

Bài 5: CÁC KIỂU DỮ LIỆU TRONG JAVA

Xem bài học trên website để ủng hộ Kteam: [Các kiểu dữ liệu trong Java](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Dẫn nhập

Ở bài trước, Kteam đã giới thiệu cho các bạn về [BIẾN TRONG JAVA](#), trong đó có nói qua về kiểu dữ liệu. Trong bài viết này Kteam sẽ hướng dẫn các bạn chi tiết hơn về **Kiểu dữ liệu trong Java**.

Nội dung

Để đọc hiểu bài này, tốt nhất các bạn nên có kiến thức cơ bản về các phần sau:

- [CÁC BIẾN TRONG JAVA](#)

Bài này chúng ta sẽ tìm hiểu những vấn đề sau:

- Kiểu dữ liệu là gì? Lý do phải có kiểu dữ liệu.
- Phân loại các kiểu dữ liệu.

Kiểu dữ liệu là gì? Lý do phải có kiểu dữ liệu

Kiểu dữ liệu là tập hợp các nhóm dữ liệu có chung đặc tính, cách lưu trữ và thao tác xử lý. Nhờ có kiểu dữ liệu thì compiler nhận biết được kích thước của một biến và khả năng lưu trữ của nó.

Lý do phải có kiểu dữ liệu

Biến cần có bộ nhớ dành riêng để lưu trữ giá trị. Nghĩa nào khi bạn khai báo biến bạn cần dữ trữ không gian cho bộ nhớ.

Dựa vào kiểu dữ liệu của biến, hệ điều hành sẽ cấp bộ nhớ và quyết định thứ gì có thể lưu trữ ở bộ nhớ đã cho. Như biến có kiểu dữ liệu số nguyên, số thực hay kí tự thì hệ điều hành sẽ xác định vùng nhớ đó lưu giá trị như thế nào.

Phân loại các kiểu dữ liệu

Java có 2 loại kiểu dữ liệu:

- Các kiểu dữ liệu nguyên thủy (Primitive Data Types)
- Các kiểu dữ liệu tham chiếu (Reference Types)

Primitive Data Types

Có đến 8 kiểu dữ liệu **primitive** trong Java. Kiểu dữ liệu Primitive đã được đặt ra trước và tên nó nằm trong từ khóa của Java. Bây giờ ta sẽ xem những kiểu dữ liệu đó:

1. Kiểu Boolean

Kiểu **Boolean** chỉ sử dụng lưu trữ cho 2 giá trị: **true** và **false**. Mục đích kiểu Boolean thường được cho những **câu điều kiện rẽ nhánh**.

- **Ví dụ:**

```
Boolean isDone = false
```

2. Kiểu byte

Kiểu dữ liệu **Byte** dùng để lưu trữ kiểu số nguyên có kích cỡ bằng **1 byte** (8 bit). Giá trị có thể lưu được nằm trong khoảng từ **-128 (-2^7)** đến **127 (2^7-1)**.

- **Ví dụ:**

```
byte a = 100
```

3. Kiểu Short

Kiểu dữ liệu **Short** dùng để lưu trữ kiểu số nguyên có kích cỡ bằng **2 byte** (16 bit). Giá trị có thể lưu được nằm trong khoảng từ **-32,768 (-2^{15})** đến **32,767 ($2^{15}-1$)**.

- **Ví dụ:**

```
short a = 10000
```

4. Kiểu Int

Kiểu dữ liệu **Int** dùng để lưu trữ kiểu số nguyên có kích cỡ bằng **4 byte** (32 bit). Giá trị có thể lưu được nằm trong khoảng từ **-2,147,483,648 (-2^{31})** đến **2,147,483,647 ($2^{31}-1$)**.

- **Ví dụ:**

```
int a = 20000000
```

5. Kiểu Long

Kiểu dữ liệu **Long** dùng để lưu trữ kiểu số nguyên có kích cỡ bằng **8 byte**. Giá trị có thể lưu được nằm trong khoảng từ **-9,223,372,036,854,775,808 (-2^{63})** đến **9,223,372,036,854,775,807 ($2^{63}-1$)**. Giá trị gán cần có kí tự **'l'** phía sau.

- **Ví dụ:**

```
long a = 1001000l
```

6. Kiểu Float

Kiểu dữ liệu **Float** dùng để lưu trữ số thực có kích cỡ bằng **4 byte** (32 bit). Giá trị có thể lưu được nằm trong khoảng từ **-3.4028235 x 10³⁸** đến **3.4028235 x 10³⁸**. Giá trị gán cần có kí tự '**f**' phía sau.

