|  |
| --- |
| , RD Dep. |
| Тест  JAVA.SE.06 Generic and Collection |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| <1.0> | Первая версия | Ольга Смолякова, Игорь Блинов | <13.12.2013> |  |  |
|  |  |  |  |  |  |

**Вопрос 1:**

Какой из интерфейсов является корневым для иерархии коллекций?

Варианты ответа:

* Collections
* Collection
* Enumeration
* Map
* List

**Вопрос 2:**

Какие из следующих утверждений верны?

Варианты ответа:

* Класс Collections реализует интерфейс Collection.
* SDK не предоставляет прямую реализацию интерфейса Collection
* Класс AbstractCollection реализует интерфейс Collection
* Интерфейс Map расширяет интерфейс Collection
* Интерфейс List расширяет интерфейс Collection

**Вопрос 3:**

Какие конструкторы должны предоставлять классы, реализующие интерфейс Collection?

Варианты ответа:

* Конструктор без аргументов, который создает пустую коллекцию.
* Конструктор с параметрами типов int и float, который создает коллекцию заданной емкости и с заданным коэффициентом загруженности.
* Конструктор с параметром типа int, который создает коллекцию заданной емкости.
* Конструктор с аргументом типа Collection, который создает новую коллекцию с теми же аргументами, что и в аргументе.
* Ничего из вышеперечисленного.

**Вопрос 4:**

Какие из следующих классов наследуются от класса AbstractCollection?

Варианты ответа:

* Collections
* HashSet
* TreeSet
* AbstractList
* AbstractMap

**Вопрос 5:**

Какие из следующих классов наследуются от класса AbstractCollection (прямо или косвенно)?

Варианты ответа:

* Collections
* ArrayList
* BitSet
* AbstractList
* AbstractMap

**Вопрос 6:**

Какие из следующих классов не являются классами коллекций?

Варианты ответа:

* LinkedHashSet
* LinkedHashList
* IdentityHashMap
* Dictionary
* BitSet

**Вопрос 7:**

Какой класс состоит исключительно из статических методов и предлагает базовые алгоритмы для работы с коллекциями?

Варианты ответа:

* Collection
* Map
* Collections
* Vector
* Enumerations

**Вопрос 8:**

Какие из следующих утверждений насчет класса java.util.HashSet корректны?

Варианты ответа:

* Он хранит элементы отсортированном порядке.
* Может содержать дубликаты.
* Хранит пары имя-значение.
* Элементы всегда возвращаются в том же порядке.
* Может хранить null.

**Вопрос 9:**

Какие из следующих утверждений корректны?

Варианты ответа:

* List хранит элементы в отсортированном порядке.
* Set хранит элементы отсортированными и List хранит элементы упорядоченными.
* SortedSet хранит элементы упорядоченными.
* OrderedSet хранит элементы отсортированными.
* OrderedList хранит элементы упорядоченными.

**Вопрос 10:**

Какие классы коллекций возвращают элементы в одном и том же порядке?

Варианты ответа:

* HashMap.
* LinkedHashSet.
* HashTable.
* LinkedHashMap.
* TreeMap.

**Вопрос 11:**

Какой класс является базовым для класса Arrays?

Варианты ответа:

* Collections.
* AbstractCollection.
* Object.
* ArrayList.
* AbstractArray.

**Вопрос 12:**

Выберите верные утверждения для двух объектов коллекций c1 и c2?

Варианты ответа:

* c1.retainAll(c2) не изменит c1.
* c1.retainAll(c2) не изменит c2.
* c1.addAll(c2) вернет новую коллекцию, содержащую элементы из c1 и c2.
* для: c2.retainAll(c1); c1.containsAll(c2); 2-я инструкция возвратит true.
* для: c2.addAll(c1); c1.retainAll(c2); 2-я инструкция не окажет никакого воздействия на c1.

**Вопрос 13:**

Какие классы реализуют интерфейс Map?

Варианты ответа:

* WeakMap.
* AbstractMap.
* RenderingHints.
* Dictionary.
* Attributes.

**Вопрос 14:**

Какие методы являются общими для интерфейсов Collection и Map?

Варианты ответа:

* Object remove(Object key)
* Collection values()
* Object add(Object o)
* void clear()
* int hashCode( )

**Вопрос 15:**

Выберите отличия интерфейса Comparator от интерфейса Comparable?

