@Description

Тест по Java. Средний уровень уровень https://geekbrains.ru/tests/55

@Question

Назовите две группы исключений в Java.

@Options @OnlyOne @Key=2

Exception и Error

Checked и Unchecked

Catched и Uncatched

Handle и Unhandled

@Question

Какой метод отсутствует в классе Object?

@Options @OnlyOne @Key=2

toString()

system()

clone()

finalize()

@Question

Какой из методов не может привести к приостановке потока?

@Options @OnlyOne @Key=2

wait()

notify()

join()

sleep()

@Question

Исключение какого типа будет сгенерировано при выполнении следующего кода:

<code>

Object obj = new Integer(10);

String str = (String) obj;

System.out.println(str);

</code>

@Options @OnlyOne @Key=2

ArrayIndexOutOfBoundsException

ClassCastException

IllegalArgumentException

ни одно из представленных

@Question

Может ли блок try существовать без блока catch?

@Options @OnlyOne @Key=2

нет

да

@Question

Можно ли при использовании класса в качестве элемента HashSet переопределить только метод hashCode(), без переопределения equals()?

@Options @OnlyOne @Key=2

нет

да

@Question

Поля, объявленные в интерфейсах, по умолчанию имеют модификаторы:

@Options @OnlyOne @Key=1

public static final

private static final

public static

static final

@Question

Метод Thread.sleep()

@Options @OnlyOne @Key=3

приостанавливает выполнение потока main

приостанавливает работу программы

приостанавливает работу потока, в котором был вызван без освобождения монитора

приостанавливает работу потока, в котором был вызван с освобождением монитора

приостанавливает выполнение всех потоков, кроме main

@Question

Выберите верное утверждение

@Options @OnlyOne @Key=1

если два объекта равны через equals() их hashCode() должен совпадать, если объекты не равны, их hashCode() может совпадать

если два объекта равны через equals() их hashCode() может не совпадать

если два объекта равны через equals() их hashCode() может совпадать

если два объекта равны через equals() их hashCode() должен совпадать, если объекты не равны, их hashCode() может не совпадать

@Question

Для создания класса checked исключения необходимо

@Options @OnlyOne @Key=3

унаследовать его только от класса Exception

указать ключевое слово checked в объвлении класса

унаследовать его от класса checked исключения

унаследовать его от класса RuntimeException

@Question

Коллекции могут работать только с ссылочными типами данных?

@Options @OnlyOne @Key=2

нет

да

@Question

Какой пакет импортирован по умолчанию?

@Options @OnlyOne @Key=2

java.text

java.lang

java.core

java.util

java.io

@Question

В каком классе или интерфейсе определены методы wait(), notify() и notifyAll()?

@Options @OnlyOne @Key=1

Object

Thread

Runnable

Class

@Question

Допустимо ли в проекте наличие нескольких методов main?

@Options @OnlyOne @Key=1

да

нет

@Question

Как называется класс, объявленный внутри метода?

@Options @OnlyOne @Key=2 http://www.quizful.net/post/inner-classes-java

внутренний

локальный

нельзя объявить класс внутри метода

вложенный

@Question

Утверждение: статический внутренний (вложенный) класс имеет доступ ко всем полям и методам внешнего класса

@Options @OnlyOne @Key=2

верно

не верно

@Question

За что отвечает метод ensureCapacity(int n) в классе ArrayList?

