестре – зачет с оценкой, курс. работа.			

Вопросы (задания) для зачета с оценкой

по дисциплине Б1.В.06 Разработка кроссплатформенных приложений

Вопросы:

1. Сравнение языка C++ с языком Java. Простейшая программа на Java.
2. Операции и операторы в языке Java. Операции сравнения. Операции инкремента, декремента. Расширенные операции присваивания.
3. Операции и операторы в языке Java. Логические операции. Операции сдвига. Операция приведения типов. Оператор — выражение.
4. Операции и операторы в языке Java. Условный оператор. Операторы цикла. Оператор цикла "foreach". Операторы break и continue.
5. Реализация принципов объектно-ориентированного подхода в Java. Ссылки на объекты. Элементарные типы данных.
6. Область видимости и время жизни переменных. Область видимости и время жизни объектов. Описание методов класса. Передача параметров.
7. Наследование классов. Класс Object. Инициализация полей при наследовании классов. Модификаторы доступа при наследовании. Преобразования типов (классов) при наследовании.
8. Полиморфизм. Ключевое слово final. Абстрактные классы. Интерфейсы.
9. Вложенные классы. Анонимные вложенные классы. Статические вложенные классы.
10. Массивы в Java. Массивы объектов. Многомерные массивы. Присваивание и копирование массивов.
11. Работа со строками.
12. Коллекции объектов Java. Итераторы.
13. Коллекции-списки.
14. Коллекции-множества.
15. Задание порядка элементов в коллекциях. Интерфейс Comparator.
16. Коллекции-ассоциативные массивы.
17. Обработка исключительных ситуаций. Поведение программы при возникновении исключения.
18. Структура и использование блока перехвата исключений. Классы исключительных ситуаций.
19. Механизм контроля перехвата исключений. Генерация исключительных ситуаций.
20. Создание собственных классов исключительных ситуаций. Исключительные ситуации и наследование.
21. Ввод/вывод в Java. Иерархия InputStream. Иерархия Reader.
22. Ввод/вывод в Java. Иерархия OutputStream. Иерархия Writer.

23. Ввод/вывод в Java. Класс RandomAccessFile. Класс File.
24. Сериализация объектов. Интерфейс Serializable. Классы ObjectOutputStream и ObjectInputStream.
25. Многопоточность в Java. Класс Thread. Интерфейс Runnable.
26. Завершение процесса и демоны. Завершение потоков. Механизм Interruption.
27. Метод sleep. Метод yield. Метод join. Приоритеты потоков.
28. Блокировки. Методы wait, notify, notifyAll.

Задания:

1. Дан следующий код:

```
abstract class A{
    public abstract void print();
}
class B extends A{
    public void print(){
        System.out.println("B");
    }
}
```

Можно ли унаследовать от класса B абстрактный класс? Если да, то что нужно сделать с методом print(): его обязательно надо переопределить, его можно не переопределять, можно ли его определить как abstract?

2. Дан следующий код:

```
abstract class A {
    int p1;
    A() {
        p1 = 1;
    }
    ...
}
class B extends A {
}
```

- 1) Можно ли описать конструкторы в абстрактном классе?
- 2) Может ли быть класс абстрактным без единого абстрактного метода? А наоборот?
- 3) При определении метода в интерфейсе подразумевается модификатор public и abstract. Можно ли при определении интерфейса явно указать модификатор abstract?

3. Дан следующий метод:

```
public void method() throws IllegalArgumentException,IOException,IOException,IOException {
    //...
}
```

Возникнет ли ошибка компиляции? Можно ли указывать `unchecked` исключения в секции `throws`? Что будет, если в секции `throws` одно и то же исключение указано несколько раз? Можно ли смешивать `checked` и `unchecked` исключения?

4. Дан следующий код:

```
1: class Formatter {
2:   public String format(String value) {
3:     return "["+value+"];
4:   }
5: }
6:
7: public class TestNPE {
8:   public static String handle(Formatter f, String s) {
9:     if(s.isEmpty()) {
10:      return "(none)";
11:    }
12:    return f.format(s.trim());
13:  }
14: }
```

Откуда-то был вызван метод `handle` с какими-то параметрами, и было получено:

```
Exception in thread "main" java.lang.NullPointerException
    at TestNPE.handle(TestNPE.java:12)
```

В чём причина исключения — в `f` или `s`? Чтобы поменялось, если бы в девятой строчке было бы написано так:

```
if("".equals(s))
```

5. Какой из предложенных ниже методов не вызовет ошибку компиляции?

