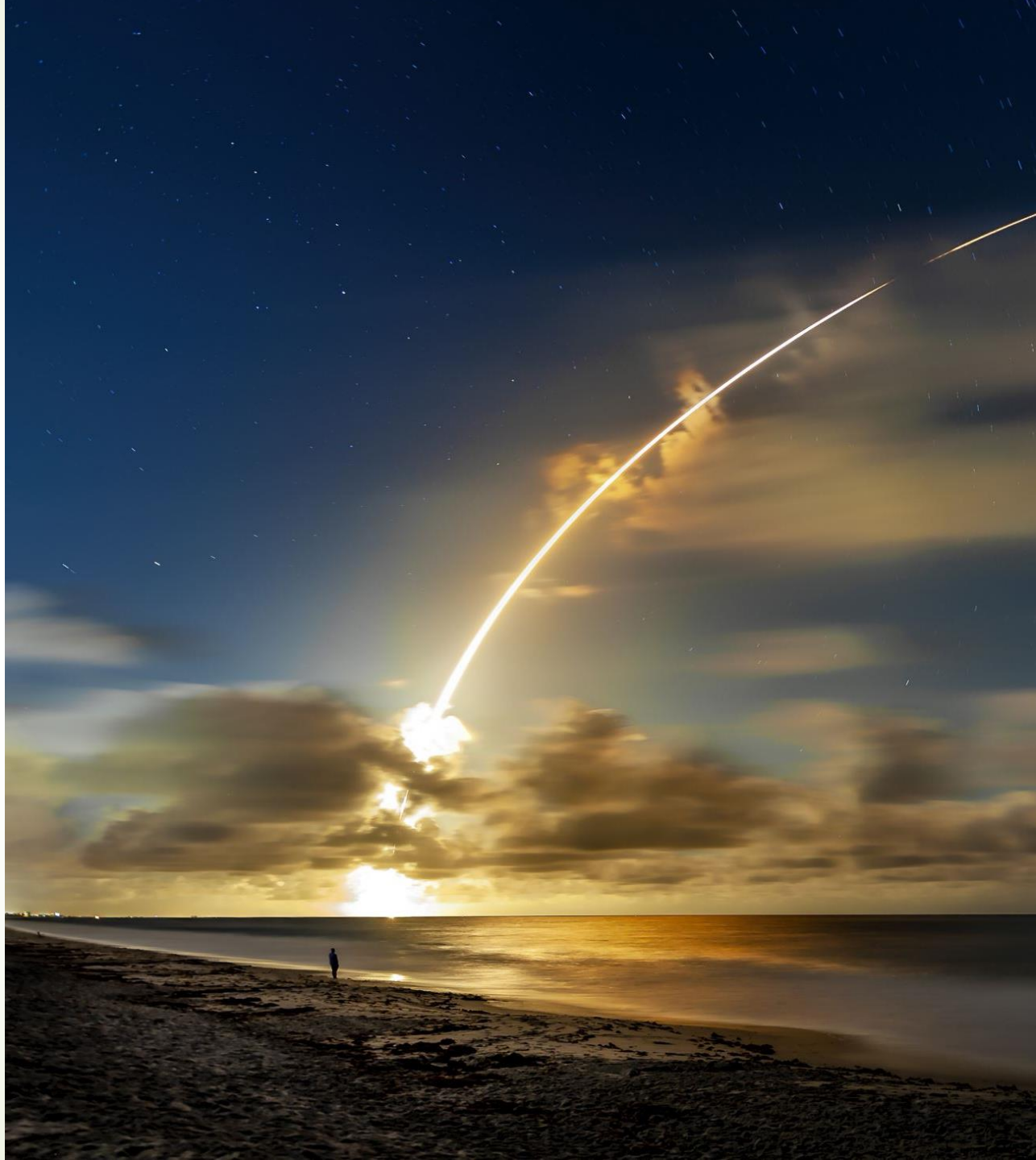
Bài 3: Các biến trong Java

- ✓ Khái niệm và đặc điểm
- ✓ Cú pháp khai báo biến
- ✓ Khởi tạo giá trị cho biến
- ✓ Gán giá trị cho biến
- ✓ Hằng số
- ✓ Các từ khóa
- ✓ Nhập dữ liệu cho biến
- ✓ Xuất dữ liệu ra màn hình



Khái niệm và đặc điểm

- Biến là tên đặt cho vùng nhớ dùng để lưu trữ dữ liệu tạm thời nhằm phục vụ hoạt động của chương trình.
- Biến sẽ bị hủy khi chương trình kết thúc
- Biến được cấp phát bộ nhớ trong RAM
- Biến được phân thành các nhóm chính: biến cục bộ, biến của lớp, hằng số.