@Options @OnlyOne @Key=2

увеличивает емкость коллекции на n элементов

увеличивает емкость до n, если она меньше n

уменьшает емкость коллекции до n, если она больше n

возвращает true если емкость коллекции больше n

@Question

Выберите неверное утверждение относительно ключевого слова static

@Options @OnlyOne @Key=3

статические методы могут работать только со статическими полями

статические поля общие для всех объектов класса

имеется возможность переопределить(override) статические методы

к статическому полю можно получить доступ через имя класса

@Question

При выполнении следующего кода

<pre><code>

public static void main(String[] args){

new Thread(new Runnable() {

@Override

public void run(){

System.out.println(1);

}

}).start();

new Thread(new Runnable() {

@Override

public void run(){

System.out.println(2);

}

}).start();

}

</code></pre>

@Options @OnlyOne @Key=2

гарантированно будет отпечатано в порядке 1 2

нельзя гарантировать порядок вывода чисел 1 и 2

печать не будет выполнена, так как не вызван метод run()

гарантированно будет отпечатано в порядке 2 1

@Question

В ArrayList поиск по индексу выполняется за одинаковое время?

@Options @OnlyOne @Key=1

да

нет

@Question

Может ли один catch заниматься обработкой и исключений типа NullPointerException, и IOException?

@Options @OnlyOne @Key=2

да

нет

@Question

Может ли HashMap хранить несколько записей с одинаковым ключом?

@Options @OnlyOne @Key=2

нет, при добавлении записи с таким же ключом будет сгенерировано исключение

нет, при добавлении записи с таким же ключом, старое значение будет перезаписано

да, при выполнении метода get() вернется список значений, лежащих под указанным ключом

@Question

Выберите верное утверждение

@Options @OnlyOne @Key=2

у перечислений может быть конструктор с параметрами, и любым модификатором доступа

у перечислений может быть конструктор с параметрами, и только с модификатором доступа private

у перечислений не может быть конструктора

у перечислений имеется только конструктор по-умолчанию

@Question

Выберите неверное утверждение

@Options @OnlyOne @Key=1

интерфейс Map наследуется от интерфейса Collection

интерфейс List не наследуется от интерфейса Collection

интерфейс Set не наследуется от интерфейса Collection

@Question

Какое ключевое слово используется для генерирования и выброса исключений?

@Options @OnlyOne @Key=3

throws

cast

throw

exception

@Question

В перечислениях можно объявлять поля?

@Options @OnlyOne @Key=2

нет

да

@Question

Выберите верное утверждение

@Options @OnlyOne @Key=2

все утверждения верны

HashMap допускает в качестве ключа использовать null, Hashtable - не позволяет

и Hashtable, и HashMap не синхронизированы

и Hashtable, и HashMap синхронизированы

@Question

Утверждения:(1) Интерфейсы могут быть унаследованы от других интерфейсов (2) Перечисления могут быть унаследованы от других перечислений

@Options @OnlyOne @Key=1

оба утверждения верны

оба утверждения неверны

(1) - не верно, (2) - верно

(1) - верно, (2) - не верно

@Question

Выберите верное утверждение

<pre><code>

public static void main(String[] args){

new Thread(new Runnable(){

@Override

public void run(){

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

try{

Thread.sleep(5);

System.out.print(1);

} catch(InterruptedException e){

e.printStackTrace();

}

}

}

}).start();

new Thread(new Runnable(){

@Override

public void run(){

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

try{

Thread.sleep(5);

System.out.print(2);

} catch(InterruptedException e){

e.printStackTrace();

}

}

}

}).start();

}

</code></pre>

@Options @OnlyOne @Key=3

нельзя гарантировать с какой цифры начнётся вывод 1 или 2, но печать будет чередоваться 1212121212 или 2121212121

порядок печати всегда будет 1212121212

порядок вывода гарантировать нельзя

порядок печати всегда будет 2121212121

@Question

Можно ли перехватить OutOfMemoryError с помощью блока try-catch?

@Options @OnlyOne @Key=1

нет

да

@Description

Тест по Java. Средний уровень уровень https://geekbrains.ru/tests/55

@Question

Коллекции могут работать только с ссылочными типами данных?