- 1) **public static double** `sqr(double arg)` {
 }
}
- 2) **public static double** `sqr(double arg)` {
 while (**true**);
}
- 3) **public static double** `sqr(double arg)` {
 if (`System.currentTimeMillis() % 2 == 0`) {
 return `arg * arg`;
 }
}
- 4) **public double** `sqr(double arg)` {
 int `k = 1`;

```
        return k;
    }
5) public static double sqr(double arg) {
    throw new RuntimeException();
}
```

6. Как сделать Singleton потоко безопасным?

7. Дан такой код:

```
class Test{
    public static void main(String[] args){
        Map map = new TestMap();
        for(Object o: map.keySet()){
            System.out.println(o);
        }
    }
}

class TestMap extends HashMap{
    @Override
    public Set keySet(){
        return null;
    }
}
```

Какой результат выполнения данного кода?

8. Что получим после выполнения кода:

```
int a = 0xFF0;
int b = 0xF0F;
int c = 0x0FF;

c = a = (a ^ b | a & b) ^ c;
System.out.println((c | a) ^ (c & a));
```

9. Например вы не знаете как отсортировать лист чисел и не хотите писать сортировку для листа, то очень удобно произвести конвертирование с листа в массив, вызвать сортировку для массива, и осталось вернуться обратного к листу. самый очевидный способ использовать циклы, но допустим вам сказали обойтись без циклов и ваших подпрограмм.

Существует ли метод для преобразования такого типа: `int[] -> ArrayList<Integer>`. Если да, назовите его.

10. Что будет напечатано после попытки компиляции и выполнения кода:

```
int[] mass = {1, 2};
List<String> list = new ArrayList(10);
list.add("03");
```

```
list.add("04");
System.out.println(mass.length + list.size() + ".");
```

11. Дан код:

```
public interface A {
    String text = "a";
}

public interface B {
    String text = "b";
}

public class Implementor implements A, B {
    public static void main(String[] args) {
        System.out.println(Implementor.text);
    }
}
```

Какое значение примет переменная text в классе Implementor? Что выведется на экран?

12. Дан следующий код. Как вы думаете, что произойдет в результате выполнения данного кода?

```
public class Test {
    class A {
        String str = "ab";

        A() {
            printLength();
        }

        void printLength() {
            System.out.println(str.length());
        }
    }

    class B extends A {
        String str = "abc";

        void printLength() {
            System.out.println(str.length());
        }
    }

    public static void main(String[] args) {
        new Test().new B();
    }
}
```

13. Измените код так, чтобы он вернул hello world на консоль.

Будут ли одинаковые hashCode у d1 и d2?

Можно ли считать такое решение удачным?

17. Какие типы данных в Java являются атомарными для операций чтения/записи с точки зрения многопоточности?

18. Можно ли использовать byte[] в качестве ключа в HashMap?

19. В чём разница при использовании new Srtring("") и простого присваивания.?

20. Почему String в Java является immutable?

21. Что будет выведено в консоли?

```
int i = 5;  
i = i++;  
System.out.println(i);
```

22. Что будет выведено в консоли?

```
i = 5;  
i = i++ + i++;  
System.out.println(i);
```

23. Что будет выведено в консоли?

```
i = 5;  
i = i++ + ++i;  
System.out.println(i);
```

25. Что будет выведено в консоли?

```
i = 5;  
i = ++i + ++i;  
System.out.println(i);
```

26. Сравните 2 фрагмента кода:

```
1: int v1=1; long v2=2; v1=v1+v2;  
2: int v1=1; long v2=2; v1+=v2;
```

Будет ли разница при исполнении этих строк кода?

27. Опишите механизм добавления элемента в HashMap

28. Что такое "**Deadlock**" (взаимная блокировка)? Причины возникновения? Методы предотвращения?

29. Дан класс точка, предложите реализацию метода equals, чтобы обеспечить корректность сравнения, в том числе и математического.

```
1. public class Point {  
    private double x;  
    private double y;
```

```
    public Point(double x, double y) {  
        this.x = x;  
        this.y = y;  
    }  
}
```

Описание показателей и критериев оценивания с указанием шкалы оценивания для очной и других форм обучения (с применением балльно-рейтинговой системы):

Оцениваются следующие показатели: понимание вопросов, правильность, полнота и логическое изложение ответов.

Оценка по дисциплине складывается из текущего рейтинга и зачетного рейтинга.

Рейтинг зачета с оценкой определяется следующим образом:

Ответы на 1 теоретический вопрос – до 10 баллов, на 2 теоретический вопрос – до 10 баллов, выполнение практического задания – до 10 баллов, дополнительные вопросы в рамках курса до 10 баллов.

Оценивание ответов на любой из теоретических вопросов:

9-10 баллов выставляется, если студент демонстрирует полное понимание вопросов, правильность ответов, полное и логически последовательное изложение материала.

