Trabalho Cap 5 e 6

Gabriel Visentin Alexandre 23000602-2

Cap 5:

1-

Python

```
X=10
def f():
        print (“f vê x =” , x)
def g():
        x = 20
        f()
g()
```

##

javaScript

```
let x = 10;
function f() {
  console.log("f vê x =", x);
}
function g() {
  let x = 20; // local a g()
  f();
}
g(); // saída: f vê x = 10
```

Resposta:

. O valor impresso depende do local de definição da função, não do local da chamada

. Tanto python quanto javaScript usam escopo estático nesse exemplo

2-

```
#include <stdio.h>

void contador(void) {
   int a = 0;       // automática: recriada a cada chamada
   static int b = 0; // estática: preserva valor entre chamadas
   a++;
   b++;
   printf("a=%d, b=%d\n", a, b);
}

int main(void) {
   contador(); // a=1, b=1
   contador(); // a=1, b=2
   contador(); // a=1, b=3
   return 0;
}
```

Resposta:

. `a` reinicia pois é automática, durante apenas a execução da função
. `b` acumula pois é estática, durante todo o programa, mesmo sendo local

Cap 6

3-

Java

```
public class Tipagem {
   public static void main(String[] args) {
      int num = 10;
      // num = "dez";
      System.out.println(num + 5);
   }
}
```

##

Python

```
num = 10
print(num + 5)  #
num = "dez"
```

Resposta:

. Em java o compilador não permite trocar int por string.
. Em python você pode reatribuir num para outra tipagem, mas o erro só aparece na hora de executar se a operação não fizer sentido.
. Vantagens: erros mais cedo (compilação), melhor otimização.

4-

```
#include <stdio.h>
#include <string.h>

struct Livro {
   char titulo[50];
   char autor[50];
   int anoPublicacao;
};

int main(void) {
   int valores[5] = {2, 4, 6, 8, 10};  // array homogêneo
   for (int i = 0; i < 5; i++) {
      printf("%d ", valores[i]);
   }
   printf("\n");
```

```
    struct Livro l1;
    strcpy(l1.titulo, "Clean Code");
    strcpy(l1.autor, "Robert C. Martin");
    l1.anoPublicacao = 2008;

    printf("Livro: %s (%d) - %s\n", l1.titulo, l1.anoPublicacao, l1.autor);
    return 0;
}
#####################################################################
```

Java

```
public class Livro {
    String titulo;
    String autor;
    int anoPublicacao;

    public Livro(String titulo, String autor, int anoPublicacao) {
        this.titulo = titulo;
        this.autor = autor;
        this.anoPublicacao = anoPublicacao;
    }
#################################################################
}

import java.util.ArrayList;

public class Livraria {
    public static void main(String[] args) {
        ArrayList<Livro> livros = new ArrayList<>();
        livros.add(new Livro("Código Limpo", "Robert C. Martin", 2008));
        livros.add(new Livro("Effective Java", "Joshua Bloch", 2018));
        livros.add(new Livro("Refactoring", "Martin Fowler", 2018));

        for (Livro l : livros) {
            System.out.println(l.titulo); // só os títulos
        }
    }
}
```

. Array: Coleção de mesmo tipo, geralmente tamanho fixo;
. Struct/Classe: agrupam vários campos de tipos diferentes para modelar um objeto do mundo real;
. Use array quando tem muitos valores do mesmo tipo; Use struct/classe quando precisa representar entidades com vários atributos