@Options @OnlyOne @Key=2

нет

да

@Question

Выберите верное утверждение:

@Options @OnlyOne @Key=2

все утверждения верны

HashMap допускает в качестве ключа использовать null, Hashtable - не позволяет

и Hashtable, и HashMap не синхронизированы

и Hashtable, и HashMap синхронизированы

@Question

Обязательно ли классу реализовывать интерфейс Comparable, чтобы его можно было использовать в TreeSet?

@Options @OnlyOne @Key=2 <?>

нет

да

@Question

Что будет результатом выполнения следующего кода:

<pre><code>

public static void main(String[] args) {

try {

return;

} finally {

System.out.println("Finally");

}

}

</code></pre>

@Options @OnlyOne @Key=1

печать в консоль сообщения Finally

RuntimeException

сообщение Finally не будет отпечатано в консоли

ошибка компиляции

@Question

Параметр loadFactory у отображения HashMap по-умолчанию равен:

@Options @OnlyOne @Key=2

0.8

0.75

1.0

0.5

@Question

Какой метод нужно выполнить у класса Thread для запуска нового потока?

@Options @OnlyOne @Key=4

submit()

run()

execute()

start()

@Question

При завершении main-потока автоматически завершается выполнение программы.

@Options @OnlyOne @Key=2

да

нет

@Question

Утверждение: единственный способ запуска новых потоков - создание класса и унаследование его от класса Thread

@Options @OnlyOne @Key=2

да

нет

@Question

При выполнении следующего кода будет fровано исключение:

<pre><code>

public static void main(String[] args){

int[] a = new int[10];

a[20] = 10;

a[5] = a[2] / 0;

}

</code></pre>

@Options @OnlyOne @Key=1

ArrayIndexOutOfBoundsException

IndexOutOfBoundsException

ArrayIndexOutOfBoundsException и ArithmetException

ArithmetException

@Question

Может ли один catch заниматься обработкой и исключений типа NullPointerException, и IOException?

@Options @OnlyOne @Key=2 <?>

да

нет

@Question

Объекты какого класса являются иммутабельными?

@Options @OnlyOne @Key=4

StringBuffer

StringBuilder

ни один из вариантов

String

@Question

Какой пакет импортирован по умолчанию?

@Options @OnlyOne @Key=2

java.text

java.lang

java.core

java.util

java.io

@Question

За что отвечает метод ensureCapacity(int n) в классе ArrayList?

@Options @OnlyOne @Key=4

возвращает true если емкость коллекции больше n

уменьшает емкость коллекции до n, если она больше n

увеличивает емкость коллекции на n элементов

увеличивает емкость коллекции до n, если она меньше n

@Question

Выберите верное утверждение:

@Options @OnlyOne @Key=1

если два объекта равны через equals() их hashCode() должен совпадать, если объекты не равны, их hashCode() может совпадать

если два объекта равны через equals() их hashCode() может не совпадать

если два объекта равны через equals() их hashCode() может совпадать

если два объекта равны через equals() их hashCode() должен совпадать, если объекты не равны, их hashCode() может не совпадать

@Question

Что будет результатом работы предложенного кода?

<pre><code>

public static void main(String[] args){

Thread t1 = new Thread(new Runnable() {

@Override

public void run(){

System.out.println(1);

}

});

Thread t = new Thread(t1);

t.start();

}

</code></pre>

@Options @OnlyOne @Key=3

будет сгенерировано исключение Runt3imeException

в консоль будет выведено число 1

в консоль ничего не будет выведено

ошибка компиляции, невозможно в конструктор одного потока подать доугой поток

@Question

Исключение какого типа сгенерировано при выполнение следующего кода

<code>

Object obj = new Integer(10);

String str = (String) obj;

System.out.println(str);

</code>

@Options @OnlyOne @Key=2

ни одно из представленных

ClassCastException

ArrayIndexOutOfBoundsException

IllegalArgumentException

@Question

Можно ли при использовании класса в качестве элемента HashSet переопределить только метод hashCode(), без переопределения equals()?

