



2a Verificação de Aprendizagem

1. (2,0) Analise o código Java abaixo:

```
public abstract class Forma {
    protected abstract double getArea(double tam);
    protected abstract double getPerimetro(double tam);
}

public class Quadrado extends Forma {
    public double getArea(double lado){
        return lado * lado;
    }
    public abstract double getPerimetro(double lado);

    public static void main(String[] args){
        Quadrado q = new Quadrado();
        System.out.print(q.getArea(4));
    }
}
```

Indique o que acontece com este programa. Caso exista algum problema, proponha duas formas de resolvê-lo.

2. (2,0) Analise o seguinte código e responda as perguntas a seguir.

```
void prin() {
    void fun1(int a) {
        a = a - 2;
        int x = a * 3;
        fun2(x);
        print(x);
    }
    void fun2(int b) {
        b = b * 2;
        x = b + x;
        print(x);
    }
    static int x = 9;
    if (x > 5)
        x = x - 2;
        prin();
    else
        fun1(++x);
}
```

Qual o resultado deste programa:

- a) caso a linguagem use escopo estático e passagem de parâmetro por valor
 - b) caso a linguagem use escopo dinâmico e passagem de parâmetro por referência
3. (2,0) Descreva duas decisões de projeto de linguagens orientadas a objeto detalhando as opções de cada uma delas e exemplificando com linguagens de programação que as escolheram.
4. (2,0) Um sistema que possui muito tráfego de dados gostaria de conhecer a quantidade de bytes que estão sendo escritos através da interface OutputStream abaixo. No entanto, eles não gostariam de alterar o código da aplicação que já está funcionando. Desta forma, crie um aspecto que guarda a quantidade de bytes que estão sendo escritos nessa aplicação.

```
interface OutputStream {
    public void write(byte b);
    public void write(byte[] b);
}
```

5. (2,0) Usando conceitos de concorrência vistos e exercitados ao longo da disciplina, descreva uma solução em Java ou em Go que simula a execução de diversas instruções de forma paralela em uma CPU considerando que a tal CPU pode ter **n** núcleos de processamento. **OBS:** não precisa colocar o código completo, mas ilustre algumas linhas da sua solução.

Boa Prova!