

# Introdução à Programação (IF-669)

## 1º Exercício Escolar

Adriano Sarmiento  
Centro de Informática-UFPE  
31 de outubro de 2008

### 1ª Questão

Explique o que é a técnica de desenvolvimento de software *em camadas*. Discuta as possíveis vantagens e desvantagens de se utilizar esta técnica. (2,0 p)

### 2ª Questão

Examine os programas abaixo escritos em Java e C respectivamente:

```
1. import br.ufpe.cin.miniJava.util.MinijavaSystem;
2.
3. public class ArrayOrdenado {
4.
5.     public static void main(String[] args) {
6.         int[] vetor = new int[10];
7.         for (int i = 0; i < 10; i++)
8.             vetor[i] = i;
9.         MiniJavaSystem console = new MiniJavaSystem();
10.        console.print("O vetor contem os elementos: ");
11.        for (int i = 0; i < 10; i++)
12.            console.print((vetor + i) + " ");
13.    }
14. }
```

```
1. #include <stdio.h>
2. #include <stdlib.h>
3.
4. int main() {
5.
6.     int* vetor;
7.     int i;
8.     vetor = (int*) malloc (10 * sizeof(int));
9.     for (i = 0; i < 10; i++)
10.        vetor[i] = i;
11.     printf("O vetor contem os elementos: ");
12.     for (i = 0; i < 10; i++)
13.        printf("%d ", *(vetor+i));
14.     return 0;
15. }
```

Descreva o que ocorrerá ao final da execução de cada main dos programas acima. Em caso de ocorrência de erro, identifique a linha do erro, o tipo de erro (de compilação ou execução) e explique o porquê do erro. (2,0p)

**Obs: Em caso de erro de compilação, o programa nem chega a executar**

### 3ª Questão

A classe Conjunto descrita abaixo representa um conjunto de inteiros.

```

public class Conjunto {
    private int[] inteiros;
    private int posicaoLivre;
    public Conjunto(int tamanho) {
        inteiros = new int[tamanho];
        posicaoLivre = 0;
    }
    /**
     * Retorna um array com os elementos do Conjunto
     */
    public int[] getArray() throws ConjuntoVazioException {
        if (posicaoLivre == 0){
            throw new ConjuntoVazioException();
        } else {
            int[] arrayConjunto = new int[posicaoLivre];
            for (int i = 0; i < posicaoLivre; i++) {
                arrayConjunto[i] = inteiros[i];
            }
            return arrayConjunto;
        }
    }
    ...
}

```

Complete esta classe com a definição (implementação) de dois métodos:

- `getArrayIntersecao` - Recebe como parâmetro outro objeto do tipo `Conjunto` e retorna um array de inteiros que representa a interseção de dois conjuntos. Se um conjunto vazio for passado como parâmetro, ou se não houver interseção entre os conjuntos, o método deve lançar a exceção `ConjuntoVazioException` (2,0p)
- `getArrayDiferenca` - Recebe como parâmetro outro objeto do tipo `Conjunto` e retorna um array de inteiros que representa a diferença do conjunto definido na classe em relação ao conjunto passado como parâmetro. Exemplos:  $\{1,2,3,4,5\} - \{1,3,5\} = \{2,4\}$ , e  $\{1,2\} - \emptyset = \{1,2\}$ . Se o conjunto resultante for vazio, o método deve lançar a exceção `ConjuntoVazioException`. (2,0p)

**Obs 1:** Lembre-se de que conjuntos não têm elementos repetidos.

**Obs 2:** Pode usar métodos existentes e definir métodos adicionais para a resolução

#### 4ª Questão

Considere o seguinte trecho de código escrito na linguagem C:

```

float nota1 = ...;
float nota2 = ...;
float nota3 = ...;
float media;
struct aluno aluno1;
salvaNotasECalculaMedia(...);

```

Completar a chamada da função com os argumentos

Defina (implemente) a estrutura `aluno` que deve armazenar o nome do aluno e três notas. Defina (implemente) a função `salvaNotasECalculaMedia`, e indique que argumentos ela deve receber acima, de forma que após a execução do trecho de código a variável `aluno1` do tipo `struct aluno` contenha as três notas, e a variável `media` contenha a média aritmética destas três notas. (2,0p)