@Options @OnlyOne @Key=2

нет

да

@Question

При выполнении следующего кода

<pre><code>

public static void main(String[] args){

new Thread(new Runnable() {

@Override

public void run(){

System.out.println(1);

}

}).start();

new Thread(new Runnable() {

@Override

public void run(){

System.out.println(2);

}

}).start();

}

</code></pre>

@Options @OnlyOne @Key=1 можно использовать

нельзя гарантировать порядок вывода чисел 1 и 2

гарантированно будет отпечатано в порядке 1 2

гарантированно будет отпечатано в порядке 2 1

печать не будет выполнена, так как не вызван метод run()

@Question

Какое ключевое слово используется для генерирования и выброса исключений?

@Options @OnlyOne @Key=2

cast

throw

throws

exception

@Question

Можно ли в классе, унаследованном от классов исключений, объявлять методы и поля?

@Options @OnlyOne @Key=1

да

нет

@Question

Какой из предложенных блоков может запустить новый поток:

<pre><code>

public class Task implements Runnable {

@Override

public void run(){}

public static void main(String[] args){

//Место под блок кода

}

}

</code></pre>

@Options @OnlyOne @Key=3 <?>

Thread t = new Thread(Task);

new Thread(new Task()).start;

ни один из предложенных

Thread t = new Thread(Task); t.start();

Thread t = new Thread(); t.run();

@Question

Может ли поток main завершиться раньше, чем порожденные им потоки?

@Options @OnlyOne @Key=1

да

нет

@Question

Подклассом класса IOException является:

@Options @OnlyOne @Key=4

Exception

NullPointerException

RuntimeException

FileNotFoundException

@Question

Какой из вариантов создания объекта недопустим?

@Options @OnlyOne @Key=4

NavigableMap<String, Integer> map = new TreeMap<>();

Map<String, Integer> map = new HashMap<>();

HashMap<String, Integer> map = new LinkedHashMap<>();

Collection<String, Integer> map = new HashMap<>();

@Question

Как называется класс, объявленный внутри метода?

@Options @OnlyOne @Key=2 http://www.quizful.net/post/inner-classes-java

внутренний

локальный

нельзя объявить класс внутри метода

вложенный

@Question

Блоки try-catch могут быть вложенными?

@Options @OnlyOne @Key=2

нет

да

@Question

Блок finally можно использовать без блока try?

@Options @OnlyOne @Key=2

да

нет

@Question

Утверждение: количество создаваемых потоков ограничивается количеством ядер процессора

@Options @OnlyOne @Key=1

не верно

верно

@Question

Выберите верное утверждение

@Options @OnlyOne @Key=2

ArrayList и Vector синхронизированы

ArrayList - не синхронизирован, Vector - синхронизирован

ArrayList и Vector не синхронизированы

ArrayList - синхронизирован, Vector - не синхронизирован

@Question

Какие модификаторы доступа можно использовать для внешнего класса?

@Options @OnlyOne @Key=3

public, default

default, protected, public

public, private

public, protected

@Description

Тест по Java. Средний уровень уровень https://geekbrains.ru/tests/55

@Question

Оператор join() предназначен для

@Options @OnlyOne @Key=3

объединения нескольких потоков в один

объединения нескольких потоков для их одновременной работы

перевода одного потока в режим ожидания завершения другого потока

обмена данными между потоками

@Question

Можно ли перехватить OutOfMemoryError с помощью блока try-catch?

@Options @OnlyOne @Key=1

нет

да

@Question

Какой из вариантов создания объекта недопустим?

@Options @OnlyOne @Key=3

Map<String, Integer> map = new HashMap<>();

HashMap<String, Integer> map = new LinkedHashMap<>();

Collection<String, Integer> map = new HashMap<>();

NavigableMap<String, Integer> map = new TreeMap<>();

@Question

Можно ли запустить поток через объект типа Thread повторно, после его завершения?

