

# **BÀI 4**

## **MẢNG VÀ CHUỖI TRONG JAVA**

# TÌNH HUỐNG DẪN NHẬP

## Bài toán: Xây dựng hệ thống quản lý khu nhà sinh viên

Một trường học muốn xây dựng hệ thống tin học để quản lý khu nhà ở dành cho sinh viên thuê.

- Qua khảo sát thực tế, ta thấy rằng các phòng thuê được xây dựng với mô hình giống nhau, được đánh số thứ tự các phòng liên tiếp nhau.
- Qua phân tích ta nhận thấy một trong những đối tượng cần quản lý ở đây là Phòng Thuê. Giả sử có 10 phòng thuê, ta phải ghi nhận thông tin của cả 10 phòng trong 10 đối tượng. Việc ghi nhận các thông tin giống nhau được thực hiện làm nhiều lần, làm mất thời gian và đôi khi việc tìm kiếm thông tin trước đó có thể gặp khó khăn.



Vậy theo anh/chị trong hệ thống máy tính làm thế nào để có thể quản lý được thông tin các phòng thuê với mô hình giống nhau mà không cần phải đưa ra 10 đối tượng phòng thuê?

# MỤC TIÊU BÀI HỌC

Trình bày được khái niệm và cách sử dụng mảng một chiều, mảng nhiều chiều.



Xây dựng chương trình Java sử dụng mảng và một số lớp chuỗi hỗ trợ sẵn trong Java.



# NỘI DUNG

- 1 Khái niệm mảng trong Java
- 2 Mảng một chiều
- 3 Mảng nhiều chiều
- 4 Khái niệm chuỗi

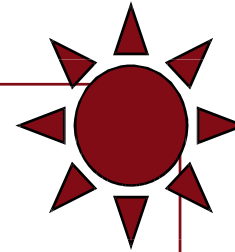
## 4.1. KHÁI NIỆM VỀ MẢNG

- Mảng là một cách lưu trữ, tổ chức dữ liệu cho phép lưu một danh sách các biến có cùng kiểu dữ liệu và được sắp xếp ở các vị trí liên tiếp nhau trong bộ nhớ.
- Lợi ích của mảng:
  - Mảng được sử dụng tối ưu trong trường hợp cần truy xuất dữ liệu vào các phần tử có cùng kiểu dữ liệu một cách ngẫu nhiên hoặc đồng thời.
  - Mảng tận dụng tối ưu tài nguyên bộ nhớ so với các biến.
  - Bộ nhớ chỉ được cấp phát cho mảng tại thời điểm mảng được sử dụng.

Vậy qua bài tập tình huống, để có thể quản lý các phòng thuê ta sử dụng mảng.

## 4.2. MẢNG MỘT CHIỀU

1. Khai báo mảng một chiều.
2. Khởi tạo mảng một chiều.
3. Khởi tạo mảng đối tượng.
4. Ví dụ về sử dụng mảng.



## 4.2.1. KHAI BÁO MẢNG MỘT CHIỀU

**Cú pháp khai báo mảng một chiều:**

```
<type> arrayName[];
```

```
<type> [] arrayName;
```

**Khai báo kích thước lưu trữ cho mảng:**

```
<type>[] arrayName;  
arrayName = new <type>[size];
```

**Ví dụ:**

```
int[] array;  
array = new int[5];
```

```
Room[] room;  
room = new Room[11];
```

## 4.2.1. KHAI BÁO MẢNG MỘT CHIỀU (tiếp theo)

Truy xuất vào phần tử trong mảng:

```
arrayName[index];
```

Ví dụ truy xuất và gán giá trị cho các phần tử của mảng:

```
int[] array;  
array=new int[5];  
array[0]=7;
```

Mỗi phần tử của mảng là một số nguyên. Kích thước (số phần tử) của mảng là 5 phần tử.

```
Room[] room;  
room=new Room[11];  
room[0]=new Room();  
room[0].info();
```

Mỗi phần tử của mảng là một đối tượng room. Kích thước (số phần tử) của mảng là 11 phần tử.