Cú pháp khai báo biến

- Cú pháp tổng quát: *kiểu danh_sách_biến;*
- Trong đó:
  - Kiểu là các kiểu dữ liệu hợp lệ trong ngôn ngữ lập trình Java
  - Danh sách biến có thể gồm 1 hoặc nhiều biến, nếu nhiều biến thì phân tách nhau bằng dấu phẩy
  - Tên biến thường là danh từ hoặc cụm danh từ thể hiện ý nghĩa mà biến đang mô tả
  - Tên biến không được trùng các từ khóa và từ dành riêng trong Java
  - Tên biến luôn bắt đầu với chữ cái thường, sau đó có thể có chữ cái, chữ số, gạch dưới. Không bao giờ bắt đầu tên biến với kí tự khác chữ cái thường
  - Nếu tên biến có nhiều từ cấu thành thì các từ viết liền và viết hoa chữ cái đầu từ kể từ từ thứ hai trở đi

Cú pháp khai báo biến

- Cú pháp tổng quát: *kiểu danh_sách_biến;*
- Trong đó:
  - Tên biến trong Java phân biệt chữ hoa chữ thường, tức là name khác Name và khác nAMe
  - Luôn kết thúc câu lệnh khai báo biến bằng dấu chấm phẩy
  - Khuyến nghị chỉ khai báo một biến trên 1 dòng

Cú pháp khai báo biến

- Ví dụ về khai báo biến:

```
int age; // tuổi
float avgGrade; // điểm TB
double interestRate; // lãi suất
long ballance; // số dư
String fullName; // họ tên đầy đủ
String address; // địa chỉ
// sau đây là các biến không hợp lệ:
int 5year; // tên bắt đầu bởi số
float my bonus; // trong tên có dấu cách
double max-volume; // trong tên có dấu -
```

Cú pháp khởi tạo biến

- Mọi biến trong Java phải khai báo trước khi nó được sử dụng
- Trong quá trình khai báo biến mà ta gán luôn giá trị cho biến gọi là khởi tạo biến
- Cú pháp tổng quát: *kiểu tên_biến* = *giá trị;*
- Trong đó:
  - Kiểu là kiểu dữ liệu hợp lệ trong Java
  - Tên biến đặt theo quy ước đặt tên biến
  - Dấu = gọi là phép gán
  - Giá trị là các giá trị hợp lệ cùng kiểu với kiểu của biến
  - Cú pháp khởi tạo giá trị cho biến luôn đi kèm kiểu của biến ở đầu

Cú pháp khởi tạo biến

- Ví dụ minh họa:

```
int age = 0; // khai báo và khởi tạo tuổi bằng 0
float avgGrade = 0.0f; // khai báo và khởi tạo điểm TB = 0.0
String fullName = ""; // khai báo và khởi tạo họ tên rỗng
```

Gán giá trị cho biến

- Khi đã có biến mà ta cung cấp giá trị cho biến qua phép gán gọi là gán giá trị cho biến
- Cú pháp tổng quát: tên_biến = giá_trị;
- Trong đó:
  - Tên biến phải là các biến đã có trước đó
  - Dấu = gọi là toán tử gán. Đây là toán tử hai ngôi
  - Giá trị được gán cho biến là giá trị cùng kiểu với kiểu của biến
- Ví dụ:

```
int age = 0; // khai báo và khởi tạo tuổi bằng 0
float avgGrade = 0.0f; // khai báo và khởi tạo điểm TB = 0.0
double interestRate; // lãi suất
age = 20; // gán giá trị 20 cho biến age
avgGrade = 3.25f; // gán giá trị 3.25 cho điểm TB
interestRate = 5.25; // gán giá trị cho lãi suất
```

Hằng số

- Hằng số là các biến mà giá trị của nó đã được biết trước, không có nhu cầu thay đổi và không thể thay đổi kể từ sau khi gán giá trị
- Cú pháp tổng quát: final kiểu TÊN_HẰNG_SỐ = giá_trị;
- Trong đó:
  - Final là từ khóa bắt buộc
  - Kiểu là bất kì kiểu dữ liệu hợp lệ nào trong Java
  - Tên hằng số viết hoa hoàn toàn mang ý nghĩa nào đó. Nếu tên có nhiều từ thì các từ phân tách nhau bởi dấu _
  - Mọi hằng số không bắt buộc gán giá trị ngay khi khai báo