@Options @OnlyOne @Key=2 <?>

да

нет

@Question

Какой из методов не может привести к приостановке потока?

@Options @OnlyOne @Key=1

notify()

wait()

join()

sleep()

@Question

Объекты какого класса являются иммунабельными?

@Options @OnlyOne @Key=4

StringBulder

StringBuffer

ни один из вариантов

String

@Question

Можно ли блок try-with-resources подключить блок finally?

@Options @OnlyOne @Key=1

да

нет

@Question

Можно ли после запуска потока сделать его daemon-потоком?

@Options @OnlyOne @Key=2 <?>

да

нет

@Question

Подклассом класса RuntimeException является

@Options @OnlyOne @Key=4

IOException

Exception

FileNotFoundException

NullPointerException

@Question

Утверждение: статический внутренний (вложенный) класс имеет доступ только к статическим полям и методам внешнего класса

@Options @OnlyOne @Key=2 <?>

не верно

верно

@Question

В чем особенность daemon-потока?

@Options @OnlyOne @Key=4 <?>

запускается при старте приложения

может быть перезапущен сколько угодно раз

не может быть остановлен

завершит свою работу по завершению работы всех обычных потоков

@Question

Запуск потоков возможен только из main-потока.

@Options @OnlyOne @Key=2

да

нет

@Question

Выберите верное утверждение

<pre><code>

public static void main(String[] args){

new Thread(new Runnable() {

@Override

public void run(){

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

try{

Thread.sleep(5);

System.out.print(1);

} catch(InterruptedException e) {

e.printStackTrace();

}

}

}

}).start();

new Thread(new Runnable() {

@Override

public void run(){

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

try{

Thread.sleep(5);

System.out.print(2);

} catch(InterruptedException e){

e.printStackTrace();

}

}

}

}).start();

}

</code></pre>

@Options @OnlyOne @Key=1

порядок вывода гарантировать нельзя

нельзя гарантировать с какой цифры начнётся вывод 1 или 2, но печать будет чередоваться 1212121212 или 2121212121

порядок печати всегда будет 1212121212

порядок печати всегда будет 2121212121

@Question

Выберите неверное утверждение

@Options @OnlyOne @Key=2

все утверждения верны

List, Map, Set являются абстрактными классами

List, Map, Set являются интерфейсами

Классы коллекций находятся в пакете java.util

@Question

Выберите верное утверждение

@Options @OnlyOne @Key=2

ArrayList и Vector синхронизированы

ArrayList - не синхронизирован, Vector - синхронизирован

ArrayList - синхронизирован, Vector - не синхронизирован

ArrayList и Vector не синхронизированы

@Question

При создании объекта типа ArrayList в конструктор в качестве параметра нельзя передать

@Options @OnlyOne @Key=4

массив элементов

объект типа List

объект типа Collection

int

@Question

Может ли HashMap хранить несколько записей с одинаковым ключом?

@Options @OnlyOne @Key=1

нет, при добавлении записи с таким же ключом, старое значение будет перезаписано

да, при выполнении метода get() вернётся список значений, лежащих под указанным ключщь

нет, при добавлении записи с таким же ключом будет сгенерировано исключение

@Question

Выберите верное утверждение

@Options @OnlyOne @Key=1

если два объекта равны через equals() их hashCode() должен совпадать, если объекты не равны, их hashCode() может совпадать

если два объекта равны через equals() их hashCode() должен совпадать, если объекты не равны, их hashCode() не должен совпадать

если два объекта равны через equals() их hashCode() может не совпадать

если два объекта равны через equals() их hashCode() может совпадать

@Question

Параметр loadFactory у отображения HashMap по-умолчанию равен:

@Options @OnlyOne @Key=3

0.5

0.8

0.75

1.0

@Question

Что будет результатом работы предложенного кода?