Варианты ответа:

* Comparator содержит два метода, Comparable – один;
* Comparable обеспечивает натуральное упорядочивание
* Интерфейс Comparator используется в классах TreeMap и TreeSet
* Интерфейс Comparable используется в классе SortedMap;
* Интерфейс Comparator расширяет интерфейс java.io.Serializable;

**Вопрос 16:**

Выберите устаревшие классы коллекций:

Варианты ответа:

* Dictionary;
* BitSet;
* Vector;
* HashMap;
* HashTable;

**Вопрос 17:**

Какие классы имеют синхронизированные методы?

Варианты ответа:

* ArrayList;
* Vector;
* LinkedList;
* HashMap;
* HashTable;

**Вопрос 18:**

Выберите верные утверждения насчет HashSet и LinkedHashSet:

Варианты ответа:

* LinkedHashSet наследуется от HashSet;
* HashSet и LinkedHashSet поддерживают упорядочивание элементов;
* LinkedHashSet имеет тот же набор конструкторов, что и HashSet;
* LinkedHashSet хранит элементы в отсортированном порядке;
* LinkedHashSet не может содержать null элемент;

**Вопрос 19:**

В каком из следующих классов коллекций объект будет удален автоматически из коллекции после того, как на ключ больше нет ссылки в программе?

Варианты ответа:

* HashMap;
* LinkedHashMap;
* IdentityHashMap;
* WeakHashMap;
* Hashtable;

**Вопрос 20:**

В каком пакете содержатся коллекции?

Варианты ответа:

* java.lang
* java.collections
* java.util
* java.utils

**Вопрос 21:**

Если необходимо создать объект со структурой, представляющей разновидность списка, в которой большинство операций выполняется над элементами расположенными в конце объекта, то следует использовать:

Варианты ответа:

* Set
* ArrayList
* LinkedList
* List

**Вопрос 22:**

Что из перечисленного являются устаревшими типами коллекций?

Варианты ответа:

* Map
* Dictionary
* HashSet
* Vector

**Вопрос 23:**

Что будет выведено в результате компиляции и запуска следующего кода?

Map m1 = new HashMap();

m1.put("k1","s");

Map m2 = new HashMap(m1);

m1.put("k2", m2);

System.out.print(m1);

Варианты ответа:

* {k1=s}
* {k2=s}
* {k1=s, k2={k1=s}};
* {k1={k2=s}};
* StackOverflowError;

**Вопрос 24:**

Что будет выведено в результате компиляции и запуска следующего кода?

Map m1 = new HashMap();

m1.put("k1","s");

Map m2 = new HashMap(m1);

m1.put("k2", m2);

Map m3 = new HashMap(m1);

m2.put("k2",m3);

m1.put("k3",m2);

System.out.print(m1);

Варианты ответа:

* {k1=s, k2={k1=s},k3={k1=s, k2={k1=s}}}
* {k1=s, k2={k1=s},k3={this Map}}}
* {k1=s, k2={k1=s}};
* {k1={k2=s}};
* StackOverflowError

**Вопрос 25:**

Что будет выведено в результате компиляции и запуска следующего кода?

Set s = new HashSet();

for (int i = 0; i < 5; i++)

s.add(""+i);

Map m = new HashMap(s);

System.out.print(m);

Варианты ответа:

* будет выведено {0,1,2,3,4}
* будет выведено {0=0,1=1,2=2,3=3,4=4}
* будет выведено {null=[0,1,2,3,4]}
* будет выведено {null=[Set]}
* никакой из перечисленных вариантов ответов не подходит;

**Вопрос 26:**

Что будет выведено в результате компиляции и запуска следующего кода?

Map m = new IdentityHashMap();

Object o;

m.put(o,"s");

m.put(null,"t");

System.out.println(m);

Варианты ответа:

* {null=s}
* {null=t}
* {null=s, null=t}
* {o=s, null=t}
* никакой из перечисленных вариантов ответов не подходит

**Вопрос 27:**

В какой строке будет ошибка компиляции?

AbstractSequentialList l = new LinkedList(); //1

l.add(2,"s"); //2

Collection c = l; //3

Iterator i = c.iterator();

ListIterator li = (ListIterator)i; //4

while(li.hasNext())

System.out.println(li.next());

Варианты ответа:

* //1
* //2
* //3 AbstractSequentialList не реализует интерфейс Collection
* //4 ошибка приведения типов во время выполнения
* все строки правильные

**Вопрос 28:**

Что будет выведено в результате компиляции и запуска следующего кода?