## 4.2.2. KHỞI TẠO MẢNG MỘT CHIỀU

Trước khi sử dụng các phần tử trong mảng cần được khởi tạo giá trị.

- Khởi tạo ngay khi khai báo mảng:

```
int[] array = {1, 2, 5, 2 5};
```

- Khởi tạo ngay khi khai báo cấp phát mảng:

```
int[] array = new int[] {3, 2, 5, 7, 2};
```

- Dùng vòng lặp for:

```
int[] array = new int[5];  
for (int i = 0; i < 5; i++) {  
    array[i] = 0;  
}
```

### 4.2.3. KHỞI TẠO MẢNG ĐỐI TƯỢNG

Sử dụng hàm tạo để khởi tạo từng phần tử trong mảng

```
Room[] room;  
room = new Room[11];  
for(int i=0; i<room.length; i++) {  
    room[i] = new Room();  
}
```

## CÂU HỎI THẢO LUẬN

Ưu điểm của việc sử dụng mảng N phần tử so với khai báo N biến?

## CÂU HỎI TRẮC NGHIỆM

Câu hỏi 1 trên 2 ▾

Điểm: 10

Lệnh sau in ra cái gì?

1: final int ARRAY\_SIZE = 5;

2: ARRAY\_SIZE = 10;

3: System.out.println("size = " + ARRAY\_SIZE);

- ☐ A. Size = 10
- ☐ B. Size = 5
- ☐ C. Lỗi ở dòng 3
- ☐ D. Lỗi ở dòng 2

### PROPERTIES

On passing, 'Finish' button:

On failing, 'Finish' button:

Allow user to leave quiz:

User may view slides after quiz:

User may attempt quiz:

Goes to Next Slide

Goes to Next Slide

At any time

At any time

Unlimited times



Properties...



Edit in Quizmaker

## VÍ DỤ 1

### **Kích thước mảng (Số phần tử trong mảng)**

- Có thể được xác định với biến hằng (`final`).

```
final int size = 10;
```

- Hằng là giá trị không được thay đổi.
- Hằng phải được khởi tạo tại thời điểm khai báo.

## VÍ DỤ 1 (tiếp theo)

```
/**
 * InitialArray.java
 */
public class InitialArray {
    public static void main(String[] args) {
        //Khai bao mang array voi gia tri khoi tao luc dau
        int[] array = {12, 34, 15, 34, 56, 36};
        System.out.println("Element  Value");

        //Dung vong lap duyet va in cac phan tu trong mang
        for (int i = 0; i < array.length; i++) {
            System.out.println(i + " \t" + array[i]);
        }
    }
}
```

**Run:**

<b>Element</b>	<b>Value</b>
0	12
1	34
2	15
3	34
4	56
5	36

Sử dụng thuộc tính length của mảng để duyệt

## VÍ DỤ 2

```
/**
 * Room.java
 */
public class Room{
    private String roomId;
    private int numberOfStudent;
    public Room(){}

    public Room(String _roomId, int _n){
        this.roomId = _roomId;
        this.numberOfStudent = _n;
    }
    public String toString() {
        return this.roomId + "\t" + numberOfStudent;
    }
}
/**
 * TestArray.Java
 */
```

## VÍ DỤ 2 (tiếp theo)

```
public class TestArray {  
    public static void main(String[] args) {  
        final int size = 5;  
        Room[] room = new Room[5]; //Khai bao mang doi tuong Room  
        //dung vong lap de khoi tao cac phan tu trong mang  
        for(int i = 0; i < size; i++){  
            room[i] = new Room("no room", 0);  
        }  
        System.out.println("STT   RoomId   NumberOfStudent");  
        //Dung vong lap duyet va in cac phan tu trong mang  
        for (int i = 0; i < room.length; i++) {  
            System.out.println(i + "\t" + room[i]);  
        }  
    }  
}
```

