Technical Test

DEV - D

Date:15/05/2024_____City/State: Recife, Pernambuco__

Course: Ciência da computação_____EducationalInstitution:_____CESAR_____School

Course Duration (in years):4Current Semester:3Graduation Year (expected):2026 _ Availability to work: 20h☐ x30h☐ ☐40h Estimated Start Date:

_____

Instructions:

This test consists of 8 multiple choice questions, 1 algorithm implementation and 1 non-technical question. The algorithm is worth 60% of the total score. The non-technical question must be answered in Portuguese.

You may use any blank space on this test as a draft.

Use the table below to record your answers.

Good luck!

Answer Sheet

	1	2	3	4	5	6	7	8
A							x	x
B			x					
C	x	x	x					
D				x	x			

Question 1

Given
:

```
1.   public class A {
2.   public void print(){ System.out.println("x");  } 3.
     }
4.   public class B extends A {
5.   public void print(){ System.out.println("y");  } 6.
     }
7.   public class Main{
8.       public static void main(String[] args) {
9.           B test = new A();
10.          test.print();
11.      }
12. }
```

What's the output?

A. x

B. y

C. Runtime or Compilation error on line 9

D. Runtime or Compilation error on line 10

Question 2

Given:

Consider the following
statements:

I. Line 18 is
not valid because
y is protected on
A

II. Line 19 is not valid
because x is private

III. Line 20 is valid
because m1() is
public

```
1. //***********************
2. // file A.java
3. //***********************
4. public class A {
     private int x;
     protected int y;
     public int m1() {return x
     }
10. //***********************
11. // file B.java
12. //***********************
     package b;

     public class B extends A
     private int z;
     public void m2(A a){
```

A.   Only I and II are
     correct

B.   Only I and III are
     correct

C.   Only II and III are
     correct

D.   I, II and III are
     correct

```
18.          z = y;
19.          z = a.x;
```

```
20.          z = a.m1();
21.      }
22. }
```

Question 3

What are the major elements in an object model?

A. Abstraction, encapsulation and persistence
B. Hierarchy, persistence and typing
C. Abstraction, encapsulation and hierarchy
D. Hierarchy, concurrency and typing

Question 4

Suppose the keys `{1,2,3,4,6,9}` are inserted into a hash table with collisions resolved by chaining. Let the table have 3 slots, and the hash function be `h(k) = k % 3`, that is, `h(k)` returns the remainder of the Euclidean division of k by 3. How will the keys be organized into this hash table?

A. 1->4
    2->6
    3->9

B. 3->6->9
    1->4
    2

C. 3->4
    1->6
    2->9

D. 1
    2->4
    3->6->9

Question 5

Consider the following statements:

I.   A Binary Tree is a tree data structure in which each node has at most two child nodes, usually distinguished as "left" and "right", and a tree with n nodes has exactly $n-1$ branches.
II.   A Hash Map is a data structure in which, if there's no collision among the keys, you can always find an element in O(1) time, even in the worst case.
III. In a doubly linked list data structure, each item is linked to both the previous and the next items in the list, allowing easy access of list items backwards as well as forwards.

A. Only II and III are correct
B. Only II is correct
C. Only III is correct
D. I, II and III are correct

Question 6

In the following code, assume that Queue is not thread-safe, there is more than one Producer thread and more than one Consumer thread running and this program is crashing on runtime. In order to fix the code below how should you fill in lines (1), (2), (3) and (4)?

Global variables	
`Queue q;` **(1)**	
Producer thread	**Consumer thread**

`runProducer()`
`  {`
`  while(true)`
`  {`
`    item = new item();`
`    `**(2)**
`    if (q is not`
`      full){`
`      q.enqueue(item);`
`      `**(3)**
`    }`
`    `**(4)**
`  }`
`}`
	`runConsumer()`
	`  {`
	`  while(true)`
	`  {`
	`    `**(2)**
	`    if (q is not empty){`
	`      item =`
	`      q.dequeue(); `**(3)**
	`    }`
	`    `**(4)**
	`  }`
	`}`

A. (1)
   (2)
   (3) `if(Consumer) sleep(1); else sleep(2);`
   (4)

B. (1) `mutex m;`
   (2)
   (3) `m.lock();`
   (4) `m.unlock();`

C. (1) `semaphore guard;`
   (2) `wait(guard);`
   (3)
   (4) `signal(guard);`

D. Alternatives A, B and C are all correct.

Question 7

Considering the following tables and data information, what would be the correct result of the SQL command below?

Orders				
Number	Order_Date	cust_id	salesperson_id	Amount
10	8/2/2010	4	2	540
20	5/6/2012	9	7	150
30	3/12/2012	8	5	1,500
40	1/30/2013	4	8	1,800
50	7/14/2009	9	1	460
60	1/29/2012	7	2	2,400
70	2/3/2012	6	7	600
80	4/1/2013	8	2	2,300
90	3/2/2012	6	7	720

```
SELECT Salesperson.Name from Salesperson
WHERE Salesperson.ID NOT IN(
    SELECT Orders.salesperson_id FROM Orders
    INNER JOIN Customer ON Orders.cust_id = Customer.ID WHERE
    Customer.Name = 'Samsonic')
AND Salesperson.ID IN
    (SELECT DISTINCT Orders.salesperson_id FROM Orders);
```

