

Nome:
Número USP:

UNIVERSIDADE DE SÃO PAULO - EACH
PRIMEIRA PROVA COMPUTAÇÃO ORIENTADA A OBJETOS
2011

SEMESTRE : III
PROFESSORA: Karina Valdivia Delgado
CURSO: Sistemas de Informação

1. Sendo as afirmações:

- I) O bloco **finally** quase sempre será executado, independentemente de ter ocorrido uma exceção ou de esta ter sido tratada ou não.
- II) O bloco **finally** sempre será executado, inclusive quando o aplicativo fechar antes chamando o método **System.exit()**
- III) **Instruções** de liberação de recursos podem ser colocadas no bloco **finally**

Está(ão) correto(s) apenas o(os) item(ns):

- A) II
B) III
C) I e II
D) I e III
E) II e III

0.5

2. Sendo as afirmações:

- I) Para poder comparar objetos podemos implementar a interface **Comparador** ou implementar a interface **Comparable**.
- II) Para adicionar um objeto a uma tabela **hash** é calculado o **hashCode** do objeto. Para que isso funcione corretamente é necessário verificar que o método **hashCode** de cada objeto retorne o mesmo valor para dois objetos, se eles são considerados iguais.
- III) O método **compareTo** da interface **Comparador** compara dois objetos e retorna um inteiro negativo se o primeiro for menor do que o segundo; zero, se forem idênticos; e um valor positivo, caso contrário.

Está(ão) correto(s) apenas o(os) item(ns):

- A) II
B) III
C) I e II
D) I e III
E) II e III
F) I, II e III

0.5

3. Dado o mapa a seguir, incluir a linha que falta para apagar todos os pares cuja chave é maior ou igual que 455 (use o método **tailMap**):

```
Map<Integer, String > mapa = new TreeMap<Integer,String>();  
mapa.put(455,"vermelho");  
mapa.put(333,"branco");  
mapa.put(678,"amarelo");  
mapa.put(455,"azul");
```

0.5

4.- Escreva duas diferenças entre **TreeSet** e **HashSet**:

TreeSet

HashSet

0.5

5.- Porque quando inserimos a seguinte linha num programa:

```
Thread.sleep(2000); //pause for 2 seconds
```

Acontece um erro de compilação?

0.5

Nome:

Número USP:

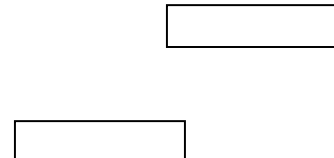
6.Existe uma classe Livro que tem o método **toString** que devolve o nome do livro e o preço. Dado o seguinte código:

```
List<Livro> list = new LinkedList< Livro>();
Livro s1=new Livro("Est. de Dados",145);
Livro s2=new Livro("Java",150);
Livro s3=new Livro("C++",120);
Livro s4=new Livro("Redes",130);
list.add(0, s1);
list.add(1, s2);
list.add(2, s3);
ListIterator<Livro> it= list.listIterator(list.size());
System.out.print(it.previous());
it.add(s4);
System.out.print(list);
```

O que será mostrado na tela quando o programa é executado?



7.-Completar a figura com as classes que implementam a interface **List**:



0.5

1

8.- Incluir as linhas de código para escrever **tree** no arquivo **data.ser** (use serialização)

```
private static final String names[] = { "amarelo", "verde", "preto", "marrom", "cinza", "branco",
"laranja", "vermelho", "verde" };
SortedSet< String > tree = new TreeSet <String> (Arrays.asList( names) );
try{
```



```
}
} catch (FileNotFoundException e) {
e.printStackTrace();
}
catch (IOException e) {
e.printStackTrace();
}
```

1

Nome:

Número USP:

9. Dada a seguinte classe que contém uma lista de nomes de animais e a quantidade correspondente de animais que existem num zoológico:

```
public class Zoo {  
    private List<String> animalList;  
    private List<Integer> quantityList;
```

```
    ...  
}
```

escrever os seguintes métodos:

construtor: que cria duas listas do tipo **ArrayList** vazias.

0.5

método que imprime todos os animais do zoológico que estão com problemas de reprodução (quantidade menor ou igual que 2)

```
public void printAnimals(){
```

```
}
```

1

método que remove todos os animais do zoológico (incluindo a quantidade correspondente) que estão com problemas de reprodução, ou seja aqueles cuja quantidade é menor ou igual que 2

```
public void removeAnimals(){
```

```
}
```

1.5

Nome:
Número USP:

Escreva um método que insere um nome de animal e uma quantidade no final das listas, sempre que esse animal não esteja já na lista.

```
public void insertAnimal(String name, Integer number){  
  
  
  
  
  
  
  
  
  
}
```

0.5

Escreva um método que devolva o maior número que existe na lista quantityList (use o método de **Collections**)

```
public Integer maxQuantityList(){  
  
  
  
  
  
  
  
  
  
}
```

0.5

10. Completar:

- A) Arquivos binários são criados com base em fluxos de _____
- B) Para entrada baseada em caracteres é usada a classe _____
- C) Para saída baseada em caracteres é usada a classe _____
- D) O compilador assegura que a exceção verificada é capturada (via blocos _____/_____) ou declarada em uma cláusula _____.
- E) Todas as subclasses de **RuntimeException** são exceções _____.

1