**Вопрос 1**

Дан код программ:

A) *public class Quest21 {*

*public static void main (String [] args) {*

*System.out.println ("Hello, java 7");*

*}}*

B) *public class Quest22 {*

*String java = "Java 7";*

*public static void main (String [] args) {*

*System.out.println (java);*

*}}*

C) *public class Quest23 {*

*{System.out.println ("Java 7"); }*

*}*

Укажите, что скомпилируется без ошибок.

1) AB

2) BC

3) ABC

4) A

5) AC

**Вопрос 2**

Выберите правильные утверждения:

1) класс — это тип данных;

2) объект класса может использоваться всюду, где используется объект подкласса;

3) объект класса можно создать только один раз;

4) на объект класса может не ссылаться объектная переменная.

**Вопрос 3**

Вставьте на место прочерка название одного из принципов ООП так, чтобы получилось верное определение:

1) наследование

2) полиморфизм

3) позднее связывание

4) инкапсуляция

\_\_\_\_\_\_\_\_\_\_ — это объединение данных и методов, предназначенных для манипулирования этими данными в новом типе — классе.

**Вопрос 4**

Дан код:

*String s; // 1*

*if ((s = "java") == "java") {// 2*

*System.out.println (s+ " true");*

*} else {*

*System.out.println (s+ " false");*

*}*

Что будет результатом компиляции и запуска этого кода?

1) ошибка компиляции в строке 1, переменная не проинициализирована

2) ошибка компиляции в строке 2, неправильное выражение для оператора if

3) на консоль выведется java true

4) на консоль выведется java false

**Вопрос 5**

Дан код:

*public class Quest5 {*

*private static void main (String [] args) {*

*System.out.println (args [2]);*

*}*

*}*

Что произойдет, если этот код выполняется следующей командной строкой:

*java Quest5 Java 7""* ?

1) выведется: Java 7

2) ошибка времени выполнения NullPointerException

3) ошибка времени выполнения ArraylndexOutOfBoundsException

4) выведется: Java 7 <пустая строка>

5) выведется: <пустая строка>

6) приложение не запустится

7) код не скомпилируется

**Вопрос 6**

Дан код:

*public class Quest6 {*

*public static void main (String [] args) {*

*System.out.print ("A");*

*main ("java7");*

*}*

*private static void main (String args) {*

*System.out.print ("B");*

*}}*

Что будет выведено в результате запуска и компиляции?

1) ошибка компиляции

2) BA

3) AB

4) AA

5) компиляция пройдет успешно, а при выполнении программа зациклится

**Вопрос 7**

Дан код:

*class Book {*

*private String book;*

*public void setBook (String b) {book = b;}*

*}*

*public class Quest7 {*

*public static void main (String [] args) {*

*Book book1 = new Book (); book1.setBook ("Java 7");*

*Book book2 = new Book (); book2.setBook ("Java 7");*

*if (book1.equals (book2)) {*

*System.out.println ("True");*

*} else {*

*System.out.println ("False");*

*}}}*

Результатом компиляции и запуска кода будет?

1) True

2) ошибка компиляции

3) False

4) код скомпилируется, но при выполнении оператор if будет пропущен.

**Вопрос 8**

Укажите строки, компиляция которых не приведет к ошибке:

1) int var1 = 356f

2) double var2 = 356f

3) float var3 = 356f

4) char var4 = 356f

5) long var5 = 356f

6) byte var6 = 356f

7) Integer var7 = 356f

8) Character var8 = 356f

9) Object var9 = 356f

**Вопрос 9**

Дан код:

*public class Quest4 {*

*public static void main (String [] args) {*

*double x=0, y=2, z;*

*z = y/x;*

*System.out.println ("z="+z);*

*}}*

Что выведется на консоль в результате компиляции и запуска программы?

1) Ошибка компиляции

2) z=Infinity

3) z=NaN

4) Ошибка времени выполнения java.lang.ArithmeticException

**Вопрос 10**

Что будет результатом компиляции и запуска следующего кода?