<pre><code>

public static void main(String[] args){

Thread t1 = new Thread(new Runnable()){

@Override

public void run(){

System.out.println(1);

}

});

Thread t = new Thread(t1);

t.start();

}

</code></pre>

@Options @OnlyOne @Key=3

ошибка компиляции, невозможно в конструктор одного потока подать другой поток

будет сгенерировано исключение RuntimeException

в консоль будет выведено число 1

в консоль ничего не будет выведено

@Question

Укажите суперкласс для классов Exception и Error?

@Options @OnlyOne @Key=3

Catched

Errors

Throwable

Checked

@Question

Для создания класса checked исключения необходимо

@Options @OnlyOne @Key=4

указать ключевое слово checked в объявлении класса

унаследовать его только от класса Exception

унаследовать его от класса RuntimeException

унаследовать его от класса checked исключения

@Question

Выберите неверное утверждение

@Options @OnlyOne @Key=3

интерфейс Set не наследуется от интерфейса Collection

интерфейс List не наследуется от интерфейса Collection

интерфейс Map наследуется от интерфейса Collection

@Question

Какой пакет импортирован по умолчанию?

@Options @OnlyOne @Key=2

java.util

java.lang

java.core

java.io

java.text

@Question

Утверждение: статический внутренний (вложенный) класс имеет доступ только к статическим полям и методам внешнего класса

@Options @OnlyOne @Key=2

верно

не верно

@Question

Какой метод нужно выполнить у класса Thread для запуска нового потока?

@Options @OnlyOne @Key=1

start()

run()

execute()

submit()

@Question

Допустимо объявление пустых интерфейсов?

@Options @OnlyOne @Key=2

нет

да

@Question

Что будет результатом работы предложенного кода?

<pre><code>

public static void main(String[] args){

Thread t1 = new Thread(new Runnable() {

@Override

public void run() {

System.out.println(1);

}

});

Thread t2 = new Thread(new Runnable() {

@Override

public void run() {

try {

t1.join();

} catch (InterruptedException e) {

e.printStackTrace();

}

System.out.println(2);

}

});

t1.start();

t2.start();

}

</code></pre>

@Options @OnlyOne @Key=1

в консоль гарантировано будет выведено 12

один поток созданный с помощью анонимного класса, не может ожидать другой поток

1 и 2 может быть выведено в любом порядке

в консоль гарантировано будет выведено 21

@Question

Выберите верное утверждение о ключевом слове final

@Options @OnlyOne @Key=1

нельзя объявить абстрактный класс с модификаторами final

можно переопределить методы с модификатором final

абстрактный класс может быть объявлен с модификатором final

можно менять значение final поля

@Question

Утверждение: внешний класс имеет доступ ко всем полям и методам внутреннего класса

@Options @OnlyOne @Key=2

верно

не верно

@Description

Тест по Java. Средний уровень уровень https://geekbrains.ru/tests/55

@Question

В перечислениях можно объявить поля?

@Options @OnlyOne @Key=1

да

нет

@Question

Подклассом класса IOException является:

@Options @OnlyOne @Key=1

FileNotFoundException

Exception

RuntimeException

NullPointerException

@Question

Подклассом класса RuntimeException является

@Options @OnlyOne @Key=3

IOException

FileNotFoundException

NullPointerException

Exception

@Question

Как называется класс, объявленный внутри метода?

@Options @OnlyOne @Key=4

нельзя объявить класс внутри метода

внутренний

вложенный

локальный

@Question

Допустимо ли в проекте наличие нескольких методов main?

@Options @OnlyOne @Key=1

да

нет

@Question

Может ли один catch заниматься обработкой и исключений типа NullPointerException, и IOException?

@Options @OnlyOne @Key=2

нет

да

@Question

Блок finally можно использовать без блока try?

@Options @OnlyOne @Key=2

да

нет

@Question

При удалении элемента ArrayList методом remove()

@Options @OnlyOne @Key=4

все элементы равные указанному

случайный элемент, равный указанному

метод remove() удаляет элемент только по индексу

удаляется первый найденный элемент

@Question

В чем особенность daemon-потока?