(двоичное представление числа 111 есть 1101111)

Set s = new BitSet(111);

System.out.println(s);

Варианты ответа:

* [1,1,1]
* [111]
* [1,1,0,1,1,1,1]
* [1101111]
* никакой из перечисленных вариантов ответов не подходит

**Вопрос 29:**

Что будет выведено в результате компиляции и запуска следующего кода?

Set st,sh;

sh = new HashSet();

sh.add("1");sh.add("2");sh.add("3");

st = new TreeSet(sh);

System.out.println(st);

System.out.println(sh);

Варианты ответа:

* [1, 2, 3][3, 2, 1]
* [1, 2, 3][1, 2, 3]
* [3, 2, 1][3, 2, 1]
* [1, 2, 3], а порядок вывода элементов sh не определен
* никакой из перечисленных вариантов ответов не подходит

**Вопрос 30:**

Что будет выведено в результате компиляции и запуска следующего кода?

HashSet sh = new HashSet();

sh.add("1");sh.add("2");sh.add("3");

TreeSet st = sh;

System.out.print(st);

System.out.print(sh);

Варианты ответа:

* [1, 2, 3][3, 2, 1]
* [1, 2, 3][1, 2, 3]
* [3, 2, 1][3, 2, 1]
* [1, 2, 3], а порядок вывода элементов sh не определен
* никакой из перечисленных вариантов ответов не подходит

**Вопрос 31:**

Дан следующий код:

Set set = new LinkedHashSet();

set.add("1");set.add("2");set.add("3");

Set synchronizedSet = Collections.synchronizedSet(set);

set.remove("1");

System.out.println(synchronizedSet);

Выберите верные утверждения:

Варианты ответа:

* будет выведено [1, 2, 3]
* будет выведено [2, 3]
* к synchronizedSet можно обращаться только из синхронизированного кода;
* к synchronizedSet можно обращаться только когда set == null;
* никакой из перечисленных вариантов ответов не подходит;

**Вопрос 32:**

Что будет выведено в результате компиляции и запуска следующего кода?

Set set = new LinkedHashSet();

set.add('1');set.add('2');set.add('3');

System.out.println(set.hashCode());

Варианты ответа:

* будет выведено 6
* будет выведено 150
* будет выведено что-то типа 3242647
* будет выведено 0
* никакой из перечисленных вариантов ответов не подходит

**Вопрос 33:**

В какой строке произойдет ошибка типа StackOverflowError?

Set set = new LinkedHashSet();

set.add("1");set.add("2");set.add("3");

set.add(set);

System.out.println(set); //1

System.out.println(set.hashCode()); //2

Варианты ответа:

* //1
* //2
* вообще не произойдет
* никакой из перечисленных вариантов ответов не подходит

**Вопрос 34:**

Что будет выведено в результате компиляции и запуска следующего кода?

import java.util.\*;

public class Q1 {

public static void main(String[] args) {

final Collection c = new HashSet();

c.add("s1"); c.add("s2"); c.add("s3");

Iterator i = c.iterator();

i.remove();

System.out.println(c);

}

}

Варианты ответа:

* [s2, s1, s3];
* [s2, s3] ;
* [s1, s2, s3];
* будет удален произвольный элемент;
* IllegalStateException;

**Вопрос 35:**

Выберите строки в которых допущена ошибка?

Collection c1 = new Collection(); //1

Collection c2 = new HashSet(); //2

Collection c3 = new SortedSet(); //3

Collection c4 = new HashMap(); //4

Collection c5 = new AbstractCollection(); //5

Варианты ответа:

* //1,//3;
* //4,//5;
* во всех
* все строки правильные
* //2

**Вопрос 36:**

Что будет выведено в результате компиляции и запуска следующего кода?

Map m = new HashMap();

m.put(null,"str1");

m.put(m,"str2");

System.out.println(m);

Варианты ответа:

* {null=str1, (this Map)=str2}
* {null=str1, {null=str1, {null=str1, …}}};
* StackOverflowError;
* ничего из вышеперечисленного;
* вывод не определен;

**Вопрос 37:**

Что будет выведено в результате компиляции и запуска следующего кода?

