

Prova 2

Instruções

- Escreva as respostas com caneta.
- Não é permitido realizar qualquer tipo de consulta.

- 1) (1,0) Explique e diferencie método estático, abstrato e sobrescrito?
- 2) (1,0) Explique e diferencie classe abstrata, concreta e interface.
- 3) (2,0) Observe o código C++ abaixo e responda as questões.

```
class Entity {
protected:
    long id; static int count;
public:
    Entity(long id) {
        this->id = id;
        Entity::count++;
    }
    virtual void display() {
        cout << "\n" << this->id << " ";
    }
};
// inicia campo estático
int Entity::count = 0;
```

```
class Person : public Entity {
private:
    string name;
public:
    Person(long id, string name):
        Entity(id) {
        this->name = name;
    }
    void display() {
        Entity::display();
        cout << this->name;
        cout << " #" << Entity::count;
    }
};
```

a) Qual a saída impressa pelo seguinte código?

```
Entity* list[2];
list[0] = new Person(10, "Adam Jensen");
list[1] = new Entity(20);
for (Entity* e : list)
    e->display();
```

b) Em C++, a palavra virtual indica que o método pode ser sobrescrito na subclasse. Ela dá suporte ao polimorfismo. Explique e exemplifique o polimorfismo em Orientação a Objetos.

4) (2,5) Considerando a linguagem funcional LISP:

- a) Escreva uma função LISP que recebe uma lista de números e devolve a quantidade de valores positivos que estão na lista.
- b) Escreva uma função **recursiva** LISP que imprime os números ímpares de N até 2. A função recebe N como parâmetro.

5) (1,0) Observe o código C# abaixo e responda as questões.

- a) Qual a saída do programa quando a chamada da linha 11 é feita com `division(25, 5)`?
- b) Qual a saída do programa quando a chamada da linha 11 é feita com `division(-10, 5)`?
- c) Qual a saída do programa quando a chamada da linha 11 é feita com `division(5, 0)`?

```
01  public class Program {
02      public float division(int num1, int num2) {
03          float res = 0;
04          res = num1 / num2;
05          if (num1 < 0) throw new Exception("We don't like negative numbers!");
06          return res;
07      }
08      public static void Main(string[] args) {
09          float res = 0;
10          try {
11              res = new Program().division(?, ?);
12          } catch (DivideByZeroException e) {
13              Console.WriteLine("Division By Zero");
14          } catch (Exception e) {
15              Console.WriteLine(e.Message);
16          } finally {
17              Console.WriteLine("Result: {0}", res);
18          }
19      }
20  }
```

6) (2,5) Considere a seguinte base de fatos em Prolog:

<code>trabalha(joao, microsoft).</code>	<code>funcao(joao, engenheiro).</code>
<code>trabalha(paula, microsoft).</code>	<code>funcao(paula, engenheiro).</code>
<code>trabalha(carla, microsoft).</code>	<code>funcao(carla, suporte).</code>
<code>trabalha(marcio, microsoft).</code>	<code>funcao(jonathan, testes).</code>
<code>trabalha(jonathan, microsoft).</code>	<code>funcao(regina, engenheiro).</code>
<code>trabalha(regina, apple).</code>	
	<code>chefe(jonathan, suporte).</code>
	<code>chefe(marcio, engenheiro).</code>

a) Escreva as seguintes pesquisas:

- Marcio é chefe?
- Quem é engenheiro?
- Lista de pessoas e funções
- Quem trabalha na microsoft e é engenheiro?

b) Escreva as seguintes regras:

- Duas pessoas são colegas de trabalho se trabalham na mesma empresa.
- Uma pessoa trabalha com desenvolvimento em uma empresa se ela trabalha na empresa e tem como função 'engenheiro' ou 'testes'. OBS: você pode usar parênteses para expressões lógicas. Ex: `A , (B ; C)`
- Uma pessoa é chefe de outra se ambas trabalham na mesma empresa e a primeira é chefe da função da segunda