- **Ví dụ:**

```
float a = 2.51f
```

7. Kiểu Double

Kiểu dữ liệu **Double** dùng để lưu trữ số thực có kích cỡ bằng **8 byte** (64 bit). Giá trị có thể lưu nằm trong khoảng từ **-1.7976931348623157 x 10³⁰⁸** đến **1.7976931348623157 x 10³⁰⁸**. Giá trị gán có thể **có hoặc không** kí tự '**d**' phía sau.

- **Ví dụ:**

```
double a = 2.52.d hoặc double a = 2.52
```

8. Kiểu Char

Kiểu dữ liệu **Char** dùng để lưu trữ kí tự có kích cỡ bằng **2 byte**. Bản chất Char lưu trữ code Unicode nhưng khi lại hiển thị ra 'kí tự' ứng với mã đó. Giá trị có thể lưu trữ nằm trong khoảng '**u0000**' đến '**uffff**'.

- **Ví dụ:** đây là bảng chứa ít kí tự Unicode

Table of UNICODE codes,
for **Czech, Hungarian, Polish, Scandinavian** and some other Central European Languages.
The hexadecimal digits hhh used in the `&#Xhhh;` code.

Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code
Ā	100	Đ	110	Ě	118	Ķ	136	Ň	143	Ó	d3	Š	15a
ā	101	đ	111	ě	119	ķ	137	ň	144	ó	f3	š	15b
Ā	102	Ď	10e	Ě	11a	Ĺ	139;	Ň	145	Œ	152	Š	160
ā	103	ď	10f	ě	11b	ĺ	13a	ņ	146	œ	153	š	161
Ā	104	Ě	112			Ĺ	13b	Ň	147	í	155	Ť	162
ā	105	ē	113	Ĝ	122	ĺ	13c	ņ	148	Ŕ	156	ť	163
Č	106	ě	115	ĝ	123	Ĺ	13d	Ŏ	14c	ŕ	157		ž
č	107	Ě	116	ĩ	12a	ĺ	13e	ø	14d	ř	158	ť	165
Č	10c	ě	117	ĩ	12b			ő	150	ř	159		z
č	10d			ĩ	12e	Ĺ	141	ó	151	š	15e		ž
				ĩ	12f	ĩ	142			š	15f		ž

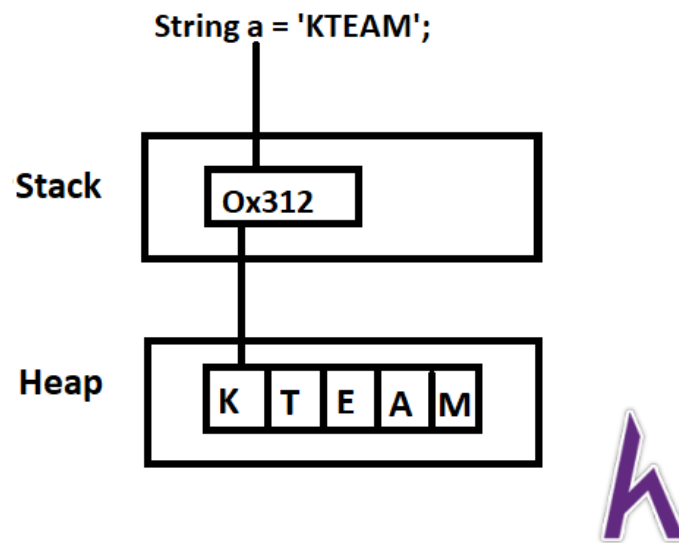
Example: `Ł=Ĺ` © 2002 B. C. Biega <http://biega.com>

Khi ta khai báo **char a = 'Ā'** thì bộ nhớ sẽ lưu mã Unicode là **'\u0102'**

Reference Types

Kiểu dữ liệu tham chiếu là kiểu dữ liệu của đối tượng. Biến của kiểu dữ liệu tham chiếu chỉ chứa địa chỉ của đối tượng dữ liệu tại bộ nhớ **Stack**. Đối tượng dữ liệu lại nằm ở bộ nhớ **Heap**. Một số kiểu dữ liệu cụ thể như các **mảng** (Array), **lớp đối tượng** (Class) hay kiểu **lớp giao tiếp** (Interface).

Như bài [BIẾN TRONG JAVA](#), Kteam đã có nói qua với kiểu String, bản chất String là một mảng lưu nhiều kí tự (char). Việc khai báo **String name = 'KTEAM'** có thể diễn đạt như sau:



Kiểu dữ liệu này Kteam sẽ giải thích cụ thể ở nhiều bài sau.

Kết luận

Như vậy chúng ta đã tìm hiểu các kiểu dữ liệu trong Java

Ở bài sau, Kteam sẽ giới thiệu đến bạn về [CÁC HANG TOÁN TỬ TRONG JAVA](#)

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên **"Luyện tập – Thử thách – Không ngại khó"**.