

Treinamento em C#

Matheus Neder

matheusneder@gmail.com

Apresentação

- Desenvolvedor Fullstack
- Engenheiro de Computação (em formação)
- Arquiteto de Software

Apresentação - assuntos de interesse

- Padrões de projeto;
- POO;
- Arquitetura de software;
- DDD;
- TDD;
- SOLID;
- DevOps ...

Apresentação - tecnologias UX / UI

- Web
- Java Swing, AWT, SWT e Android
- C GTK+
- C++ GTKmm, QT
- C# Xamarin.Forms
- Objective-C iOS

Apresentação - tecnologias em geral

- ASP
- VB6 (COM+)
- JavaScript
- Java
- C/C++
- Linux
- Bash

Visão geral

- Tipos de dados;
- Gerenciamento de memória;
- Passagem de valores;
- Estruturas de desvio condicional / repetição;
- Delegates e eventos;

- **POO:**

- Tipos genéricos;
- Encapsulamento;
- Polimorfismo;

Visão geral

- POO:
 - Tratamento de exceções (try ... catch);
 - Classes abstratas e interfaces;
 - **SOLID**;
 - IoC e DI;
 - **DDD**;

Visão geral

- Programação funcional:
 - LINQ;
 - Funções lambda;
- ORMs;
- Threads:
 - Tasks;
 - Programação assíncrona (async e await).

Tipos de dados

- Tipos primitivos (bool, char, int ...);
- Tipos complexos;
- Tipos de valor (value types);
- Tipos de referência (reference types).

Tipos primitivos (internos)

- bool
- byte
- sbyte
- char
- decimal
- double
- float
- int
- uint
- long
- ulong
- object
- short
- ushort
- string

Tipos complexos

Definição:

```
struct Coord
{
    public double x;
    public double y;
}
```

Utilização:

```
Coord coord = new Coord();
coord.x = 10;
coord.y = 20;
```

Tipos de dados

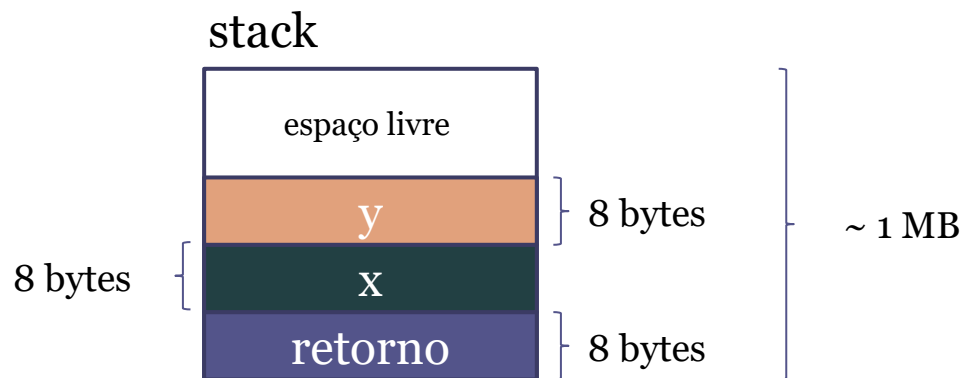
Tipos de valor

VS

Tipos de referência

Gerenciamento de memória

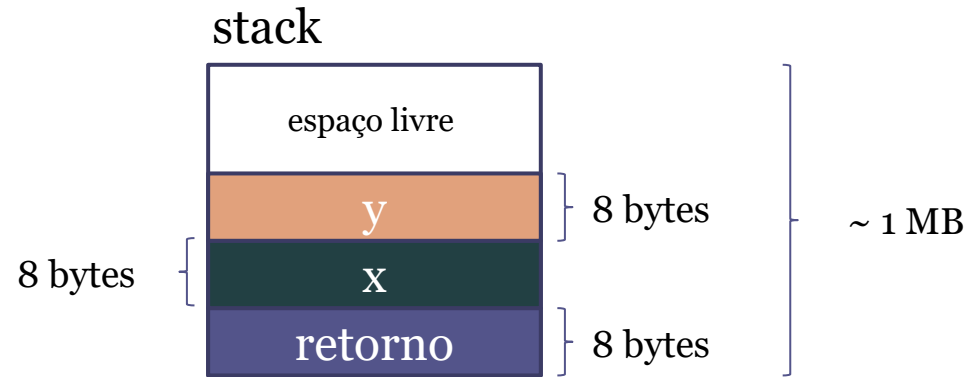
```
public long Soma(long x, long y)
{
    return x + y;
}
```