*public class Quest {*

*public static void main (String [] args) {*

*MedicalStaff medic = new HeadDoctor ();*

*if (medic instanceof Nurse) {*

*System.out.println ("Nurse");*

*} else if (medic instanceof Doctor) {*

*System.out.println ("Doctor");*

*} else if (medic instanceof HeadDoctor) {*

*System.out.println ("HeadDoctor");*

*}}}*

*class MedicalStaff {}*

*class Doctor extends MedicalStaff {}*

*class Nurse extends MedicalStaff {}*

*class HeadDoctor extends Doctor {}*

1) Nurse

2) Doctor

3) HeadDoctor

4) Ошибка компиляции

**Вопрос 11**

Дан фрагмент кода

*if (e1) if (e2) S1; else S2;*

(e1, e2, S1, S2 — корректные java-выражения). Какому другому фрагменту кода он эквивалентен?

1) if (e1) {if (e2) S1; else S2;}

2) if (e1) {if (e2) S1;} else S2;

3) if (e1) if (e2) S1; else; else S2;

4) if (e1) if (e2) S1; else S2; else;

**Вопрос 12**

Какие из фрагментов кода неверно решают задачу «Найти сумму первых 100 натуральных чисел»?

1) i = 1; sum = 0; for (; i <= 100; i++) sum += i;

2) sum = 0; for (i = 1; i <= 100;) sum += i++;

3) for (i = 1, sum = 0; i <= 100; sum += i+, i++);

4) for (i = 1, sum = 0; i <= 100; sum += i++);

5) for (i = 0, sum = 0; i++, i <= 100; sum += i);

**Вопрос 13**

Какие утверждения о классах-оболочках корректны?

1) Классы оболочки Double, Long, Float размещаются в пакете java.util

2) Объекты классов оболочек могут хранить те же значения, что и соответствующие им базовые типы

3) Объекты классов-оболочек хранят изменяемые значения аналогично переменным базовых типов

4) Объекты классов-оболочек по умолчанию получают значение null

5) В классах оболочках определены методы преобразования к базовому типу

**Вопрос 14**

Дан код:

*class Item {}*

1) int [] mas1 = new int [24];

2) Integer mas2 [] = new Integer [24];

3) char [] mas3 = new Character [] {'a', 'b', 'c'};

4) Item [] mas4 = new Item {new Item (), new Item ()};

5) double [] mas5 = {5, 10, 15, 20};

6) int [] mas6 [] = new int [4] [5];

7) int mas7 [] [] = new int [4] [];

Компиляция каких строк приведет к ошибке?

**Вопрос 15**

Какие описания класса содержат синтаксическую ошибку? Код написан в файле Quest1.java

1) public class Quest1 {}

2) public static class Quest1 {}

3) public abstract final class Quest1 {}

4) private class Quest1 {}

5) final class Quest1 {}

**Вопрос 16**

Выберите правильное утверждение, подходящее для окончания фразы «Константное поле может быть проинициализировано ...»:

1) только один раз;

2) один раз при объявлении, а затем в конструкторе класса;

3) в логическом блоке инициализации;

4) в статическом блоке инициализации;

5) при объявлении или в конструкторе класса.

**Вопрос 17**

Дан код:

*public class Quest3 {*

*public static int method () {*

*final int loc;*

*System.out.println (loc);//1*

*loc=4;//2*

*return loc+1;//3*

*}*

*public static void main (String [] args) {*

*method (); method (); method ();*

*System.out.println (method ());*

*}}*

Каким будет результат компиляции и запуска программы?

1) на консоль выведется число 4

2) на консоль выведется число 0

3) ошибка компиляции в строке 1

4) ошибка компиляции в строке 2

5) ошибка компиляции в строке 3

**Вопрос 18**

Выберите утверждения, корректно характеризующие модификаторы доступа:

1) статические private-члены класса доступны только статическим методам этого класса;

2) статические public-члены классы доступны всем методам этого класса;

3) protected-члены класса доступны подклассам другого пакета;

4) поле — член класса, объявленное без модификатора доступа, доступно в классах другого пакета.