@Options @OnlyOne @Key=3

не может быть остановлен

может быть перезапущен сколько угодно раз

завершит свою работу по завершению работы всех обычных потоков

запускается при старте приложения

@Question

Выберите неверный способ создания объекта:

@Options @OnlyOne @Key=1

HashSet<String> s = new TreeSet<String>()

HashSet<String> s = new LinkedHashSet<String>()

Set<String> s = new TreeSet<String>()

Set<String> s = new HashSet<String>()

@Question

Параметр loadFactory у отображения HashMap по-умолчанию равен:

@Options @OnlyOne @Key=3

0.8

1.0

0.75

0.5

@Question

Какой из методов не может привести к приостановке потока?

@Options @OnlyOne @Key=4

wait()

join()

sleep()

notify()

@Question

Что будет результатом выполнения следующего кода:

<pre><code>

public static void main(String[] args) {

try {

return;

} finally {

System.out.println("Finally");

}

}

</code></pre>

@Options @OnlyOne @Key=1

печать в консоль сообщения Finally

RuntimeException

ошибка компиляции

сообщение Finally не будет отпечатано в консоли

@Question

Объекты какого класса являются иммутабельными?

@Options @OnlyOne @Key=3

StringBuffer

StringBulder

String

ни один из вариантов

@Question

Выберите верное утверждение

@Options @OnlyOne @Key=4

и внутренний и внешний класс имеют доступ только к статическим членам другого класса

внутренный класс имеет доступ только к открытым полям и методам внешнего класса, внешний класс имеет полный доступ ко всем полям и методам внутреннего

внутренный класс имеет доступ ко всем полям и методам внешнего класса, внешний класс не имеет прямого доступ ко всем полям и методам внутреннего класса

внутренный класс имеет полный доступ ко всем полям и методам внешнего класса, внешний класс имеет полный доступ ко всем полям и методам внутреннего

@Question

LinkedHashSet хранит элементы в порядке

@Options @OnlyOne @Key=1

добавления

случайном

отсортированном через compareTo()

отсортированном по возрастанию hashCode()

@Question

Назовите две группы исключений в Java:

@Options @OnlyOne @Key=4

Handle и Unhandled

Catched и Uncatched

Exception и Error

Checked и Unchecked

@Question

Укажите суперкласс для классов Exception и Error?

@Options @OnlyOne @Key=1

Throwable

Checked

Catched

Errors

@Question

Можно ли бросить объект типа Error?

@Options @OnlyOne @Key=1

да

нет

@Question

Запуск потоков возможен только из main-потока

@Options @OnlyOne @Key=2

да

нет

@Question

Выберите верное утверждение

@Options @OnlyOne @Key=1

HashMap допускает в качестве ключа использовать null, Hashtable - не позволяет

все утверждения верны

и Hashtable, и HashMap не синхронизированы

и Hashtable, и HashMap синхронизированы

@Question

При выполнении следующего кода будет сгенерировано исключение:

<pre><code>

public static void main(String[] args) {

int[] a = new int[10];

a[20] = 10;

a[5] = a[2] / 0;

}

</code></pre>

@Options @OnlyOne @Key=3

IndexOutOfBoundsException

ArithmetException

ArrayIndexOutOfBoundsException

ArrayIndexOutOfBoundsException и ArithmetException

@Question

Какой из вариантов создания объекта недопустим?

@Options @OnlyOne @Key=1

Collection<String, Integer> map = new HashMap<>();

Map<String, Integer> map = new HashMap<>();

NavigableMap<String, Integer> map = new TreeMap<>();

HashMap<String, Integer> map = new LinkedHashMap<>();

@Question

Какой метод в HashMap предоставляет возможность получить ключ по значению?