Observação: 8 bytes = sizeof(long)

Tipos de dados

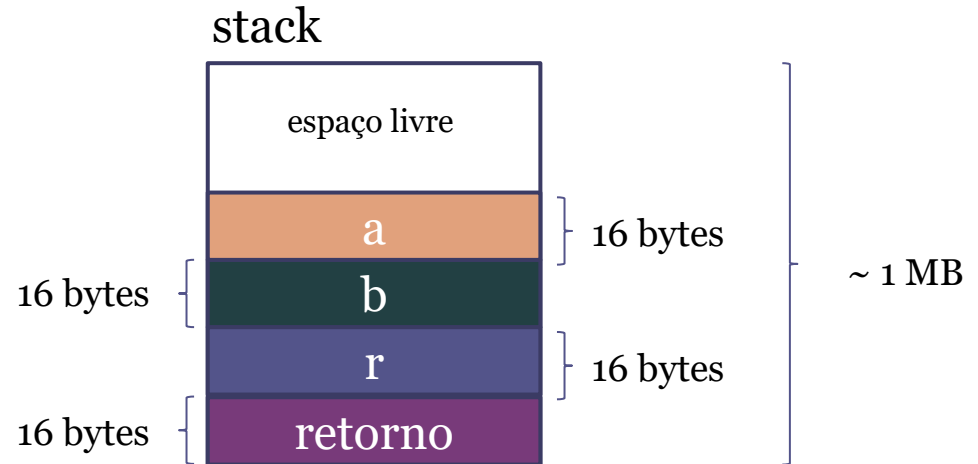
Tipos de valor (value types) são armazenados na pilha (stack).



Observação: 8 bytes = sizeof(long)

Tipos de dados

```
public Coord Midpoint(Coord a, Coord b)
{
    Coord r = new Coord();
    ...
    return r;
}
```



Observação: 16 bytes = sizeof(Coord)

