



---

**Curso:** Engenharia Eletrônica, **Disciplina:** Fundamentos de Programação II (IF62C), **Turma:** S\_\_  
**Professores:** Hermes Del Monego ( ), Jean Simão ( ), Robinson Vida ( ) **Data:** \_\_/\_\_/\_\_.

**Aluno:** \_\_\_\_\_ **Código:** \_\_\_\_\_, **Início** \_\_:\_\_:\_\_ **Fim** \_\_:\_\_:\_\_

---

## 2ª Prova

1) [2,5 pt] Crie uma classe que representa um ponto no plano cartesiano. Em seguida, crie uma classe que representa um triângulo, reusando a classe anterior por composição. Ambas as classes devem ter os atributos apropriados e não públicos, bem como os métodos apropriados (e.g. construtores, destrutores, sets e gets). É importante salientar que não é necessário verificar se os pontos formam um triângulo. Finalmente, escreva um programa que receba do usuário as coordenadas dos vértices do triângulo e imprima o valor do perímetro. <sup>1</sup>

2) [2,0 pt] Complete o que está faltando nas seguintes afirmações.

- I) \_\_\_\_\_ é uma forma de reuso na composição de software na qual novas classes absorvem os conjuntos de dados e de comportamentos definidos nas classes existentes e ampliam essas classes com novas capacidades.
- II) Em um relacionamento \_\_\_\_\_, um objeto de uma classe derivada pode também ser tratado como um objeto da classe base.
- III) Em um relacionamento \_\_\_\_\_, um objeto tem um ou mais objetos de outras classes como membros ou atributos.
- IV) Se uma classe contém pelo menos uma função virtual pura, essa classe é uma classe \_\_\_\_\_.
- V) Uma função membro ou método em uma classe pode ser definido como "static" se ele não acessar nenhum membro da classe que seja \_\_\_\_\_.

---

<sup>1</sup>Questão adaptada de prof. Vítor E. Silva Souza - UFES

3) [3,0 pt] Um programador definiu uma classe para delimitar as coordenadas de uma área retangular dentro de uma superfície de duas dimensões X e Y. O código da classe é apresentado no quadro a seguir:

**Definição em C++ da classe AreaRetangular**

```
class AreaRetangular
{
public:
    typedef struct Ponto { int x; int y; } Ponto;
private:
    Ponto p1, p2;
public:
    AreaRetangular(int $x1,int $y1,int $x2,int $y2)
        { p1.x = $x1; p2.x = $x2; p1.y = $y1; p2.y = $y2; }

    virtual ~AreaRetangular(){}

    Ponto getPonto1(){ return p1;}

    Ponto getPonto2(){ return p2;}

    void setPonto1(int x, int y) { p1.x = x; p1.y = y;}

    void setPonto1(Ponto v) { p1=v;}

    void setPonto2(int x, int y) { p2.x = x; p2.y = y;}

    void setPonto2(Ponto v) { p2=v;}

    bool existeSobreposicao(AreaRetangular *v)
    {
        bool saida = false;
        if ( (this->p1.x > v->p1.x) && (this->p1.x < v->p2.x)
            && (this->p1.y > v->p1.y) && (this->p1.y < v->p2.y) ) saida = true;

        /* Códigos estão faltando aqui para o correto funcionamento da classe*/

        return saida;
    }
};
```

Ao tentar utilizar a classe, ele verificou que o método "bool existeSobreposicao(AreaRetangular \*v)" não funcionou de acordo com o esperado. Não ocorreram algumas situações nas quais esse método deveria identificar a sobreposição. Identifique, nas opções a seguir, se as opções de códigos deveriam ser inseridos no método "bool existeSobreposicao(AreaRetangular \*v)" de tal forma que a função realize o esperado. Salienta-se que os pontos coincidentes devem ser desconsiderados.

