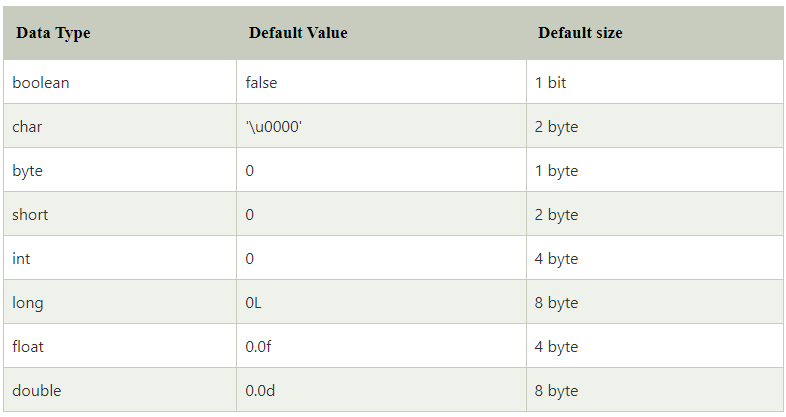
**Estrutura de Dados**



**Tipo de dado Boolean**

O tipo de dado boolean é usado para armazenar somente dois possíveis valores: verdadeiro e falso. Este tipo de dado é usado por sinalizadores simples que localizam condições verdadeira/falsa.

Especifica um bit de informação, mas o "tamanho" não pode ser especificado precisamente.

**Exemplo:** boolean one = false

### Tipo de dado byte

É um exemplo de um tipo de dado primitivo. É um inteiro de complemento de dois com assinatura de 8 bits. Sua faixa de valor está entre -128 até 127 (inclusive). Valor mínimo é -128 e o valor máximo é 127. Seu valor padrão é 0 (zero).

O tipo de dado byte é usado para salvar em memória grandes matrizes onde a memória salva é muito requisitada. Isso economiza espaço porque um byte é quatro vezes menor que um número inteiro. Também pode ser usado no lugar de tipos de dados "int".

**Exemplo:** byte a = 10, byte b = -20

### Tipo de dado short

O tipo de dado short é um inteiro de complemento de dois com assinatura de 16 bits. Sua faixa de valor está entre -32.768 até 32.767 (inclusive). Seu valor mínimo é -32.768 e valor máximo é 32.767. Seu valor padrão é 0 (zero).

O tipo de data short também pode ser usado para economizar memória assim como o tipo de dado byte. Um tipo de dado short é duas vezes menor que um inteiro.

**Exemplo:** short s = 10000, short r = -5000

### Tipo de dado int

O tipo de dado int é um inteiro de complemento de dois com assinatura de 32 bits. Sua faixa de valor está entre -2.147.483.648 (-2³¹) até 2.147.483.647 (2³¹?¹) (inclusive). Seu valor mínimo é -2.147.483.648 e valor máximo é 2.147.483.647. Seu valor padrão é 0 (zero).

O tipo de dado int geralmente é usado como um tipo de dado padrão para valores inteiros sem nenhum problema de memória.

**Exemplo:** int a = 100000, int b = -200000

### Tipo de dado long

O tipo de dado long é um inteiro de complemento de dois com assinatura de 64 bits. Sua faixa de valor está entre -9.223.372.036.854.775.808 (-2?³) até 9.223.372.036.854.775.807 (2?³?¹) (inclusive). Seu valor mínimo é -9.223.372.036.854.775.808 e o valor máximo é de 9.223.372.036.854.775.807. O valor padrão é 0 (zero). O tipo de dado long é usado quando você precisa de uma faixa de valores maior do que a fornecida por int.

**Exemplo:** long a = 100000L, long b = -200000L

### Tipo de dado float

O tipo de dado float é um ponto flutuante IEEE 754 de 32 bits de precisão única. Sua faixa de valor é ilimitada. É recomendado usar um ponto flutuante (em vez de double) se você precisar economizar memória em grandes arrays (matrizes) de número de ponto flutuante. O tipo de dado float nunca deve ser usado para valores precisos, como moeda. Seu valor padrão é 0.0f.

**Exemplo:** float f1 = 234.5f

### Tipo de dado double

O tipo de dado double é um ponto flutuante IEEE 754 de 64 bits de precisão dupla. Seu valor é ilimitado. O tipo de dado double é geralmente usado para valores decimais assim como float. O tipo de dado double também não deve ser usado para precisar valores, como moeda. Seu valor padrão é 0.0d.

**Exemplo:** double d1 = 12.3

### Tipo de dado char

O tipo de dado char é um único caractere unicode de 16 bits. Sua faixa de valor está entre '\u0000' (ou 0) até '\uffff' (ou 65.535 inclusive). O tipo de data char é usado para para armazenar caracteres.

**Exemplo:** char letraA = 'A'

**Atribuição e Referência**

Ao usarmos um tipo primitivo o Java usa como referência o valor em memória, por isso o valor de intB não é alterado quando o valor de intA passa a ser 5.

Ao usarmos um objeto o Java usa como referência o endereço em memória, por isso o valor de objB recebe o mesmo valor do objA após esse mesmo ter sido alterado.

**Conceito de Pilha**

É uma estrutura de dados do tipo LIFO (last-in first-out), onde o último elemento a ser inserido, será o primeiro a ser retirado. Assim, uma **pilha** permite acesso a apenas um item de dados - o último inserido. Para processar o penúltimo item inserido, deve-se remover o último”.

**Conceito de Fila**

É uma estrutura de dados do tipo FIFO: “o primeiro elemento que entra é o primeiro que sai” (first in, first out). A idéia fundamental da **fila** é que só podemos inserir um novo elemento no final da **fila** e só podemos retirar o elemento no início.