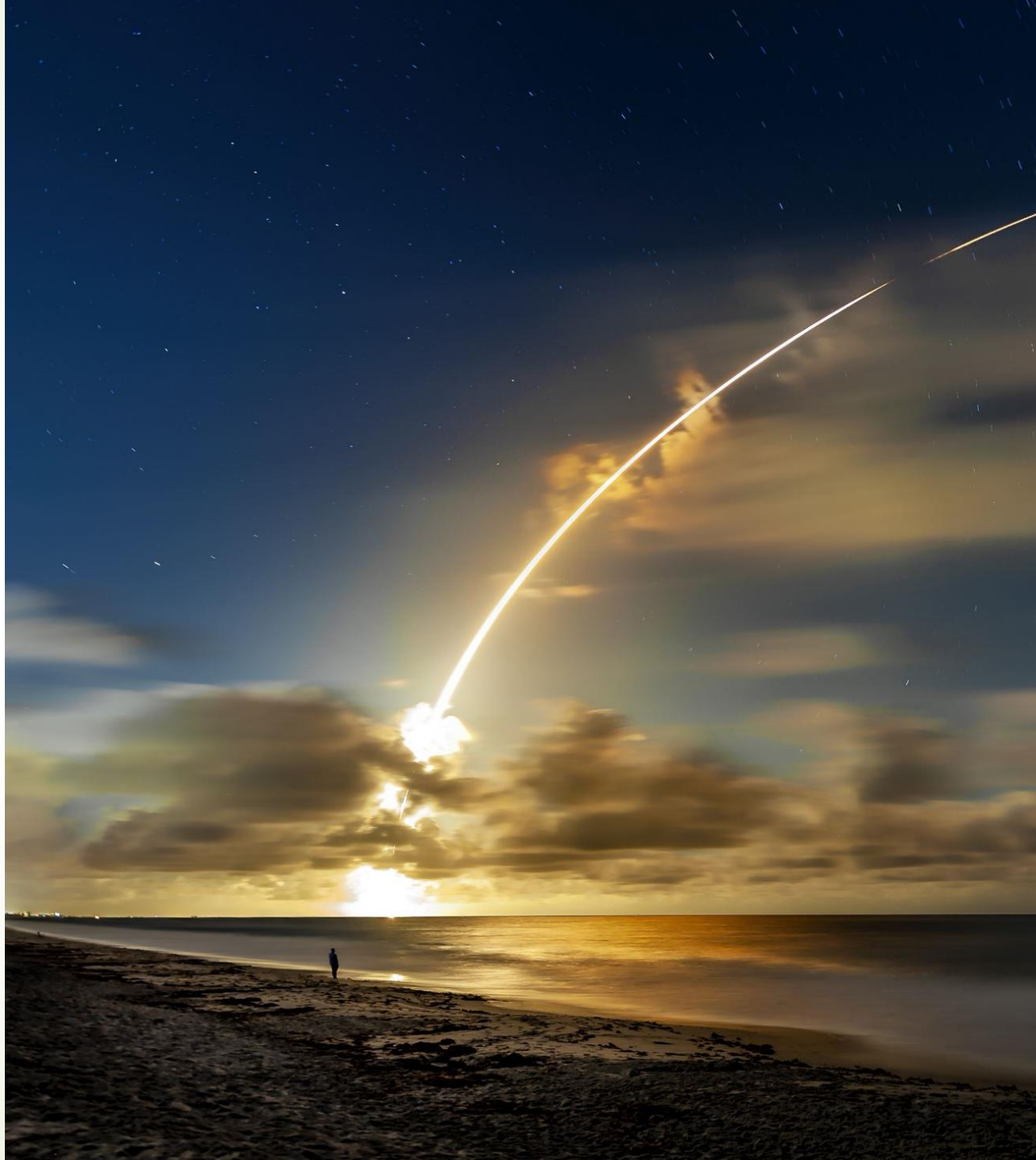
Bài 6: Lớp String trong Java

- ✓ Bản chất của String
- ✓ Phép nối chuỗi
- ✓ Các kí tự đặc biệt
- ✓ Phép cộng String với số
- ✓ Các phương thức của lớp String
- ✓ Nhập dữ liệu vào từ bàn phím
- ✓ Bài tập thực hành



Bản chất của String

- String dùng để lưu trữ chuỗi các kí tự trong ngôn ngữ lập trình Java.
- Chuỗi kí tự là tập các kí tự đặt trong cặp nháy kép “”.
- String là kiểu bất biến: giá trị không thể thay đổi sau khi gán.
- String là kiểu tham chiếu nên biến của String là các biến đối tượng không phải biến kiểu nguyên thủy.
- Ví dụ:
  - String message = “I Love You”;
  - Biến message là biến đối tượng có kiểu là String. Giá trị của message, tức “I Love You” không thể thay đổi. Ví dụ chữ L không thể thay đổi thành l hay các kí tự khác.