**Вопрос 19**

Дан код:

*public class Quest5 {*

*public Quest5 () {}*

*public Quest5 (int i) {this (i, i);}*

*public Quest5 (int i, int j) {this ();}*

*public static void main (String [] args) {*

*Quest5 q = new Quest5 (2,3); //1*

*}}*

Сколько конструкторов вызовется при создании объекта в строке 1?

1) один;

2) два;

3) три;

4) ошибка компиляции.

**Вопрос 20**

Дан код:

*public class Quest6 {*

*public void meth (Number obj) {System.out.print ("1");}*

*public void meth (Character obj) {System.out.print ("2");}*

*private static void meth (Integer obj) {System.out.print ("3");}*

*public void meth (int i) {System.out.print ("4");}*

*public void meth (double d) {System.out.print ("5");}*

*public static void main (String [] args) {*

*Quest6 q = new Quest6 ();*

*Number n = 67;*

*Integer i = 78;*

*q.meth (n);*

*q.meth (i);*

*}}*

Что выведется на консоль после компиляции и запуска этой программы?

1) 14

2) 11

3) 33

4) 44

5) 13

6) ошибка компиляции

7) ошибка выполнения

**Вопрос 21**

Дан код:

*public class Quest7<T> {*

*private T pole;*

*public Quest7 (T pole) {this.pole = pole;} //1*

*public void setPoleDefault () {pole.setTime (1000);} //2*

*public static void main (String [] args) {*

*Quest7<Date> obj = new Quest7<Date> (new Date ());//3*

*obj.setPoleDefault ();*

*}}*

Каким будет результат компиляции и запуска программы?

1) компиляция и запуск пройдут без ошибок

2) ошибка компиляции в строке 1

3) ошибка компиляции в строке 2

4) ошибка компиляции в строке 3

**Вопрос 22**

Укажите корректый способ создания экземпляра класса

*public class Quest8<T1, T2> {}*?

1) Quest8 obj = new Quest8 ()

2) Quest8<Object> obj = new Quest8<Object> ()

3) Quest8<Object, Object> obj = new Quest8<Object, Object> ()

4) Quest8<..., Object> obj = new Quest8<..., Object> ()

5) Quest8<Object, Integer> obj = new Quest8<Integer, Object> ()

6) Quest8<Number, Integer> obj = new Quest8<Integer, Integer> ()

**Вопрос 23**

Дан код:

*enum Numbers {ONE, TWO, THREE, FOUR, FIVE}*

*public class Quest9 {*

*public static void main (String [] args) {*

*Numbers n1 = Numbers.ONE;*

*Numbers n2 = Numbers.ONE;//1*

*if (n1 == n2) {System.out.print ("true");}*

*else {System.out.print ("false");}*

*System.out.println (Numbers.FIVE.ordinal ());//2*

*}}*

Что выведется на консоль в результате компиляции и запуска приложения?

1) false4

2) true4

3) false5

4) true5

5) произойдет ошибка компиляции в строке 1

6) произойдет ошибка компиляции в строке 2

**Вопрос 24**

Выберите неправильные утверждения:

1) перечисление является классом

2) при объявлении перечисления его необходимо явно наследовать от класса java.lang.Enum

3) в теле перечисления можно объявлять только методы

4) конструктор перечисления может быть объявлен со спецификатором public

**Вопрос 25**

Дан класс:

*package ch04.q01;*

*class Quest41 {}*

Укажите правильные варианты наследования от этого класса:

1) package ch04.q01; class Quest4 extends Quest41 {}

2) package ch04.q01.\_2; public class Quest42 extends Quest41 {}

3) package ch04.q01; public class Quest43 implements Quest41 {}

4) package ch04.q01.\_2; import ch04.q01.Quest41;

5) public class Quest44 extends Quest41 {}

6) package ch04.q01; public class Quest45 extends Quest41 {}

**Вопрос 26**

Выберите правильные утверждения:

1) Класс может быть использован в качестве суперкласса для себя самого.

2) В конструкторе класса можно совместно использовать вызовы this и super.