A. Abe
   Chris
   Dan

B. Abe
   Bob
   Chris
   Dan

C. Abe
   Chris
   Dan
   Joe

D. Bob
   Ken

Question 8

Given this output on a Linux terminal:

```
$ cat unix_like_systems.txt
MINIX operating system
UNIX operating system
Linux operating system
MINIX and UNIX operating system
MINIX and Linux operating system
UNIX and Linux operating system
```

What will be the correct result of the command below?

```
$ cat unix_like_systems.txt | grep UNIX | sort
```

- MINIX   and   UNIX   operating
  system   UNIX   and   Linux
  operating   system   UNIX
  operating system

- operating system UNIX
  and MINIX operating system
  UNIX and Linux operating
  system UNIX

- and   MINIX   operating   system
  UNIX   and   Linux   operating
  system UNIX operating system
  UNIX

- UNIX operating system
  MINIX and UNIX operating
  system UNIX and Linux
  operating system

Infinite Coins

Given an infinite number of quarters (25 cents), dimes (10 cents), nickels (5 cents) and pennies (1 cent), write a method **makeChange()** to calculate and return a Set containing the ways of representing n cents using those coins. Each element of the returned Set should represent the number of coins to compose the entry value, an array like this: `[quarters, dimes, nickels, pennies]`.

The method **makeChange()** can use a Set data structure to store each representation of n cents, and then, return it. A **Set** is a collection that contains no duplicate elements and the order of elements is irrelevant. Consider the following interface defined for Set:

Method signature	Method description
`boolean add(Element e)`	Adds the specified element to this set if it is not already present (optional operation).
`boolean addAll(Set s)`	Adds all elements from s that are not already present in this set.
`boolean contains(Element e)`	Returns true if this set contains the specified element.
`boolean equals(Set s)`	Compares the specified set s with this set for equality.
`Iterator<Element> iterator()`	Returns an iterator over the elements in this set.
`boolean remove(Element e)`	Removes the specified element from this set if it is present (optional operation).
`int size()`	Returns the number of elements in this set (its cardinality).
`Element[] toArray()`	Returns an array containing all of the elements in this set.

Table: Set interface

Input example:
    n=12

Output for the given example:
   [[0,0,0,12], [0,0,1,7], [0,0,2,2], [0,1,0,2]]*

* this is the content of the **Set** which should be returned by the function.

Your proposed solution can be written in **pseudo-code** or any well-known language (C, C++, Java, etc) and you are free to implement any auxiliary functions. Besides, write down a comment to the implemented function, explaining how your function will work like the one below.

```
/**
 * The function below will ...
 * - Obtain the input
 * - Iterate over the elements
 * …
 * - Print the output and return ...
 */
```

Algorithm Solution

disponivel no link: https://github.com/arthursuzuki/respostasencora/blob/main/InfiniteCoins.java
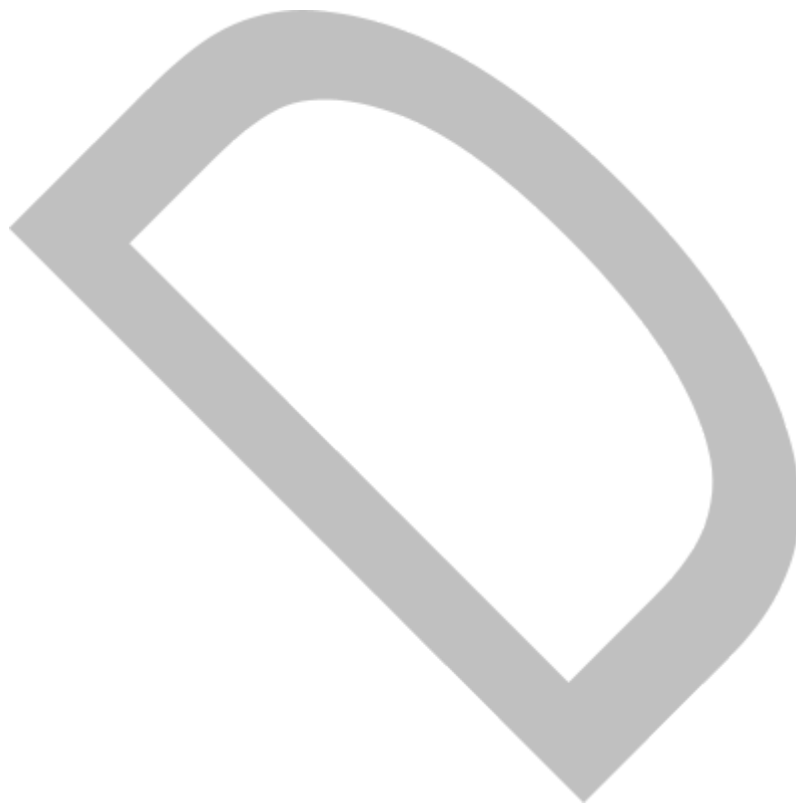
Qual a disciplina que você mais gostou de cursar na faculdade e por quê? (Responder em português)

Em Fundamentos de Desenvolvimento de Software (FDS), pude aprender como desenvolver um software utilizando metodologias ágeis, aplicando a teoria, na prática. No caso, o professor ensinava apenas a teoria e cabia a nós estudantes pesquisarmos sobre tudo que nos era requisitado. Nesse caso, tivemos que pesquisar como criar e desenvolver e manter um projeto Django, como fazer um deployment de uma aplicação web na Azure e testes automatizados usando Selenium. Além de termos bastantes hard skills, como o projeto era feito em grupo de 6, seguindo a regra das 2 pizzas, usamos pair programming, que foi fundamental para praticarmos e desenvolvermos as soft skills de comunicação e trabalho em equipe.