JAVA

1 - Escreva um código que a partir da seguinte lista, imprima apenas as palavras iniciadas com a letra “a”, independente se maiuscula ou minuscula.

“Pedro”, “Maria”, “Joana”, “André”, “Carlos”, “anna”, “augusto”, “Henrique”;

class Nome{

public static void main(String args[]){

StringBuffer palavra = new StringBuffer();

String nome = args[0];

for(int i = 0; i<nome.length();i++){

palavra.append(nome.charAt(i));

System.out.println(palavra.toString());

}

for(int x = palavra.length()-1; x>=0; x--){

palavra.deleteCharAt(x);

System.out.println(palavra.toString());

Scanner entrada = new Scanner(System.in);

System.out.print("Digite uma String: ");

String palavra = entrada.nextLine();

palavra.toLowerCase(); // Esta instrução não altera a palavra digitada.

System.out.println("\nPalavra digitada: " + palavra); // Irá exibir a palavra exatamente como ela foi digitada.

System.out.println("Palavra alterada: " + palavra.toLowerCase()); // Irá exibir a palavra com todas as letras minúsculas.

System.out.println("Palavra digitada: " + palavra); // Veja novamente que a palavra não é alterada.

}

}

}

2 - Escreva um código que demonstre a utilização de um Map.

import java.util.HashMap;

import java.util.Map;

public class TestaInterfaceMap {

public static void main(String[] args) {

Map<integer, string=""> mapaNomes = new HashMap<integer, string="">();

mapaNomes.put(1, "Alex Rogério");

mapaNomes.put(2, "Wang Jiao Luo XinXin");

mapaNomes.put(3, "Brasil");

System.out.println(mapaNomes);

//resgatando o nome da posição 2

System.out.println(mapaNomes.get(2));

}

}

**Código completo**

import java.util.Collection;

import java.util.HashMap;

import java.util.Iterator;

import java.util.Map;

import java.util.Set;

import java.util.Map.Entry;

public class TestaInterfaceMap {

public static void main(String[] args) {

Map<integer, string=""> mapaNomes = new HashMap<integer, string="">();

mapaNomes.put(1, "Alex Rogério");

mapaNomes.put(2, "Wang Jiao Luo XinXin");

mapaNomes.put(3, "Brasil");

//Collection<integer> conjNomes = mapaNomes.keySet();

Set<entry<integer, string="">> set = mapaNomes.entrySet();

Iterator it = set.iterator();

System.out.println("Código\t\tValor");

//getKey() - recupera a chave do mapa

//getValue() - recupera o valor do mapa

while(it.hasNext()){

Entry<integer, string=""> entry = (Entry)it.next();

System.out.println(entry.getKey() + "\t\t"+entry.getValue());

}

}

}

SQL

1 - Construa um modelo capaz de armazenar as informações de um filme e os atores dele,

lembrando apenas que um mesmo ator pode participar de diferentes filmes.

CREATE TABLE Ator (

id int(11) NULL,

id\_filme int(11) NULL,

valor\_aluguel int(11) NULL,

data\_Ano datetime NULL,

PRIMARY KEY(id)

)

ALTER TABLE `Ator`

ADD CONSTRAINT `fk\_id\_filmes`

FOREIGN KEY(`id\_filme`)

REFERENCES `filmes`(`id`)

INSERT INTO alugueis(id, id\_filme, valor\_aluguel, data\_aluguel)

VALUES(1, 1, 2, '2008-09-10 5:29:57.0')

GO

INSERT INTO alugueis(id, id\_filme, valor\_aluguel, data\_aluguel)

VALUES(2, 1, 1, '2008-09-13 5:29:57.0')

GO

INSERT INTO alugueis(id, id\_filme, valor\_aluguel, data\_aluguel)

VALUES(3, 1, 1, '2008-09-16 5:29:57.0')

GO

INSERT INTO alugueis(id, id\_filme, valor\_aluguel, data\_aluguel)

VALUES(4, 1, 1, '2008-09-18 5:29:57.0')

GO

INSERT INTO alugueis(id, id\_filme, valor\_aluguel, data\_aluguel)

VALUES(5, 1, 1, '2008-09-20 5:29:57.0')

GO

CREATE TABLE filmes (

id int(11) NOT NULL DEFAULT '0',

custo int(11) NOT NULL,

categoria int (11) NOT NULL,

data\_Ano datetime NULL,

PRIMARY KEY(id)

)

INSERT INTO filmes(id, custo)

VALUES(1, 4)

GO

SELECT \* FROM ator;

SELECT filme.titulo, categoria.nome

FROM filme, filme\_categoria, categoria

WHERE filme.filme\_id =

filme\_categoria.filme\_id

AND categoria.categoria\_id =

filme\_categoria.categoria\_id

ORDER BY filme.titulo

2 - Com base na questão anterior (script dos filmes), desenvolva uma query capaz de

retornar os filmes em que “Keanu Reeves” já participou.

SELECT filme.titulo

FROM filme, filme\_ator, ator

WHERE filme.filme\_id =

filme\_ator.filme\_id

AND ator.ator\_id =

filme\_ator.ator\_id

AND ator.primeiro\_nome = 'Keanu'

AND ator.ultimo\_nome='Reeves'