3) Статические методы можно определять в подклассах с той же сигнатурой, что и в базовом классе.

4) Статические методы можно перегружать в подклассах.

5) Динамическое связывание определяет версию вызываемого метода на этапе выполнения.

**Вопрос 27**

Дан код:

*package ch04.q03;*

*public class Quest43 {*

*private final void method () {} //1*

*}*

*class Quest431 extends Quest43 {*

*public void method () {} //2*

*}*

Что произойдет в результате компиляции этого кода?

1) ошибка компиляции в строке 1

2) ошибка компиляции в строке 2

3) компиляция без ошибок

**Вопрос 28**

Дан код двух классов:

*// класс 1*

*package ch04.q04;*

*public class Quest41 {}*

*// класс 2*

*package ch04.q04.\_2;*

*import ch04.q04.Quest41;*

*public class Quest43 extends Quest41 {*

*public Quest43 () {*

*super ();*

*}}*

С каким атрибутом доступа объявлен конструктор по умолчанию в базовом классе Quest41?

1) public

2) private

3) protected

4) friendly

**Вопрос 29**

Дан код:

*package ch04.q05;*

*public class Quest51 {*

*public String toString () {*

*return getClass ().getSimpleName ();*

*}*

*public static void main (String [] args) {*

*Quest53 q = new Quest53 ();*

*System.out.println (q.toString ());*

*}}*

*class Quest52 extends Quest51 {}*

*class Quest53 extends Quest52 {}*

Что выведется на консоль в результате компиляции и запуска программы?

1) Quest52

2) Quest53

3) Quest51

4) ошибка компиляции

**Вопрос 30**

Дан код:

*package ch04.q06;*

*class Item {*

*public int item;*

*Item (int item) {*

*this.item = item;*

*}}*

*public class Quest61 {*

*public static void main (String [] args) {*

*Item ar1 [] = {new Item (1), new Item (2), new Item (3)};*

*Item ar2 [] = ar1.clone ();*

*ar2 [0].item = 4;*

*System.out.println (ar1 [0].item +" " + ar1 [1].item + " " + ar1 [2].item);*

*}}*

Что выведется на консоль после компиляции и запуска этой программы?

1) 1 2 3

2) 1 4 3

3) 4 2 3

4) ошибка компиляции

5) ошибка выполнения

**Вопрос 31**

Дан код:

*public class Quest1 {*

*public static void main (String [] args) {*

*for (Numbers num: Numbers.values ()) {*

*System.out.print (num.getNumber ());*

*}}}*

*enum Numbers {*

*ONE (1), TWO (2) {public int getNumber () {return x + x;}*

*}, THREE (3) {public int getNumber () {return x + x + x;}*

*}, FOUR (4), FIVE (5);*

*int x;*

*Numbers (int x) {*

*this.x = x;*

*}*

*public int getNumber () {*

*return x;*

*}}*

Что будет результатом компиляции и выполнения данного кода?

1) строка 12345

2) строка 54321

3) строка 14945

4) строка 54941

5) строка 12945

6) строка 54921

7) строка 14345

8) строка 54341

9) ошибка компиляции

**Вопрос 32**

Дан код:

*public class Quest3 {*

*public static void main (String [] args) {*

*Outer obj = new Outer ().new Inner1 ();*

*obj.print ();*

*}}*

*class Outer {*

*public void print () {}*

*class Inner1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ // line 1*

*{*

*public void print () {*

*System.out.println ("In inner.");*

*}}}*

Что необходимо дописать в line 1, чтобы при компиляции и запуске в консоль вывелась строка In inner ?

1) ничего, код написан верно

2) implements Outer

3) extends Outer

4) нет верного варианта

**Вопрос 33**

Выберите правильный вариант доступа из внутреннего Inner класса к экземпляру его внешнего Outer класса:

1) Outer.class.this;

2) new Outer ().this;

3) Inner.Outer.class.this;

4) Outer.class.newInstance ().this;

5) Outer.this.

