



TUHOCC.CC

LIVE ((●))

TỰ HỌC JAVA siêu tốc



LESSON

6

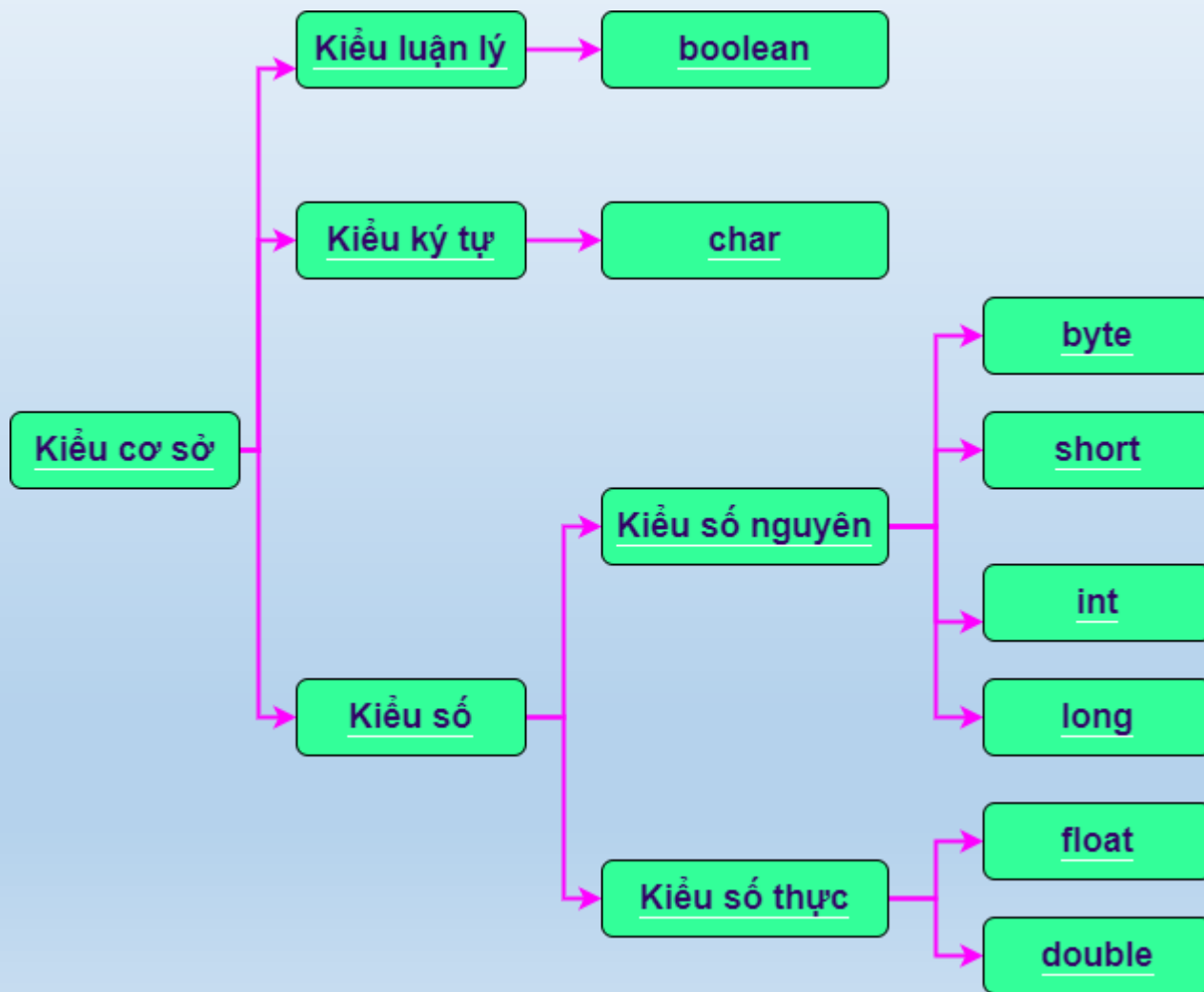
Ép kiểu dữ liệu java

double kq = (double) 2/4

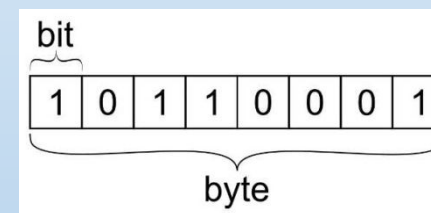


@galailaptrinh

1 Các kiểu dữ liệu cơ sở



Data Type	Size (bits)	Data Range
Byte	8 bit	-128 to 127
Short	16 bit	-32768 to 32767
Int	32 bit	-2,147,483,648 to 2,147,483,647
Long	64 bit	-9,223,372,036,854,775,808 to +9,223,372,036,854,775,807
Float	32 bit	1.40129846432481707e-45 to 3.40282346638528860e+38
Double	64 bit	4.94065645841246544e-324 to 1.79769313486231570e+308



2

Tại sao phải ép kiểu

- ✓ *Như bài học trước, chúng ta đã biết các kiểu dữ liệu sinh ra , để phù hợp với mục đích sử dụng, tiết kiệm bộ nhớ hệ thống.*
- ✓ *Trong quá trình tính toán, có thể kiểu dữ liệu trả về không còn giống với kiểu ban đầu chúng ta khai báo => Cần ép kiểu để sử dụng kiểu phù hợp*

```
int a =2;
int b =4;
double kq=a/b;
System.out.println(kq);
```

0.0

```
int a =2;
int b =4;
double kq=(double) a/b;
System.out.println(kq);
```

0.5

double kq = int/int → 2/4 = 0 (lấy phần nguyên)

→ Vì kq kiểu double nên giá trị trả về là **0.0**



3

Các loại ép kiểu

☐ Java Có 2 dạng ép kiểu



Ép kiểu rộng :

ép kiểu từ dữ liệu *bé* > *lớn*

Ex: *Int* → *Long* → *Float* → *Double*

Không lo mất dữ liệu



Ép kiểu hẹp :

ép kiểu từ dữ liệu *lớn* > *bé*

Ex: *double* → *float* → *long* → *int*

Có thể mất dữ liệu



4

Ép kiểu hẹp



Ép kiểu hẹp :

ép kiểu từ dữ liệu lớn > bé

Ex: double → float → long → int

Có thể mất dữ liệu

```
//ép kiểu hẹp
int c=128;
byte d =(byte) c;
//byte chỉ lưu trữ tối đa đc 127
System.out.println(c);
System.out.println(d);
```

128
-128

```
//ép kiểu hẹp
int e=15;
byte f =(byte) e;
//byte chỉ lưu trữ tối đa đc 127
System.out.println(e);
System.out.println(f);
```

15
15