Hằng số

- Ví dụ minh họa:

```
final double PI = 3.1415926535897793; // số PI
final double E = 2.718281828459; // số E
final int MAX_VALUE;
MAX_VALUE = 100; // gán giá trị 100 cho hằng số MAX_VALUE
MAX_VALUE = 500; // error! giá trị của hằng số không thể thay
đổi
final String MONDAY = "MONDAY"; // ok
MONDAY = "FRIDAY"; // error!
```

Các từ khóa và từ dành riêng

- Từ khóa và từ dành riêng trong Java là các từ đã được gán sẵn 1 chức năng cụ thể nào đó và chỉ sử dụng với mục đích của chức năng đã được gán đó
- Các từ khóa và từ dành riêng luôn viết thường hoàn toàn
- Ví dụ về các từ khóa, từ dành riêng:

abstract	continue	for	new	switch	assert
if	package	synchronized	boolean	do	goto
this	break	double	implements	protected	throw
else	import	public	throws	case	enum
return	transient	catch	extends	int	short
char	final	interface	static	void	class
long	strictfp	volatile	const	float	native
while	default	private	byte	instanceof	super
try	finally	_	const	goto	

Các từ khóa và từ dành riêng

- Dấu gạch dưới _ là một từ dành riêng để sử dụng trong tương lai
- Giá trị *true*/*false* không phải từ khóa nhưng nó là giá trị riêng của kiểu Boolean nên cũng không được phép sử dụng làm tên các thành phần khác
- Giá trị *null* cũng không được dùng làm tên biến
- Từ khóa *var*, *yiel* cũng không được sử dụng làm tên biến
- Các từ khóa bị giới hạn khác: *open*, *requires*, *transitive*, *exports*, *opens*, *to*, *uses*, *provides*, *with*

Nhập dữ liệu vào cho các biến

- Để nhập dữ liệu vào cho các biến từ bàn phím, ta sử dụng đối tượng của lớp Scanner: Scanner input = new Scanner(System.in);
- *System.in* là chuẩn dữ liệu đầu vào từ bàn phím
- Để đọc dữ liệu vào cho các kiểu dữ liệu khác nhau ta dùng phương thức nextX. Với X là tên của kiểu biến cần lấy giá trị
- Ta thường đưa ra lời nhắc cho người dùng nhập đúng dữ liệu của kiểu cần nhập
- Khi đọc vào 1 từ hoặc 1 số, việc đọc sẽ dừng khi gặp khoảng trắng(dấu cách, tab, xuống dòng)

Nhập dữ liệu vào cho các biến

- Cú pháp nhập liệu cụ thể cho từng kiểu:

Kiểu biến	Phương thức tương ứng	Ví dụ
char	`next().charAt(0)`	char c = input.next().charAt(0);
byte	`nextByte()`	byte b = input.nextByte();
short	`nextShort()`	short s = input.nextShort();
int	nextInt()	int age = input.nextInt();
long	nextLong()	long longNumber = input.nextLong();
float	nextFloat()	float avgGrade = input.nextFloat();
double	nextDouble()	double interestRate = input.nextDouble();
boolean	nextBoolean()	boolean isPassed = input.nextBoolean();
String	Đọc 1 từ: next() Đọc cả dòng: nextLine()	String name = input.next(); String fullName = input.nextLine();

Xuất dữ liệu ra màn hình

- Để hiển thị dữ liệu ra màn hình ta có thể dùng phương thức print, printf, println của System.out
- *System.out* là chuẩn đầu ra dữ liệu mặc định của kiểu Java
- Cú pháp và mô tả:

Phím tắt	Phương thức	Ý nghĩa sử dụng
sout+Tab	`System.out.println();`	In ra nội dung cho trước sau đó đưa con trỏ chuột xuống đầu dòng kế tiếp
Soutf+Tab	`System.out.printf();`	In ra nội dung cho trước theo định dạng cụ thể tương ứng giống như trong ngôn ngữ C
	`System.out.print();`	In ra nội dung cho trước và sau đó con trỏ chuột ở nguyên vị trí kết thúc không xuống dòng mới

Xuất dữ liệu ra màn hình

- Một số định dạng chuyển đổi của các kiểu dữ liệu nguyên thủy

Kiểu	Kí tự chuyển đổi định dạng
Char	%c
String	%s
Int, byte, short, long	%d
Float, double	%f

Nội dung tiếp theo

Các toán tử