**Вопрос 34**

Даны два фрагмента кода:

Фрагмент 1

*new Object () {*

*public void hello () {*

*System.out.print ("Hello!");*

*}*

*}.hello ();*

Фрагмент 2

*Object obj = new Object () {*

*public void hello () {*

*System.out.print ("Hello!");*

*}};*

*obj.hello ();*

Каким будет результат компиляции и запуска этих фрагментов кода?

1) и первый, и второй фрагменты кода скомпилируются и выведут на консоль строку «Hello!»;

2) первый фрагмент кода скомпилируется и выведет на консоль строку «Hello!», при компиляции второго фрагмента возникнет ошибка;

3) второй фрагмент кода скомпилируется и выведет на консоль строку «Hello!», при компиляции первого фрагмента возникнет ошибка;

4) ни первый, ни второй фрагмент кода не скомпилируются.

**Вопрос 35**

Выберите неправильные утверждения:

1) методы внутреннего (нестатического) класса имеют прямой доступ только к статическим полям и методам внешнего класса;

2) доступ к элементам внутреннего класса возможен только из внешнего класса через объект внутреннего класса;

3) внутренние классы могут содержать static-поля;

4) внутренние классы не могут реализовывать интерфейсы;

5) внутренние классы могут быть объявлены только с параметрами final, abstract, public, protected, private;

6) внутренний локальный класс обладает доступом только к полям метода, в котором он объявлен;

7) внутренние классы могут содержать static-поля, только если они final static;

8) внутренние классы могут быть производными классами, но не базовыми.

**Вопрос 36**

Что будет результатом компиляции и запуска следующего кода?

*interface Quest10{ Number returner(); }*

*abstract class Quest100{*

*public abstract Integer returner();*

*}*

*public class Quest1 extends Quest100 implements Quest10{//line 1*

*@Override //line 2*

*public Integer returner() {// line 3*

*return new Integer(6);*

*}*

*public static void main(String[] args) {*

*Quest1 quest = new Quest1();*

*Quest10 quest10 = quest;*

*Quest100 quest100 = quest;*

*System.out.println(quest10.returner()+""+quest100.returner()); }}*

1) ошибка компиляции в строке 1

2) ошибка компиляции в строке 2

3) ошибка компиляции в строке 3

4) компиляция и запуск программы осуществится без ошибок

**Вопрос 37**

Даны объявления интерфейсов. Которые из них скомпилируются с ошибкой?

1) *interface Quest20{*

*class Inner{*

*private int x;*

*}}*

2) *interface Quest21{*

*static class Inner{ int x; }*

*}*

3) *class Quest22{*

*interface Inner{ int x; }*

*}*

*class Quest23{*

*static interface Inner{*

*int x;*

*}}*

**Вопрос 38**

Дан код:

*interface Inter1{*

*void f();*

*}*

*interface Inter2 extends Inter1{*

*void f();*

*}*

*class Cl1 implements Inter1{*

*public void f(){*

*System.out.println("one");*

*}}*

*class Cl2 implements Inter2{*

*public void f(){*

*System.out.println("two");*

*}}*

Какой метод f() будет вызван при выполнении следующего кода (1)?

*Inter2 obj = new Cl2();*

*((Inter1) obj).f();*

1) метод f() класса Cl1

2) метод f() класса Cl2

3) произойдет ошибка компиляции, т. к. ссылка типа Inter2 не может ссылаться на объект Cl2

4) произойдет ошибка компиляции, т. к. ссылку obj1 нельзя привести к типу Inter1

**Вопрос 39**

Укажите, какое ключевое слово используется для объявления аннотации:

1) @interface;

2) @class;

3) @annotation;

4) @custom\_annotation.

**Практические задания:**

0. Если возможно, создать подкласс для класса. Ответ обосновать.

