

Linguagens de Programação
Prof. Andrei Rimsa Álvares

Lista de Exercícios V

Verificação de Tipos

Exercício 01) Considere os seguintes trechos de códigos em Java. Verifique se é preciso fazer coerção explícita para algum tipo (erro de compilação) e em caso positivo indique como. Mostre o valor resultante da variável d em cada caso.

a) `byte a = 40;`
`byte b = 50;`
`byte c = 100;`
`int d = a * b / c;`

b) `byte b = 50;`
`byte d = b * 2;`

Exercício 02) Considere as seguintes atribuições em Java:

```
byte b = 42;  
char c = 'a';  
short s = 1024;  
int i = 50000;  
float f = 5.67f;  
double d = 0.1234;
```

A partir delas, derivou-se a seguinte expressão, dada abaixo. Mostre qual o tipo resultante em cada uma das etapas intermediárias dessa expressão:

$(f * b) + (i / c) - (d * s)$

- a) $(f * b)$
- b) (i / c)
- c) $(f * b) + (i / c)$
- d) $(d * s)$
- e) $(f * b) + (i / c) - (d * s)$

Exercício 03) Cada um dos programas a seguir executa um comando de atribuição, indicado pela linha sublinhada. Explique o que acontece em cada caso. As respostas possíveis são: (i) o comando de atribuição impede a compilação do programa; (ii) o programa compila, mas ocorre um erro em tempo de execução; (iii) o programa compila e pode ocorrer um erro em tempo de execução; (iv) o programa compila e executa corretamente. Justifique sua resposta citando alguns dentre esses pontos: (a) tipagem forte vs. tipagem fraca; (b) equivalência de estrutura vs. equivalência de nome.

Linguagens de Programação
Prof. Andrei Rimsa Álvares

a) em C++

```
struct point { int x, y; };
struct triple { int p1, p2, p3; };
int main() {
    struct point* p = new struct point;
    struct triple* t = new struct triple;
    t->p1 = t->p2 = t->p3 = 1;
    p = (struct point*) t;
    printf("%d, %d\n", p->x, p->y);
    return 0;
}
```

b) em Java

```
class Point { int x, y; }
class Triple extends Point { int p1, p2, p3; }
class Main {
    public static void main(String[] args) {
        Point p = new Point();
        Triple t = new Triple();
        t.p1 = t.p2 = t.p3 = 1;
        p = (Point) t;
        System.out.printf("%d, %d\n", p.x, p.y);
    }
}
```

c) em C

```
struct point { int x, y; };
struct pair { int x, y; };
void main() {
    struct point p;
    struct pair t = {1, 1};
    p = t;
    printf("%d, %d\n", p.x, p.y);
}
```