@Options @OnlyOne @Key=1

такой метод не предусмотрен

getKey()

get()

findKey()

@Question

Утверждение: статический внутренний (вложенный) класс имеет доступ ко всем полям и методам внешнего класса

@Options @OnlyOne @Key=1

не верно

верно

@Question

Что будет результатом работы предложенного кода?

<pre><code>

public static void main(String[] args){

Thread t1 = new Thread(new Runnable()){

@Override

public void run(){

System.out.println(1);

}

});

Thread t = new Thread(t1);

t.start();

}

</code></pre>

@Options @OnlyOne @Key=1

в консоль будет выведено число 1

будет сгенерировано исключение RuntimeException

ошибка компиляции, невозможно в конструктор одного потока подать другой поток

в консоль ничего не будет выведено

@Question

Если несколько потоков должны работать с отображением Map, какой из классов желательно использовать?

@Options @OnlyOne @Key=1

java.util.concurrent.ConcurrentHashMap

java.util.ParallelHashMap

java.util.LinkedHashMap

java.util.HashMap

@Question

Какой метод нужно выполнить у класса Thread для запуска нового потока?

@Options @OnlyOne @Key=1

start()

run()

execute()

submit()

@Question

За что отвечает метод ensureCapacity(int n) в классе ArrayList?

@Options @OnlyOne @Key=1

увеличивает емкость коллекции до n, если она меньше n

возвращает true если емкость коллекции больше n

увеличивает емкость коллекции на n элементов

уменьшает емкость коллекции до n, если она больше n

@Question

Выберите верное утверждение о ключевом слове final

@Options @OnlyOne @Key=1

нельзя объявить абстрактный класс с модификаторами final

можно переопределить методы с модификатором final

можно менять значение final поля

абстрактный класс может быть объявлен с модификатором final

@Description

Тест по Java. Средний уровень уровень https://geekbrains.ru/tests/55

@Question

Выберите неверное утверждение

@Options @OnlyOne @Key=3

LinkedHashSet хранит элементы в порядке добавления

TreeSet представляет собой отсортированное множество

в HashSet элементы упорядочены по возрастанию

в HashSet хранятся только уникальные элементы

@Question

Допустимо объявление пустых интерфейсов?

@Options @OnlyOne @Key=1

да

нет

@Question

При попытке вызвать метод у не созданного объекта будет сгенерировано исключение:

@Options @OnlyOne @Key=1

NullPointerException

IllegalArgumentException

IOException

ClassCastException

@Question

Какое условие необходимо выполнить для использования класса в качестве ключа HashMap?

@Options @OnlyOne @Key=1

переопределить методы hashCode() и equals()

переопределить только метод equals()

переопределить только метод hashCode()

реализовать интерфейс Serializable

@Question

Выберите верное утверждение

@Options @OnlyOne @Key=4

RuntimeException - checked, Exception - unchecked

RuntimeException - unchecked, Exception - unchecked

RuntimeException - checked, Exception - checked

RuntimeException - unchecked, Exception - checked

@Question

При выполнении static synchronized метода в роли монитора выступает?

@Options @OnlyOne @Key=2

объект, у которого вызвали этот метод

класс

нельзя указывать модификатор synchronized у static метода

одно из полей объекта

@Question

Выберите верное утверждение

@Options @OnlyOne @Key=4

внутренний класс имеет полный доступ ко всем полям и методам внешнего класса, внешний класс имеет полный доступ ко всем полям и методам внутреннего

и внутренний и внешний класс имеют доступ только к статическим членам другого класса

внутренний класс имеет доступ только к открытым полям внешнего класса, внешний класс имеет полный доступ ко всем полям и методам внутреннего

внутренний класс имеет доступ ко всем полям и методам внешнего класса, внешний класс не имеет прямого доступа к полям и методам внутреннего класса

@Question

Существует ли возможность унаследовать класс от абстрактного и не реализовать абстрактные методы родительского класса?

@Options @OnlyOne @Key=1

нет

да