*class Policy {*

*private int a;*

*Policy(int a){*

*this.a=a;}}*

1. В строки, помеченные комментариями //1 и //2, записать оптимальные вызовы метода *execute()*

*public class Hope {*

*public static void execute(){}*

*private void taste(){//1*

*}}*

*public class Dream {*

*private Float taste(){//2*

*}}*

2. Записать реализацию интерфейса

*interface Application {*

*void hello();}*

3. Если возможно, создать подкласс абстрактного класса. Ответ обосновать

*abstract class Cook {*

*protected abstract void sell();}*

4. Реализовать метод hashCode для класса

*class Quest {*

*private int a;*

*private short b;}*

5. Реализовать метод equals для класса

*class Bus {*

*protected String type;}*

6. Дана строка вида String num = "1234"; преобразовать к int n четырьмя способами.

**Вопрос 40**

Дан код:

*String s1 = "Minsk";*

*String s2 = new String("Minsk");*

*if(s1.equals(s2.intern())){*

*System.out.print("true");*

*} else {*

*System.out.print("false");*

*}*

*if(s1 == s2){*

*System.out.print("true");*

*} else{*

*System.out.print("false");*

*}*

В результате при компиляции и запуске будет выведено:

1) truefalse

2) falsetrue

3) truetrue

4) falsefalse

5) ошибка компиляции: заданы некорректные параметры для метода equals()

**Вопрос 41**

Дан код:

*StringBuilder sb1 = new StringBuilder("I like Java.");//1*

*StringBuilder sb2 = new StringBuilder(sb1);//2*

*if (sb1.equals(sb2)){*

*System.out.println("true");*

*} else {*

*System.out.println("false");*

*}*

В результате при компиляции и запуске будет выведено (1):

1) true

2) false

3) ошибка компиляции в строке 1

4) ошибка компиляции в строке 2

**Вопрос 42**

Что будет результатом компиляции и выполнения следующего кода?

*Pattern p = Pattern.compile("(1\*)0");*

*Matcher m = p.matcher("111110");*

*System.out.println(m.group(1));*

1) вывод на консоль строки «111110»;

2) вывод на консоль строки «11111»;

3) ошибка компиляции;

4) ошибка выполнения.

**Вопрос 43**

Что выведется на консоль при компиляции и выполнении следующих строчек кода?

*Locale loc = new Locale("ru", "RU");*

*System.out.println(loc.getDisplayCountry(Locale.US));*

1) United States;

2) Соединенные Штаты;

3) Россия;

4) Russia;

5) ошибка компиляции.

**Вопрос 44**

Укажите, какому пакету принадлежат классы, позволяющие форматировать числа и даты: NumberFormat и DateFormat.

1) java.text

2) java.util.text

3) java.util

4) java.lang

**Вопрос 45**

Выберите правильные утверждения:

1) Проверяемые (checked) исключения являются наследниками класса java.lang.Exception

2) Непроверяемые (unchecked) исключения являются наследниками класса java.lang.Error

3) Непроверяемые (unchecked) исключения являются наследниками класса java.lang.Exception

4) Проверяемые (checked) исключения обязательно обрабатываются

5) Непроверяемые (unchecked) исключения невозможно обработать

**Вопрос 46**

Дан код:

*try { FileReader fr1 = new FileReader("test1.txt");*

*try { FileReader fr2 = new FileReader("test2.txt");*

*} catch (IOException e) {*

*System.out.print("test2");*

*}*

*System.out.print("+");*

*} catch (FileNotFoundException e) {*

*System.out.print("test1");*

*}*

*System.out.print("+");*

Какая строка выведется на консоль при компиляции и запуске этого кода, если файл test1.txt существует и доступен, а test2.txt нет?

1) test1

2) test1+

3) test1++

4) test2

5) test2+

6) test2++

7) ошибка компиляции

**Вопрос 47**

Дана иерархия исключений:

*class A extends java.lang.Exception{}*

*class B extends A{}*

*class C extends B{}*

*class D extends A{}*

*class E extends A{}*

*class F extends D{}*

*class G extends D{}*

*class H extends E{}*

Выберите цепочки блоков catch, использование которых не приведет к ошибке компиляции, если в соответствующем блоке try могут генерироваться исключения типа C,D,G,H:

1) catch(C e){}catch(D e){}catch(H e){}catch(A e){}

2) catch(C e){}catch(D e){}catch(E e){}catch(A e){}

3) catch(C e){}catch(D e){}catch(G e){}catch(A e){}

4) catch(A e){}catch(D e){}catch(G e){}catch(H e){}

5) catch(E e){}catch(D e){}catch(B e){}catch(A e){}

**Вопрос 48**

Дан код:

*class A{*

*public void f() throws IOException{}*

*}*

*class B extends A{}*

Каким образом можно переопределить метод f() в классе B, не вызвав при этом ошибку компиляции?