- |   |   |
|---|---|
| a) [ V ] / [ F ]<br>if ( (this->p2.x > v->p2.x)<br>&& (this->p2.x > v->p2.x)<br>&& (this->p1.y > v->p1.y)<br>&& (this->p1.y > v->p2.y)) saida = true; | b) [ V ] / [ F ]<br>if ( (this->p1.x > v->p1.x)<br>&& (this->p1.x < v->p2.x)<br>&& (this->p2.y > v->p1.y)<br>&& (this->p1.y < v->p2.y)) saida = true; |
| c) [ V ] / [ F ]<br>if ( (this->p2.x < v->p1.x)<br>&& (this->p2.x < v->p2.x)<br>&& (this->p1.y > v->p1.y)<br>&& (this->p1.y > v->p2.y)) saida = true; | d) [ V ] / [ F ]<br>if ( (this->p1.x > v->p1.x)<br>&& (this->p1.x < v->p2.x)<br>&& (this->p1.y < v->p1.y)<br>&& (this->p1.y < v->p2.y)) saida = true; |
| e) [ V ] / [ F ]<br>if ( (this->p2.x > v->p1.x)<br>&& (this->p2.x > v->p1.x)<br>&& (this->p1.y > v->p2.y)<br>&& (this->p1.y < v->p2.y)) saida = true; | f) [ V ] / [ F ]<br>if ( (this->p2.x > v->p1.x)<br>&& (this->p2.x < v->p2.x)<br>&& (this->p2.y > v->p1.y)<br>&& (this->p2.y < v->p2.y)) saida = true; |
| g) [ V ] / [ F ]<br>if ( (this->p2.x < v->p1.x)<br>&& (this->p2.x < v->p2.x)<br>&& (this->p1.y < v->p1.y)<br>&& (this->p1.y < v->p2.y)) saida = true; | h) [ V ] / [ F ]<br>if ( (this->p2.x > v->p1.x)<br>&& (this->p2.x < v->p2.x)<br>&& (this->p1.y > v->p1.y)<br>&& (this->p1.y < v->p2.y)) saida = true; |

4) [2,5 pt] O que será impresso na tela se o programa a seguir for compilado e executado?

```
#include <iostream>

class A
{
public:
    virtual ~A(){std::cout << "Liberando A. \n";}
    virtual void print() { std::cout << "Sou da classe A \n";}
};

class B: public A
{
public:
    virtual ~B() {std::cout << "Liberando B. \n";}
    void print(){std::cout << "Sou da classe B \n";}
};

int main()
{
    A *elementos[3];
    B outroElemento;

    elementos[0] = new A();
    elementos[1] = new B();
    elementos[2] = &outroElemento;

    elementos[0]->print();
    elementos[1]->print();
    elementos[2]->print();

    delete elementos[0];
    delete elementos[1];

    return 0;
}
```

Marque a alternativa correta.

- |                 |                 |
|-----------------|-----------------|
| a)              | b)              |
| Sou da classe B | Sou da classe A |
| Sou da classe A | Sou da classe B |
| Sou da classe A | Sou da classe B |
| Liberando A.    | Liberando A.    |
| Liberando A.    | Liberando B.    |
| Liberando A.    | Liberando A.    |
| Liberando B.    | Liberando B.    |
| Liberando B.    | Liberando A.    |
| c)              | d)              |
| Sou da classe A | Sou da classe A |
| Sou da classe A | Sou da classe B |
| Sou da classe B | Sou da classe A |
| Liberando A.    | Liberando B.    |
| Liberando A.    | Liberando B.    |
| Liberando B.    | Liberando A.    |
| Liberando B.    | Liberando A.    |
| Liberando A.    | Liberando B.    |

**Formulário (Apenas sugestão, poderá não estar completo)**

---

Dados:

Equação geral da reta:  $ax + by + c = 0$

Área do triângulo:  $A = \frac{b \cdot h}{2}$

Perímetro de um triângulo:  $P = a + b + c$ , onde a,b,c são os lados do triângulo.

Perímetro de um círculo:  $P = 2 * \pi * r$

Área de um círculo:  $\pi r^2$

Distância Euclidiana:  $d = \sqrt{(y_a - y_b)^2 + (x_a - x_b)^2}$ , utilizar para determinar a distância entre dois pontos. Poderá ser útil para o cálculo do raio , por exemplo.

---

**A interpretação faz parte da avaliação!**