Map m = new HashMap();

m.put(null,"str1"); //1

m.put(null,"str2"); //2

m.put(null,"str3"); //3

m.put(null,null); //4

System.out.println(m);

Варианты ответа:

* будет выведено {null=str1, null=str2, null=str3};
* будет выведено {null=str1, null=str2, null=str3, null=null};
* будет выведено {null=null};
* ошибки в строках //1,//2,//3
* ошибка в строке //4;

**Вопрос 38:**

Что будет выведено в результате компиляции и запуска следующего кода?

Collection c1 = new LinkedHashSet();

c1.add("s1"); c1.add("s2"); c1.add("s3"); c1.add("s4");

Collection c2 = new HashSet(c1);

System.out.println(c1);

System.out.println(c2);

Варианты ответа:

* будет выведено [s1, s2, s3, s4] и [s1, s2, s3, s4];
* будет выведено [s1, s2, s3, s4] и [s2, s4, s1, s3];
* будет выведено [s1, s2, s3, s4], а порядок вывода элементов c2 не определен
* порядок вывода элементов c1 и c2 не определен;
* будет выведено [s2, s4, s1, s3] и [s2, s4, s1, s3];

**Вопрос 39:**

Что будет выведено в результате компиляции и запуска следующего кода?

import java.util.\*;

public class Test {

public static void main(String[] a) {

Map m = new HashMap(null);

System.out.print(m.size());

}

}

Варианты ответа:

* compile time error
* runtime error
* 0
* [null]

**Вопрос 40:**

Что будет выведено в результате компиляции и запуска следующего кода?

import java.util.\*;

public class Test {

public static void main(String[] a) {

SortedSet ss = new SortedSet();

ss.add(null);

System.out.print(ss.size());

}

}

Варианты ответа:

* compile time error
* runtime error
* 0
* [null]

**Вопрос 41:**

Что будет выведено в результате компиляции и запуска следующего кода?

import java.util.\*;

public class Test {

public static void main(String[] a) {

SortedSet ss = new Set();

ss.add(null);

System.out.print(ss.size());

}

}

Варианты ответа:

* compile time error
* runtime error
* 0
* [null]

**Вопрос 42:**

Что будет выведено в результате компиляции и запуска следующего кода?

import java.util.\*;

public class Test {

public static void main(String[] a) {

Set ss = new SortedSet(){};

ss.add(null);

System.out.print(ss.size());

}

}

Варианты ответа:

* compile time error
* runtime error
* 0
* [null]

**Вопрос 43:**

Что будет выведено в результате компиляции и запуска следующего кода?

import java.util.\*;

public class Q1 {

public void finalize() {

System.out.println("Oops ");

}

class W {

public void finalize() {

System.out.println("I did it again ");

}

}

public static void main(String[] args) {

Map m1 = new WeakHashMap();

m1.put(new Q1().new W(),"s");

System.gc(); //1

System.out.println(m1);

}

}

Варианты ответа:

* ошибка компиляции
* I did it again Oops {};
* о выводе нельзя сказать ничего определенного из-за строки //1;
* Oops I did it again;
* Oops I did it again {};

**Вопрос 44:**

Что будет выведено в результате компиляции и запуска следующего кода?

import java.util.\*;

public class Q1 {

public void finalize() {

System.out.print("I did it again ");

}

class W {

public void finalize() {

System.out.print("Oops ");

}

}

public static void main(String[] args) {

Map m1 = new WeakHashMap();

m1.put(new Q1().new W(),"s");

m1.put(new HashMap(m1),"s");

System.gc(); //1

System.out.print(m1);

}

}

Варианты ответа:

* ошибка компиляции
* Oops I did it again Oops I did it again {};
* I did it again Oops I did it again Oops {};
* о выводе нельзя сказать ничего определенного из-за строки //1;
* Oops I did it again {};
* I did it again Oops {};
* StackOverflowError;

**Вопрос 45:**

Что будет выведено в результате компиляции и запуска следующего кода?

import java.util.\*;

public class Test {

public static void main(String[] a) {

Set s = new HashSet();

s.add(null);

s.add(new String("s"));

s.add(null);

System.out.print(s);

}

}

Варианты ответа:

* compile time error
* runtime error
* [null, s]
* [null, s, null]

**Вопрос 46:**

Что будет выведено в результате компиляции и запуска следующего кода?

import java.util.\*;

public class Test {

public static void main(String[] a) {

Set s = new HashSet();

s.add(null);

s.add(new String("s"));

s.add(new String());

a = (String[])s.toArray();

System.out.print(a);

}

}

Варианты ответа:

* compile time error
* runtime error
* [null, s]
* hashCode массива a

**Вопрос 47:**

Что будет выведено в результате компиляции и запуска следующего кода?

import java.util.\*;

public class Test {

public static void main(String[] a) {

Set s = new HashSet();

s.add(null);

s.add(new String("s"));

s.add(new String());

a = s.toArray();

System.out.print(a);

}

}

Варианты ответа:

* compile time error
* runtime error
* [null, s]
* hashCode массива a

**Вопрос 48:**

Что будет выведено в результате компиляции и запуска следующего кода?

import java.util.\*;

public class Test {

public static void main(String[] a) {

List l = new ArrayList();

l.add(null);

l.add(new String("string"));

l.add(null);

l.remove(1);

System.out.print(l);

}

}

Варианты ответа:

* compile time error
* runtime error
* [null, null ]
* hashCode l

**Вопрос 49:**

Что будет выведено в результате компиляции и запуска следующего кода?

import java.util.\*;

public class Test {

public static void main(String[] a) {

List l = new ArrayList();

l.add(null);

l.add(new String("string"));

l.add(null);

l.remove(null);

System.out.print(l);

}

}

Варианты ответа:

* compile time error
* runtime error
* [string, null ]
* hashCode l
* [null,string]

**Вопрос 50:**

Что будет выведено в результате компиляции и запуска следующего кода?

import java.util.\*;

public class Test {

public static void main(String[] a) {

List l = new ArrayList();

l.add(null);

l.add(new String("string"));

l.add(null);

l.remove(null);

l.trimToSize();

System.out.print(l);

}

}

Варианты ответа:

* compile time error
* runtime error
* [string, null ]
* hashCode l
* [null,string]

**Вопрос 51:**

В каком классе коллекций возможно обращение к элементам в порядке их вставки?

Варианты ответа:

* PriorityQueue
* LinkedHashSet
* TreeMap
* HashMap
* HashSet

**Вопрос 52:**

Какой из следующих методов определен только в классе Arrays?

Варианты ответа:

* toArray
* binarySearch
* hashCode
* asList
* toString
* swap

**Вопрос 53:**

Какие из следующих объявлений корректны?

Варианты ответа:

* Queue<String> str = new LinkedList<String>();
* Queue<String> str = new PriorityQueue<String>();
* Queue<String> str = new Vector<String>();
* Queue<String> str = new ArrayList<String>();
* Queue<String> str = new LinkedHashMap<String>();

**Вопрос 54:**

Какие объявления корректны?

Варианты ответа:

* List<Object> myList = new ArrayList<JButton>();
* List<JButton> myList = new ArrayList<JButton>();
* List<Number> numbers = new ArrayList<Integer>();
* List<Object> myList = new ArrayList<Object>();

**Вопрос 55:**

Какоие из следующих объявлений верны?

Варианты ответа:

* AbstractList<String> set=new LinkedList<String>();
* Iterable<String> set=new LinkedList<String>();
* Vector<String> set=new LinkedList<String>();
* AbstractCollection<String> set=new LinkedList<String>();

**Вопрос 56:**

выберите правильную полную(без пропусков) цепочку наследования:

(<-- наследование)

Варианты ответа:

* <<interface>>Collection <-- <<interface>>Set <-- Treeset
* <<interface>>Collection <-- <<interface>>List <-- ArrayList
* <<interface>>Map <-- <<interface>>SortedMap <-- HashMap
* <<interface>>Map <-- <<interface>>SortedMap <-- Hashtable

**Вопрос 57:**

Какой метод отсутствует в классе PriorityQueue<String>

Варианты ответа:

* comparator
* iterator
* toArray
* binarySearch

**Вопрос 58:**

Что будет выведено в результате компиляции и выполнения программы?

import java.util.\*;

class Cl2{

public static void main(String[] args){

String[] s={"apple","mango","orange","mango"};

Set set=new HashSet(Arrays.asList(s));

set.add("apple");

set.add("apple");

set.add("pear");

System.out.print(set.size());

}

}

Варианты ответа:

* 3
* 4
* 5
* 6
* 7

**Вопрос 59:**

Что будет выведено в результате компиляции и выполнения следующего кода?

import java.util.\*;

public class CMain {

public static void main(String[] args) {

List l=new ArrayList();

l.add(new Integer(1));

l.add(new Integer(3));

l.add(new Integer(3));

Iterator i=l.iterator();

i.next();

i.next();