Sử dụng hằng size để duyệt mảng

Sử dụng thuộc tính length của mảng để duyệt



## VÍ DỤ 2 (tiếp theo)

Run:

STT	RoomId	NumberOfStudent
0	no room	0
1	no room	0
2	no room	0
3	no room	0
4	no room	0
5	no room	0

## BÀI TẬP

Viết một lớp IntegerManager với những đặc điểm được mô tả như sau:

- IntegerManager có khả năng quản lý và làm việc với một mảng các số nguyên.
- Có tính năng khởi tạo các giá trị ban đầu cho mảng.
- Có tính năng truy xuất vào một phần tử của mảng thông qua chỉ số.
- Có tính năng sắp xếp mảng theo thứ tự tăng dần hoặc giảm dần.
- Có tính năng tìm kiếm một phần tử theo giá trị và trả về vị trí của phần tử đó trong mảng.

**Đáp án tham khảo**

IntegerManager
- array: int[]
+ IntegerManager(int[]) + IntegerManager() + find(int): int + getLength(): int + getValueAt(int): int + initArray(int): void + setArray(int[]): void + setValueAt(int, int): void + sortArray(boolean): void + toArray(): int[]

## 4.3. MẢNG ĐA CHIỀU

- Trong bài toán xây phòng cho sinh viên, nếu như số sinh viên ngày một tăng nhu cầu sử dụng phòng càng nhiều mà diện tích xây phòng chỉ có giới hạn. Để giải quyết vấn đề trên nhà thiết kế không chỉ xây các phòng dưới một tầng, mà thiết kế để xây thêm các tầng khác nữa.
- Để quản lý được các phòng với số tầng khác nhau ta sử dụng mảng 2 chiều với số tầng là chiều thứ nhất, số phòng trên mỗi tầng là chiều thứ 2.
- Mảng được khai báo như sau:
  - `Room[i][j]`
  - Với `i`: số tầng, `j`: số phòng trên mỗi tầng
  - **`a[0][0]`** là phần tử đầu tiên trong mảng cũng là số phòng thứ 1 của tầng thứ nhất

	Column 0	Column 1	Column 2	Column 3
Row 0	a[ 0 ][ 0 ]	a[ 0 ][ 1 ]	a[ 0 ][ 2 ]	a[ 0 ][ 3 ]
Row 1	a[ 1 ][ 0 ]	a[ 1 ][ 1 ]	a[ 1 ][ 2 ]	a[ 1 ][ 3 ]
Row 2	a[ 2 ][ 0 ]	a[ 2 ][ 1 ]	a[ 2 ][ 2 ]	a[ 2 ][ 3 ]

a tên mảng

Chỉ số dòng

Chỉ số cột

## 4.3. KHỞI TẠO MẢNG

Để khởi tạo mảng

```
int [][]a = new int[][]{{1, 2}, {3, 4}};
```

Row 0

Row 1

1	2
3	4

Nếu một phần tử chưa được khởi tạo mặc định được gán bằng null

Khi truy xuất vào phần tử null sẽ sinh lỗi

```
int [][]b = new int[][] {{1}, {3, 4}};
```

1	null
3	4

## 4.3. KHỞI TẠO MẢNG (tiếp theo)

```
public class DemoTwoDemension{
    public static void main(String[] args) {
        int [][]a = new int[][]{{1, 2}, {3, 4}};
        int [][]b = new int[][] {{1}, {3, 4}};
        for(int i = 0; i < 2; i++){
            for(int j = 0; j < 2; j++){
                System.out.println("a["+i+"]["+j+"]": " + a[i][j]);
            }
        }
        //truy xuất vào phần tử thu 2 chưa giá trị null
        System.out.println("b[0][1]" + b[0][1]);
    }
}
```

a[0][0]: 1

a[0][1]: 2

a[1][0]: 3

a[1][1]: 4

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 1  
at DemoTwoDemension.main(DemoTwoDemension.java:18)

Java Result: 1

## CÂU HỎI TRẮC NGHIỆM

Câu hỏi 1 trên 3 ▾

Điểm: 10

Đọc đoạn mã sau

```
String s, t;
```

```
s = "Hello";
```

```
t = "" + "my" + "";
```

```
s.append(t);
```

```
s.toLowerCase();
```

```
s += "friend";
```

```
System.out.println(s);
```

Kết quả sẽ là ?