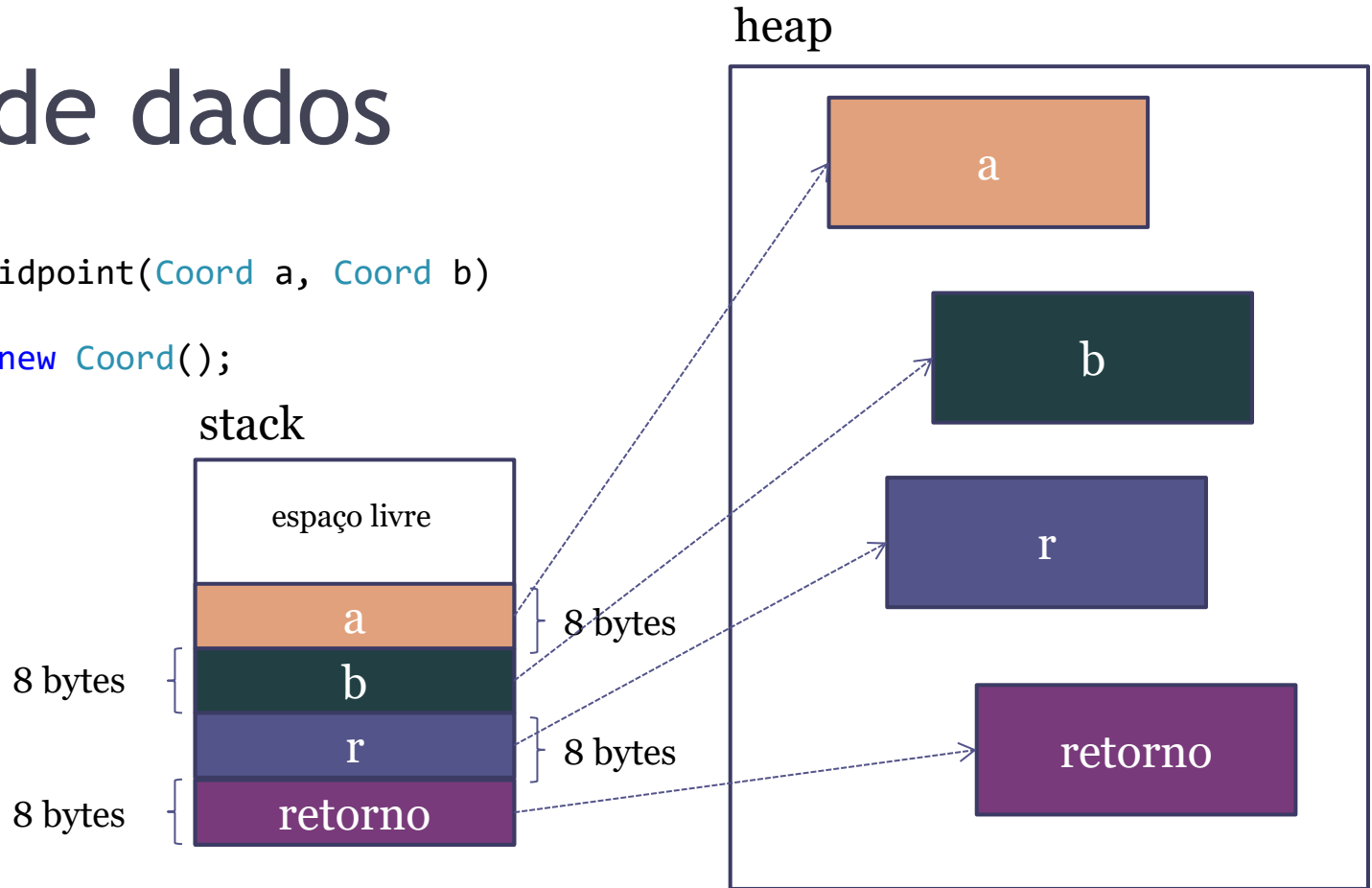
Tipos de dados

Considere uma nova versão do tipo *Coord*:

```
class Coord
{
    public double x;
    public double y;
}
```