System.out.print(i.next());

i.remove();

ListIterator pi=l.listIterator();

pi.add(new Integer(9));

System.out.print(pi.previous());

System.out.print(pi.next());

System.out.print(pi.next());

System.out.print(pi.next());

}

}

Варианты ответа:

* 39913
* Runtime error
* 39339
* нет верного варианта ответа
* Compiller error

**Вопрос 60:**

Что будет выведено в результате компиляции и выполнения следующего кода?

import java.util.\*;

class CMain{

public static void main(String[] args){

Integer[] a={12,5,333,5,221,0,00};

Arrays.sort(a);

List l=Arrays.asList(a);

int in=Arrays.binarySearch(l.toArray(),12);

System.out.println(in);

}

}

Варианты ответа:

* Runtime error
* Compiller error
* 4
* 5
* null

**Вопрос 61:**

Что будет выведено в результате компиляции и выполнения следующего кода?

import java.util.\*;

public class CMain{

public static void main(String[] args){

ArrayList arr = new ArrayList ();

arr.add("B");

arr.add("A");

List list = new ArrayList ();

list = arr;

System.out.print(list);

}

}

Варианты ответа:

* Compiller error
* Throws ClassCastException
* [A,B]
* [B,A]
* ничего не будет выведено

**Вопрос 62:**

В какой строке возникнет ошибка компиляции?

import java.util.\*;

class Shape{}

class Circle extends Shape{}

class Rectangle extends Shape{int x;{x=0;}}

class CMain{

public static void main(String[] args){

Vector picture = new Vector (); // 1

picture.add(new Circle()); // 2

picture.add(new Rectangle()); // 3

Object rect = picture.get(1); // 4

System.out.println(rect.x); // 5

}

}

Варианты ответа:

* //5
* будет выведено 0
* //3
* //4
* //2
* //1

**Вопрос 63:**

Что будет выведено в результате компиляции и выполнения следующего кода?

import java.util.\*;

class CComp implements Comparator{

public int compare(Object a,Object b){

String aStr=(String)a;

String bStr=(String)b;

return aStr.compareTo(bStr);

}

}

class CMain{

public static void main(String[] args){

TreeSet ts=new TreeSet(new CComp());

ts.add("1");

ts.add("5");

ts.add("3");

ts.add("2");

ts.add("4");

Iterator i=ts.iterator();

while(i.hasNext()) {

System.out.print(i.next());

}

}

}

Варианты ответа:

* 15324
* Ошибка компиляции
* Ошибка выполнения
* 12345
* ничего

**Вопрос 64:**

Что будет выведено в результате компиляции и выполения программы?

import java.util.PriorityQueue;

public class Cl5 {

public static void main(String[] args) {

String[] sa = {"f\_f", "\_ff", "ff\_", "FF"};

PriorityQueue<String> pq3 = new PriorityQueue<String>();

for(String s : sa)

pq3.offer(s);

for(String s : sa)

System.out.print(pq3.poll() + " ");

}

}

Варианты ответа:

* ошибка компиляции
* \_ff,f\_f,ff\_,FF,
* FF,f\_f,\_ff,ff\_,
* FF,\_ff,f\_f,ff\_,
* ff\_,f\_f,\_ff,FF,

**Вопрос 65:**

Что будет выведено в результате компиляции и выполения программы?

import java.util.\*;

public class Cl7 {

public static void main(String[] args) {

List<Integer> myList = new ArrayList<Integer>();

myList.add(4);

myList.add(6);

Adder adder = new Adder();

int total = adder.addAll(myList);

System.out.println(total);

}

}

class Adder {

int addAll(List list) {

Iterator it = list.iterator();

int total = 0;

while (it.hasNext()) {

int i = ((Integer) it.next()).intValue();

total += i;

}

return total;

}

}

Варианты ответа:

* ошибка компиляции
* 4
* 10
* 6

**Вопрос 66:**

Что будет выведено в результате компиляции и выполения программы?

import java.util.\*;

public class Cl10 {

public static void main(String[] args) {

AbstractList<String> set=new LinkedList<String>();

set.add("Java");

set.add("XML");

set.add("Swing");

set.add("JSP");

set.add("Java");

for(Object o : set)

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

}

}

Варианты ответа:

* Ошибка компиляции
* Java XML Swing JSP Java
* Java XML Swing JSP
* JSP Java Swing XML
* Java JSP Swing XML