Phép nối chuỗi

- Để nối chuỗi String ta có thể dùng toán tử +.
- Chuỗi được nối sẽ dính liền vào cuối chuỗi trước đó.
- Chuỗi kết quả là một đối tượng String khác chứa chuỗi gốc và chuỗi cần nối.
- Ví dụ:

```
// Learn String
public class Lesson6 {
    public static void main(String[] args) {
        String first = "Minh";
        String middle = "Van";
        String last = "Tran";
        // nối chuỗi với dấu +
        String fullName = last + middle + first;
        System.out.println("Hello \"" + fullName + "\"");
    }
}
```

Các kí tự đặc biệt

- Các kí tự đặc biệt là các kí tự được gán sẵn cho 1 chức năng cụ thể nào đó. Khi sử dụng các kí tự này ta sẽ nhận được các chức năng đã gán sẵn.
- Sau đây là 1 số các kí tự đặc biệt:

Kí tự đặc biệt	Mô tả chức năng
\'	Đại diện cho kí tự '
\"	Đại diện cho kí tự "
\\	Đại diện cho kí tự \
\n	Xuống dòng mới
\t	Tạo một tab ngang thường tương đương với 4 kí tự

```
// sử dụng \n
System.out.println("I\nLove\nYou\nSo Much");
```

Kết quả:

```
I
Love
You
So Much
```

Cộng String với các giá trị số

- Mặc định chương trình Java coi rằng nếu có 1 toán hạng của phép + là String thì đó là phép nối chuỗi.
- Khi cộng số với String thì kết quả là String. Cụ thể xét ví dụ sau:

```
int numberA = 200;
int numberB = 500;
String result = "numberA + numberB = " + numberA + numberB;
// Mong muốn: numberA + numberB = 700
System.out.println(result);
// Thực tế: numberA + numberB = 200500
```

- Nếu muốn kết quả chính xác phải gộp nhóm với ngoặc ( ):

```
int numberA = 200;
int numberB = 500;
String result = "numberA + numberB = " + (numberA + numberB);
// Mong muốn: numberA + numberB = 700
System.out.println(result);
// Thực tế: numberA + numberB = 700
```

Các phương thức của lớp String

- Vị trí của kí tự đầu tiên trong String của Java luôn bắt đầu từ 0. Do đó phần tử đầu tiên trong chuỗi kí tự có chỉ số là 0. Phần tử cuối cùng có chỉ số n-1 với n là độ dài chuỗi.

Phương thức	Ý nghĩa sử dụng
length()	Cho biết độ dài của chuỗi tính theo số kí tự. Luôn >= 0
toUpperCase()	Viết hoa toàn bộ chuỗi
toLowerCase()	Viết thường toàn bộ chuỗi
indexOf()	Tìm vị trí đầu tiên của một kí tự/một chuỗi con trong chuỗi kí tự
lastIndexOf()	Tìm vị trí cuối cùng của một kí tự/chuỗi con trong chuỗi gốc
`charAt(index)`	Lấy ra kí tự tại vị trí index. Giá trị index phải >= 0 và < length() của chuỗi đang xét
compareTo(other)	So sánh sự tương đương có phân biệt chữ hoa, chữ thường về mặt giá trị của hai chuỗi. Kết quả trả về giá trị số âm, 0 hoặc dương tùy vào mức độ tương quan của hai chuỗi.
compareToIgnoreCase(other)	So sánh hai chuỗi về mặt giá trị. Không phân biệt chữ hoa chữ thường. Kết quả trả về giá trị âm nếu chuỗi hiện thời đứng trước other. Bằng 0 nếu hai chuỗi cùng giá trị. Dương nếu chuỗi hiện thời đứng sau chuỗi other.

Các phương thức của lớp String

trim()	Loại bỏ kí tự khoảng trắng ở đầu và cuối chuỗi. Khoảng trắng là các kí tự dấu cách, dấu tab, xuống dòng.
replace(old, newStr)	Thay thế kí chuỗi kí tự old bằng kí tự/chuỗi kí tự newStr. Thực hiện thay thế toàn bộ chuỗi old bằng newStr xuất hiện trong chuỗi.
replaceAll(regex, newStr)	Thay thế tất cả các chuỗi con thỏa mãn regex trong chuỗi gốc bằng newStr.
substring(startIndex)	Trích xuất chuỗi con của chuỗi gốc từ vị trí startIndex đến hết chuỗi gốc.
substring(start, end)	Trích xuất chuỗi con của chuỗi gốc từ vị trí start đến trước vị trí end.
isEmpty()	Kiểm tra xem chuỗi hiện thời có rỗng hay không. Kết quả nhận được là true nếu độ dài chuỗi bằng 0 và ngược lại kết quả là false.
isBlank()	Kiểm tra xem chuỗi hiện thời có rỗng hoặc chỉ chứa các kí tự khoảng trắng hay không. Kí tự khoảng trắng là các kí tự như dấu cách, dấu tab, xuống dòng. Như vậy một chuỗi là blank chưa chắc empty.

Các phương thức của lớp String

```
String first = "Minh";
System.out.println("Sau khi viết hoa: " + first.toUpperCase());
System.out.println("Sau khi viết thường: " +
first.toLowerCase());
System.out.println("Độ dài chuỗi: " + first.length());
```

Kết quả:

```
Sau khi viết hoa: MINH
Sau khi viết thường: minh
Độ dài chuỗi: 4
```

```
String nam = "Nam";
String other = "nam";
System.out.println(nam.compareTo(other));
System.out.println(nam.compareToIgnoreCase(other));
```

Kết quả:

```
-32
0
```

Nhập dữ liệu từ bàn phím

- Để nhập dữ liệu từ bàn phím vào cho String ta có hai cách:
  - Dùng phương thức next(): đọc vào một từ. Việc đọc vào sẽ dừng khi gặp khoảng trắng hoặc khi gặp dấu hiệu kết thúc dòng(ấn enter chẳng hạn).
  - Dùng phương thức nextLine(): đọc vào cả dòng String. Khi đọc vào cả dòng mà thực hiện sau đọc vào 1 số hoặc 1 từ, 1 giá trị thì luôn chú ý đọc bỏ kí tự thừa trước.
- Ví dụ đọc vào cả dòng:

```
Scanner input = new Scanner(System.in);
String name;
System.out.println("Hey, what your name?");
name = input.nextLine();
System.out.println("Hello " + name + "!");
```

- Khi hiển thị dữ liệu trong String với printf() thì dùng định dạng chuyển đổi %s.

Nội dung tiếp theo

Kiểu tự suy luận trong Java