7-8 баллов выставляется, если студент демонстрирует: значительное понимание вопросов, правильность, но недостаточную полноту ответов на заданные теоретические вопросы; допущение неточности ответа;

5-6 баллов выставляется, если студент демонстрирует: понимание вопросов, по существу излагает материал, но не усвоил его деталей, есть погрешности в ответах; допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении материала;

Менее 5 баллов выставляется, если студент демонстрирует: непонимание вопросов; студент не знает значительной части материала, не ответил на дополнительные вопросы или отказался от ответов на вопросы и задания.

Оценивание любого из практических заданий:

9-10 баллов выставляется, если студент демонстрирует полное понимание заданий, правильность ответов; полное, точное и логически последовательное изложение материала;

7-8 баллов выставляется, если студент демонстрирует: значительное понимание заданий, правильность, но недостаточную полноту ответов на заданные задания; допущение

неточности ответа;

5-6 баллов выставляется, если студент демонстрирует: понимание заданий, основные этапы задания выполнены, но есть погрешности в ответах

Менее 5 баллов выставляется, если студент демонстрирует: непонимание заданий; основные шаги задания не выполнены или выполнены неправильно, не ответил на дополнительные вопросы или отказался от ответов на вопросы и задания.

Минимальный балл зачетного рейтинга в соответствии с положением о рейтинге равен 24.

В итоге по курсу, суммируя итоги текущего рейтинга и зачетного рейтинга:

- оценка «отлично» выставляется обучающемуся, если он набрал 87-100 баллов;
- оценка «хорошо» выставляется обучающемуся, если он набрал 73-86 баллов;
- оценка «удовлетворительно» выставляется обучающемуся, если он набрал 60-72 балла;
- оценка «неудовлетворительно» выставляется обучающемуся, если он набрал менее 60 баллов;

Перечень тем для курсового проектирования (курсовых работ/проектов)

по дисциплине Б1.В.06 Разработка кроссплатформенных приложений

1. Компьютерная игра "Пять в ряд" (крестики нолики на неограниченном поле).
2. Графический редактор.
3. Текстовый редактор с поддержкой различных кодировок.
4. Компьютерная игра морской бой.
5. Компьютерная игра "Шашки".
6. MP3-проигрыватель.
7. Программа "Видеопроигрыватель".
8. Архиватор.
9. Клиент-серверное приложение "Текстовый чат".
10. Клиент-серверное приложение "Голосовой чат".
11. Компьютерная игра "Змейка".
12. Компьютерная игра "Карточная игра в "Подкидного дурака".
13. Программа для построения графиков функций.
14. Компьютерная игра "Тетрис".
15. Компьютерная игра "Lines".
16. Файловый менеджер.
17. FTP-клиент.
18. Компьютерная игра "Балда".
19. Компьютерная игра "Танчики".
20. Компьютерная игра "Сапер".
21. Компьютерная игра "Угадай мелодию".
22. Почтовый клиент.

Описание показателей и критериев оценивания с указанием шкалы оценивания для очной и других форм обучения (с применением бально-рейтинговой системы):

Оценка за курсовую работу складывается из текущего рейтинга и рейтинга за защиту курсовой работы.

Текущий контроль осуществляется поэтапно.

Текущий рейтинг определяется: правильно подобранным материалом, знанием теоретических основ соответствующих разделов курсовой работы, умением применить их на практике и обосновать используемое решение. На каждом этапе текущего контроля обучающийся может набрать:

18-30 баллов, если студент подобрал и освоил теоретический материал, соответствующий разрабатываемой теме, выполнил необходимые этапы при разработке курсовой работы в соответствии с требованиями задания, владеет теоретическим материалом, связывает его с программной реализацией.

0-18 баллов выставляется, если студент недостаточно проработал этапы курсовой работы в соответствии с требованиями задания, недостаточно правильно и полно владеет теоретическим материалом, не связывает его с программной реализацией.

Рейтинг за защиту курсовой работы определяется следующим образом.

Оцениваются следующие показатели: соответствие требованиям задания, соответствие требованиям оформления отчета, правильность схем алгоритмов, расчетов и работы разработанной программы во время демонстрации, корректность и обоснованность выводов, самостоятельность выполненной работы.

Максимальная оценка при защите курсовой работы – 40 баллов; минимальная – 24 балла. Минимальный балл, свидетельствующий об успешной защите курсовой работы – 24.

32-40 баллов выставляется, если выполнены все требования, предъявляемые к выполнению курсовой работы, выполнены требования по оформлению отчета, приведены правильные схемы алгоритмов, расчетов и разработанная программа работает корректно, предоставлены выводы, работа выполнена самостоятельно, даны полные и правильные ответы на поставленные вопросы.

24-31 балла выставляется, если требования, предъявляемые к выполнению курсовой работы, выполнены с существенными отклонениями и даны неполные и неточные ответы на поставленные вопросы.