1) public void f() throws Exception {}

2) public void f() throws IOException {}

3) public void f() throws InterruptedException, IOException {}

4) public void f() throws IOException, FileNotFoundException {}

5) public void f() throws FileNotFoundException {}

6) public void f() throws FileNotFoundException, InternalError {}

**Вопрос 49**

Дан код:

*public class Quest {*

*private int qQ;*

*public Quest(int q) {*

*qQ = 12 / q;//1*

*}*

*public int getQQ() {*

*return qQ;//2*

*}*

*public static void main(String[] args) {*

*Quest quest = null;*

*try {*

*quest = new Quest(0);//3*

*} catch (Exception e) {//4*

*}*

*System.out.println(quest.getQQ());//5*

*}}*

Укажите строку, выполнение которой приведет к необрабатываемой в данном коде исключительной ситуации :

1) 1

2) 2

3) 3

4) 4

5) 5

**Вопрос 50**

Укажите классы, которые стоят во главе символьной иерархии потоков ввода/вывода в Java (2):

1) InputStreamReader

2) OutputStreamWriter

3) Reader

4) Writer

5) InputStream

6) OutputStream

**Вопрос 51**

Дан код:

*public class Quest {*

*public static void main(String[] args) throws IOException {*

*Scanner sc1 = new Scanner(System.in);*

*int x1 = 0;*

*x1 = sc1.nextInt();*

*sc1.close();*

*int x2=0;*

*x2 = System.in.read();*

*System.out.println(x1+" "+(char)x2);*

*}}*

Что будет результатом компиляции и запуска этого кода, если предполагается при первом считывании с консоли ввести 1, а при втором — 2?

1) компиляция и запуск пройдут успешно, на консоль выведется строка «1 2»;

2) компиляция и запуск пройдут успешно, на консоль выведется строка «1 0»;

3) компиляция и запуск пройдут успешно, на консоль выведется строка «1»;

4) компиляция кода осуществится успешно, а при работе программы возникнет исключительная ситуация;

5) ошибка компиляции.

**Вопрос 52**

Дан код:

*class A {*

*public int a=0;*

*public A() {*

*a = 1;*

*}}*

*class B extends A implements Serializable {*

*public int b=0;*

*public B() {*

*a = 2;*

*b = 3;*

*}}*

Какие значения примут поля объекта, созданного от класса B с помощью конструктора без параметров, на который ссылается ссылка b при десериализации?

1) b.a=1, b.b=3

2) b.a=2, b.b=3

3) b.a=0, b.b=0

4) b.a=0, b.b=3

5) b.a=1, b.b=0

6) десериализация невозможна, так как класс A не реализует интерфейс Serializable

**Вопрос 53**

Укажите классы байтовых потоков ввода-вывода, для которых не существует аналогичных классов в символьной иерархии потоков ввода-вывода:

1) DataInputStream

2) InputStreamReader

3) ObjectOutputStream

4) StringWriter

**Вопрос 54**

Компиляция каких строк следующего кода приведет к ошибке (3)?

*public class Quest {*

*public static void main(String[] args) throws IOException {*

*File file = new File("files.t1.txt");*

*FileWriter obj1 = new FileWriter(file); //1*

*ByteArrayInputStream obj2 = new ByteArrayInputStream(file); //2*

*InputStreamReader obj3 = new InputStreamReader(file); //3*

*BufferedReader obj4 = new BufferedReader(file); //4*

*PrintWriter obj5 = new PrintWriter(file); //5*

*}}*

1) ошибка компиляции в строке 1;

2) ошибка компиляции в строке 2;

3) ошибка компиляции в строке 3;

4) ошибка компиляции в строке 4;

5) ошибка компиляции в строке 5.