- ☐ A. Biên dịch lỗi.
- ☐ B. Hello my friend.
- ☐ C. Hello my Friend.
- ☐ D. hello my friend.

### PROPERTIES

On passing, 'Finish' button:

On failing, 'Finish' button:

Allow user to leave quiz:

User may view slides after quiz:

User may attempt quiz:

Goes to Next Slide

Goes to Next Slide

At any time

At any time

Unlimited times



Properties...



Edit in Quizmaker

## 4.4. CHUỖI



## 4.4.1 LỚP STRING

- Lớp `java.lang.String` được dùng mô tả các dữ liệu dạng chuỗi kí tự cố định. Dữ liệu được khởi tạo cho chuỗi luôn là hằng số.
- Ví dụ:

```
String s1 = new String("Hello");  
String s2 = "World";
```

Phương thức	Mô tả
<code>int length()</code>	Trả về độ dài của chuỗi.
<code>char charAt(int index)</code>	Trả về ký tự ở vị trí index trong chuỗi.
<code>String trim()</code>	Loại bỏ khoảng trắng hai đầu của chuỗi.
<code>String replace(char oldChar, char newChar)</code>	Thay thế một ký tự trong chuỗi.
<code>String substring(int start, int end)</code>	Cắt một chuỗi từ vị trí start đến vị trí end.
<code>int compareTo(String s)</code>	So sánh chuỗi hiện tại với chuỗi s. Kết quả = 0 nếu hai chuỗi bằng nhau, >0 nếu chuỗi s nhỏ hơn, <0 nếu chuỗi s lớn hơn.
<code>int indexOf(String s)</code>	Tìm kiếm chuỗi s trong chuỗi nếu tìm thấy trả về vị trí của s trong chuỗi nếu không trả về -1.



## 4.4.1 LỚP STRING (tiếp theo)

```
public class StringDemo{
    public static void main(String[] args) {
        String s1 = "string 1";
        String s2 = "string 2";
        double x = 10.23;
        //Cong 2 chuoi
        System.out.println("s1 + s1: " + (s1 + s2));
        //Cong chuoi voi so thuc, tra ve chuoi
        System.out.println("x = " + x);
        //Noi 2 chuoi
        System.out.println("s1.concat(s2): "+s1.concat(s2));
        //Do dai chuoi s1
        System.out.println("s1.length(): " + s1.length());
    }
}
```

```
s1 + s1: string 1string 2
x = 10.23
s1.concat(s2): string 1string 2
s1.length(): 8
```

## 4.4.2. LỚP STRINGBUILDER

- Lớp `java.lang.StringBuilder` dùng mô tả các dữ liệu dạng chuỗi có thể sửa đổi linh động.
- Sử dụng `StringBuilder` để thao tác chuỗi có ưu điểm tiết kiệm bộ nhớ và tăng tốc khi chương trình có nhiều thao tác xử lý với chuỗi.
- Một số phương thức thường dùng:

Phương thức	Mô tả
<code>append()</code>	Đính thêm các ký tự hoặc chuỗi vào cho <code>StringBuilder</code> .
<code>insert()</code>	Chèn thêm ký tự hoặc chuỗi vào vị trí xác định trong <code>StringBuilder</code> .
<code>delete()</code>	Xóa ký tự hoặc chuỗi trong <code>StringBuilder</code> .
<code>reverse()</code>	Đảo ngược chuỗi trong <code>StringBuilder</code> .
<code>toString()</code>	Chuyển <code>StringBuilder</code> thành <code>String</code> .