0-23 балла выставляется, если не выполнены требования, предъявляемые к выполнению курсовой работы, студент не отвечает на поставленные вопросы.

Оценка за курсовую работу оценивается следующим образом:

- оценка «отлично» выставляется обучающемуся, если он набрал 87-100 баллов;
- оценка «хорошо» выставляется обучающемуся, если он набрал 73-86 баллов;
- оценка «удовлетворительно» выставляется обучающемуся, если он набрал 60-72 балла;
- оценка «неудовлетворительно» выставляется обучающемуся, если он набрал менее 60 баллов;

Вопросы для защиты лабораторных работ и контроль выполнения лабораторных работ

по дисциплине Б1.В.06 Разработка кроссплатформенных приложений

Лабораторная работа 1.

Проверка хода выполнения лабораторной работы.

Вопросы для защиты лабораторной работы:

1. Назовите возможные модификаторы доступа у полей и методов класса?
2. Что означает ключевое слово `final` для класса?
3. Что означает ключевое слово `final` для метода класса?
4. Что означает ключевое слово `static` для метода класса?
5. Можно ли наследоваться от анонимного класса?

Лабораторная работа 2.

Проверка хода выполнения лабораторной работы.

Вопросы для защиты лабораторной работы:

1. Чем массив в Java отличается от класса `ArrayList`?
2. Какие операции быстрее выполняются в `ArrayList`, по сравнению с `LinkedList`?
3. Какие операции быстрее выполняются в `LinkedList`, по сравнению с `ArrayList`?
4. Какова внутренняя структура `HashMap`?
5. Какова внутренняя структура `TreeMap`?

Лабораторная работа 3.

Проверка хода выполнения лабораторной работы.

Вопросы для защиты лабораторной работы:

1. Какими способами может быть сгенерирована исключительная ситуация?
2. Какими способами может быть обработана исключительная ситуация?
3. Каковы особенности работы с исключениями при наследовании?
4. В чем особенность классов, наследуемых от `RuntimeException`?
5. В каком порядке необходимо осуществлять перехват исключений в блоке `try...catch`?

Лабораторная работа 4.

Проверка хода выполнения лабораторной работы.

Вопросы для защиты лабораторной работы:

1. Для чего служит класс `File`?
2. В чем особенность класса `RandomAccessFile` по сравнению с другими классами для работы с файлами?
3. Какие условия должны быть выполнены, чтобы класс мог быть подвергнут сериализации?
4. Перечислите базовые классы для основных иерархий, использующихся при операциях ввода-вывода?

Лабораторная работа 5.

Проверка хода выполнения лабораторной работы.

Вопросы для защиты лабораторной работы:

1. Для чего служит метод yield?
2. Для чего служит метод join?
3. В каком классе определены методы wait, notify, notifyAll?
4. Как работает блок synchronized?
5. Какими способами можно создать нить в Java?

Описание показателей и критериев оценивания с указанием шкалы оценивания для очной и других форм обучения (с применением бально-рейтинговой системы)

Оцениваются следующие показатели: знание теоретических основ лабораторной работы, умение применить их на практике, обосновать используемое решение, выполнение в установленные сроки. В рамках защиты по каждой лабораторной работе задается несколько вопросов.

Для лабораторных работ 1-3:

9-10 баллов выставляется, если студент выполнил работу в установленный срок, правильно и полно отвечает на вопросы, объясняет их на примерах, связывает с программной реализацией.

7-8 баллов выставляется, если студент отвечает на вопросы, объясняет их на примерах, связывает с программной реализацией, но недостаточно полно и с некоторыми неточностями.

5-6 баллов выставляется, если студент отвечает на большинство из заданных вопросов, и может объяснить ход их решения на примере

0-4 баллов выставляется, если студент не отвечает на большую часть заданных вопросов, не может объяснить их на примере.

Для лабораторных работ 4-5:

13-15 баллов выставляется, если студент выполнил работу в установленный срок, правильно и полно отвечает на вопросы, объясняет их на примерах, связывает с программной реализацией.

9-12 баллов выставляется, если студент отвечает на вопросы, объясняет их на примерах, связывает с программной реализацией, но недостаточно полно и с некоторыми неточностями.

5-8 баллов выставляется, если студент отвечает на большинство из заданных вопросов, и может объяснить ход их решения на примере

0-4 баллов выставляется, если студент не отвечает на большую часть заданных вопросов, не может объяснить их на примере.

**Оформление сведений о дополнениях и изменениях, внесенных в ФОС
дисциплины**

Сведения о дополнениях и изменениях, внесенных в ФОС дисциплины

Учебный год	Решение кафедры (№ протокола, дата)	Внесенные в ФОС дополнения и изменения	Подпись заведующего кафедрой