Tipos de dados

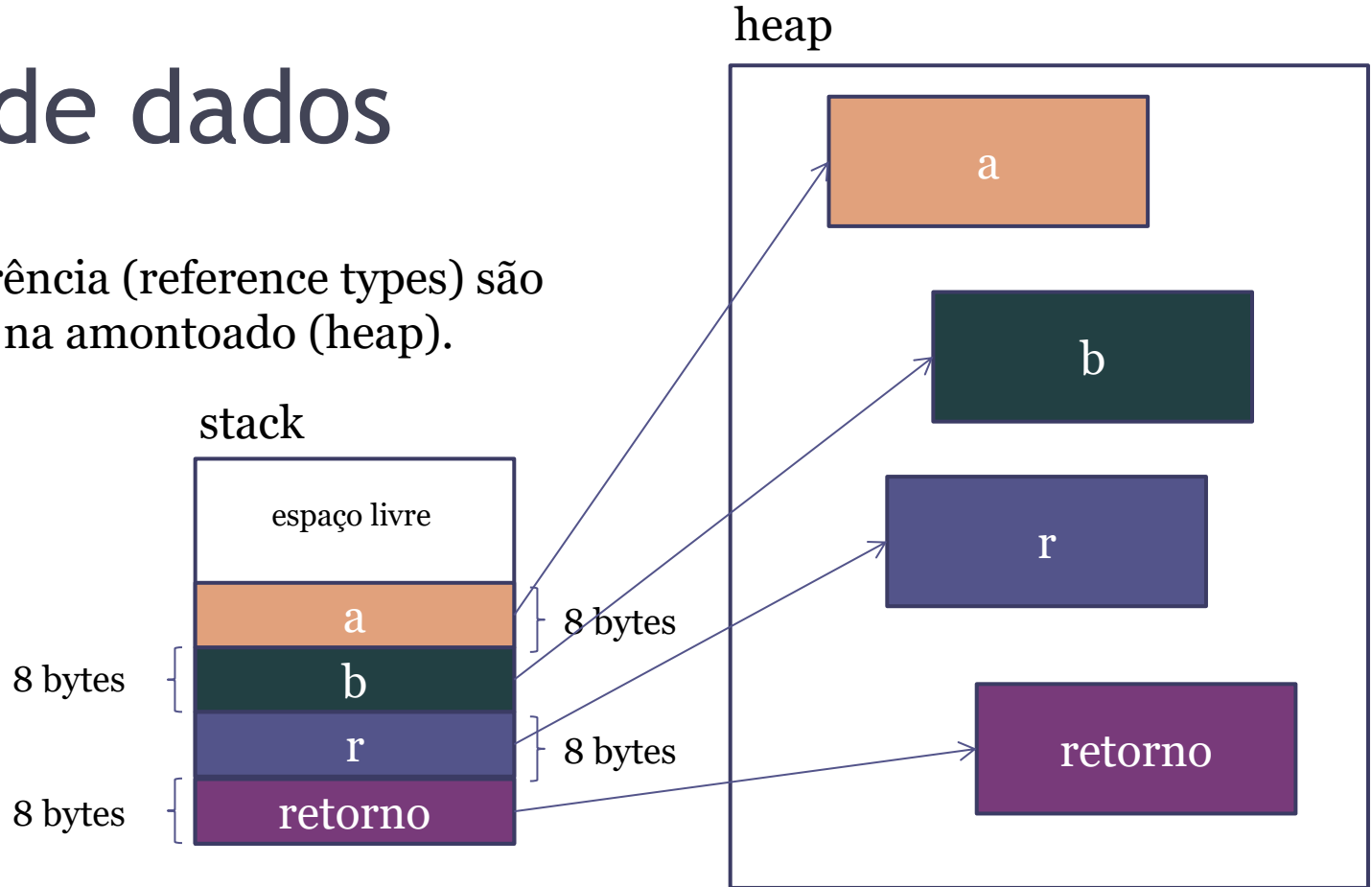
```
public Coord Midpoint(Coord a, Coord b)
{
    Coord r = new Coord();
    ...
    return r;
}
```



Observação: 8 bytes = tamanho do ponteiro

Tipos de dados

Tipos de referência (reference types) são armazenados na amontoado (heap).



Observação: 8 bytes = tamanho do ponteiro

Tipos de dados

```
struct Coord
{
    public double x;
    public double y;
}

public void Swap(Coord coord)
{
    double aux = coord.x;
    coord.x = coord.y;
    coord.y = aux;
}
```

```
Coord coord = new Coord();
coord.x = 10.0;
coord.y = 5.0;

Swap(coord);

Console.WriteLine($"x: {coord.x}");
Console.WriteLine($"y: {coord.y}");
```



Tipos de dados

```
class Coord
{
    public double x;
    public double y;
}

public void Swap(Coord coord)
{
    double aux = coord.x;
    coord.x = coord.y;
    coord.y = aux;
}
```

```
Coord coord = new Coord();
coord.x = 10.0;
coord.y = 5.0;

Swap(coord);

Console.WriteLine($"x: {coord.x}");
Console.WriteLine($"y: {coord.y}");
```



Tipos de dados