**Вопрос 67:**

Что будет выведено в результате компиляции и выполения программы?

import java.util.\*;

public class cl12 {

public static void main(String[] args) {

List<Integer> s=new ArrayList<Integer>();

s.add(1);

s.add(1);

s.add(1);

s.size();

String[] ss= (String[]) s.toArray();

for(String str : ss)

System.out.print(str);

}

}

Варианты ответа:

* ошибка выполнения
* будет выведено 111
* ошибка компиляции
* будет выведено 1 1

**Вопрос 68:**

Дан код:

import java.util.\*;

public class Cl1{

public static void main(String[] args){

ArrayList<String> arr = new ArrayList<String>();

arr.add("B");

arr.add("A");

// 1

list = arr;

System.out.print(list);

}

}

Что надо вставить в строку //1, чтобы компиляция прошла без ошибок? (выберите 2)

Варианты ответа:

* List list = new ArrayList();
* List<Object> list = new ArrayList<Object>();
* List<Object> list = new List<String>();
* LinkedList<String> list = new LinkedList<String>();
* Collection list = new ArrayList();

**Вопрос 69:**

Дан код

public class Basket<E> {

private E element;

public void setElement(E x) {

element = x;

}

public E getElement() {

return element;

}

}

class Fruit {

}

class Apple extends Fruit {

}

class Orange extends Fruit {

public static void main() {

Basket <Fruit> basket = new Basket <Fruit>();

basket.setElement(new Orange()); // 2

//1

}

}

Что надо подставить в //1, чтобы код скомпилировался без ошибок? (выберите 2)

Варианты ответа:

* Orange orange = basket.getElement();
* Apple orange = basket.getElement();
* Orange orange = (Orange) basket.getElement();
* Fruit orange = basket.getElement();

**Вопрос 70:**

Что надо вставить на место /\*--1--\*/ в коде, чтобы он скомпилировался без ошибок?

class Shape{}

class Circle extends Shape{}

class Rectangle extends Shape{}

class Cl15{

public static void main(String[] args){

Vector picture = new /\*--1--\*/ ();

picture.add(new Circle());

picture.add(new Rectangle());

Rectangle rect = (Rectangle)picture.get(1);

}

}

Варианты ответа:

* Vector<? extends Shape>
* Vector<Shape>
* Vector<?>
* нет правильного ответа

**Вопрос 71:**

Что будет выведено в результате компиляции и выполнения следующего кода?

import java.util.\*;

class CMain{

public static void main(String[] args){

String[] s={"apple","mango","orange","mango"};

Set set=new HashSet(Arrays.asList(s));

set.add("apple");

set.add("pear");

System.out.print(set.toString());

}

}

Варианты ответа:

* Compiler error
* orange, pear, apple, mango
* orange pear apple mango
* [orange, pear, apple, mango]
* Runtime error

**Вопрос 72:**

Какой из следующих методов из java.lang.Comparator требует переопределения?

Варианты ответа:

* int compare(Object o1)
* int compareTo(Object o1, Object o2)
* int equalsTo(Object o1, Object o2)
* int compare(Object o1, Object o2)
* overloaded operator =

**Вопрос 73:**

Что будет выведено в результате компиляции и выполнения следующего кода?

import java.util.\*;

class Cl4 {

public static void main(String [] args) {

String [] sa = {"one", "on", "on\_", "ony"};

Arrays.sort(sa);

ReSortComparator rs = new ReSortComparator();

Arrays.sort(sa,rs);

System.out.println("on\_ = " + Arrays.binarySearch(sa,"on\_",rs));

}

static class ReSortComparator implements Comparator<String> {

public int compare(String a, String b) {

return b.compareTo(a);

}

}

}

Варианты ответа:

* on\_ = 1-неверно
* on\_ = 3-неверно
* on\_ = 4-неверно
* ошибка компиляции
* on\_ = 2

**Вопрос 74:**

Что будет выведено в результате компиляции и выполнения следующего кода?

import java.util.\*;

class Shape{}

class Circle extends Shape{}

class Rectangle extends Shape{}

class cl14{

public static void main(String[] args){

ArrayList<Circle> a=new ArrayList<Circle>();

a.add(new Circle());

ArrayList b=a;

ArrayList<Rectangle> c=b;

c.add(new Rectangle());

System.out.println(c.size());

}

}

Варианты ответа:

* 1
* ошибка компиляции
* null
* 2