## 4.4.2. LỚP STRINGBUILDER (tiếp theo)

```
public class StringBuilderDemo{  
    public static void main(String[] args) {  
        //khai tao constructor voi mot chuoi xac dinh  
        StringBuilder sb = new StringBuilder("string string");  
  
        System.out.println("Truoc dao nguoc: " + sb.toString());  
        //thuc hien dao nguoc chuoi, ket qua luu chinh chuoi do  
        sb = sb.reverse();  
        System.out.println("Sau dao nguoc: " + sb.toString());  
    }  
}
```

```
Truoc dao nguoc: string string  
Sau dao nguoc: gnirts gnirts
```

### 4.4.3. LỚP STRINGTOKENIZER

- Lớp `java.util.StringTokenizer` là lớp tiện ích dùng để phân cắt chuỗi thành các chuỗi thành phần (token) dựa trên một mẫu phân cắt nhất định (delimiter).
- Ví dụ tạo một `StringTokenizer` dùng để phân tích một chuỗi thành các từ dựa trên khoảng trống giữa các từ.

```
import java.util.StringTokenizer;

public class StringTokenizerDemo {
    public static void main(String[] args) {
        StringTokenizer stken = new StringTokenizer("string
string", " ");
        System.out.println(stken.countTokens());
        while (stken.hasMoreTokens()) {
            System.out.println(stken.nextToken());
        }
    }
}
```

```
2
string
string
```

## BÀI TẬP

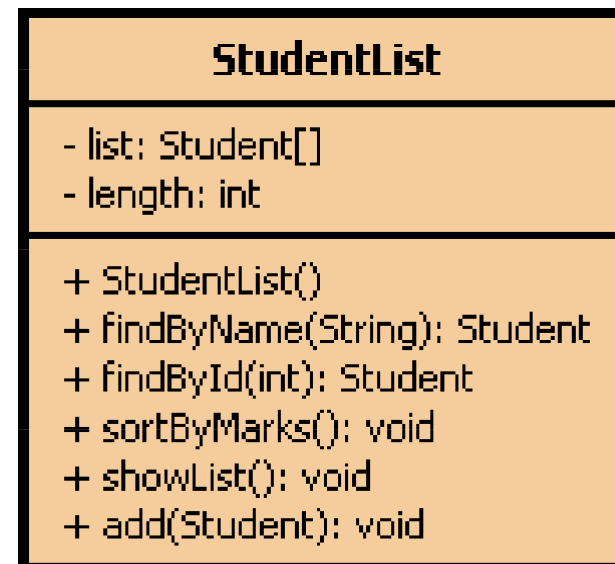
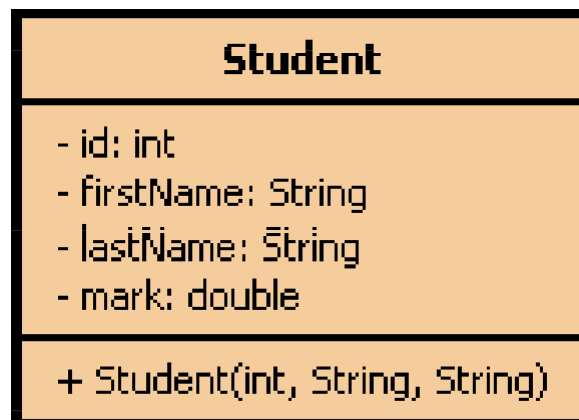
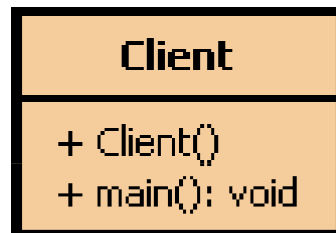
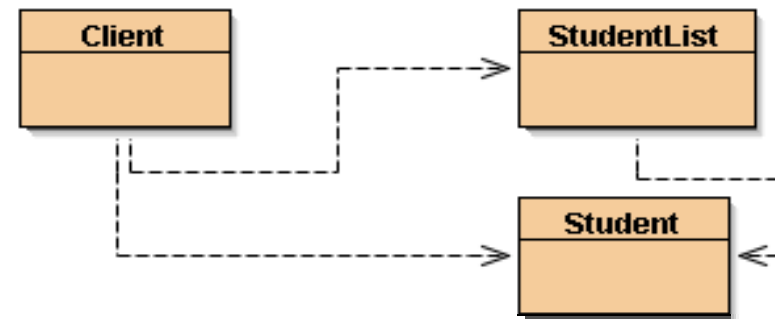
Viết một ứng dụng quản lý danh sách sinh viên với mô tả như sau:

Lớp Client thể hiện chương trình như sau:

===== Student Manager =====

1. Thêm mới một sinh viên vào danh sách.
2. Xóa một sinh viên ra khỏi danh sách.
3. Tìm kiếm sinh viên theo tên.
4. Tìm sinh viên theo Id.
5. In ra danh sách sinh viên theo thứ tự điểm giảm dần.
6. Thoát

#Chon:\_



**Đáp án tham khảo**

## TÓM LƯỢC CUỐI BÀI

**Sau khi học xong bài này chúng ta đã nắm được các kiến thức sau:**

- Nắm được khái niệm về mảng;
- Sử dụng được mảng một chiều và hai chiều trong một số bài toán nhỏ;
- Sử dụng các lớp làm việc với chuỗi.

## CÂU HỎI TRẮC NGHIỆM

Câu hỏi 1 trên 5 ▾

Điểm: 10

Chỉ ra các lệnh khi biên dịch sẽ gặp lỗi

- ☐ A. `float f2[] = new float[];`
- ☐ B. `float[] f1 = new float[3];`
- ☐ C. `float f3[] = new float[3];`
- ☐ D. `float f5[] = { 1.0f, 2.0f, 2.0f };`

### PROPERTIES

On passing, 'Finish' button:

On failing, 'Finish' button:

Allow user to leave quiz:

User may view slides after quiz:

User may attempt quiz:

Goes to Next Slide

Goes to Next Slide

At any time

At any time

Unlimited times



Properties...



Edit in Quizmaker

**Search**

## Làm thế nào để khai báo mảng?

Làm thế nào để khai báo mảng?

### Gợi ý:

Khai báo mảng với cú pháp như sau: `typedata[] arrayname = new typedata[size];`

Với:

- Typedata: kiểu dữ liệu hợp lệ (kiểu cơ sở, kiểu đối tượng);
- Arrayname: tên mảng, tuân theo quy tắc đặt tên định danh;
- Size: số phần tử trong mảng, phải là số nguyên.

## Làm thế nào để truy xuất vào một phần tử trong mảng?

## Điều gì sẽ xảy ra khi chương trình của bạn cố tình truy xuất...?

## Truyền mảng vào trong phương thức là cách truyền tham trị hay tham chiếu?

## Vì sao nói mảng 2 chiều thực chất là mảng 1 chiều?

## So sánh giữa các lớp String, StringBuilder, StringTokenizer

### PROPERTIES

Allow user to leave interaction:

Show 'Next Slide' Button:

Completion Button Label:

Anytime

Don't show

Next Slide







A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Select a term:

Array

ArrayName

Java.lang.String

Java.lang.StringBuilder

Java.util.StringTokenizer

Từ khóa final

Type\_array

## Array

Mảng - tập hợp các phần tử có cùng một kiểu dữ liệu.

### PROPERTIES

Allow user to leave interaction:

Show 'Next Slide' Button:

Completion Button Label:

[Anytime](#)

[Don't show](#)

[Next Slide](#)