**Вопрос 55**

Какой класс коллекции позволяет наращивать и сокращать размер, предостав-

ляет индексный доступ к элементам, но его методы несинхронизированы (1)?

1) java.util.HashSet

2) java.util.ArrayList

3) java.util.LinkedHashSet

**Вопрос 56**

Дан класс:

*class X{*

*private int x;*

*public X(int x){this.x = x;}*

*public int hashCode(){*

*return 2;*

*}}*

Каков будет размер коллекции set после выполнения следующего кода?

*X obj1 = new X(1);*

*X obj2 = new X(1);*

*Set<X> set = new HashSet<X>();*

*set.add(obj1);*

*set.add(obj2);*

1) 0;

2) 2;

3) 1;

4) при выполнении данного кода возникнет исключительная ситуация.

**Вопрос 57**

Даны два фрагмента кода:

1. *Vector<String> vct = new Vector<String>();*

*vct.add("One"); vct.add("Two");*

*vct.add("Three"); vct.add("Four");*

*vct.add("Five"); vct.add("Six");*

*Iterator itr = vct.iterator();*

*vct.remove(0);*

*System.out.println(itr.next());*

2. *Vector<String> vct = new Vector<String>();*

*vct.add("One"); vct.add("Two");*

*vct.add("Three"); vct.add("Four");*

*vct.add("Five"); vct.add("Six");*

*Enumeration enm = vct.elements();*

*vct.remove(0);*

*System.out.println(enm.nextElement());*

Укажите разницу при выполнении первого и второго фрагмента кода:

1) при выполнении первого фрагмента на консоль выведется строка «Two», а при выполнении второго произойдет исключительная ситуация;

2) при выполнении второго фрагмента на консоль выведется строка «Two», а при выполнении первого произойдет исключительная ситуация;

3) выполнение двух фрагментов кода приведет к исключительной ситуации;

4) при выполнении двух фрагментов кода на консоль выведется строка «Two»;

5) при компиляции первого фрагмента приведет к ошибке;

6) компиляция второго фрагмента кода приведет к ошибке;

7) компиляция двух фрагментов кода приведет к ошибке.

**Вопрос 58**

Дан код:

*import java.util.\*;*

*enum PCounter {UNO, DOS, TRES, CUATRO, CINCO, SEIS, SIETE};*

*public class Quest {*

*public static void main(String[] args) {*

*EnumSet<PCounter> enst1 = EnumSet.range(PCounter.TRES, PCounter.CINCO);//1*

*EnumSet<PCounter> enst2 = EnumSet.complementOf(enst1);//2*

*System.out.println(enst2);*

*}}*

Что будет выведено при попытке компиляции и запуска программы?

1) [UNO, DOS, TRES, CINCO, SEIS, SIETE]

2) [DOS, TRES, CUATRO, CINCO, SEIS]

3) [UNO, DOS, SEIS, SIETE]

4) ошибка компиляции в строке 1

5) ошибка компиляции в строке 2

6) ничего из перечисленного

**Вопрос 59**

Что будет результатом компиляции и запуска следующей программы?

*import java.util.\*;*

*public class Quest {*

*public static void main(String[] args) {*

*NavigableMap<String, Number> nmap = new TreeMap<String, Number>();*

*nmap.put("one", new Integer(1)); nmap.put("two", new Integer(2));*

*nmap.put("three", new Integer(3)); nmap.put("four", new Integer(4));*

*Map<String, Number> map = nmap.headMap("three");*

*System.out.println(map);*

*}}*

1) {four=4, one=1}

2) {one=2, two=2}

3) {three=3, four=4}

4) {four=4, one=1, three=3}

5) {one=2, two=2, three=3}

**Вопрос 60**

Что будет результатом компиляции и запуска следующей программы?

*public void testInt() {*

*Integer i = null;*

*if(i == 0) {*

*System.out.println(true);*

*} else {*

*System.out.println(false);*

*}*

*}*

1) false

2) Ошибка компиляции

3) Ошибка выполнения

4) true