

Bài 08: MẢNG HAI CHIỀU (TWO DIMENSIONAL ARRAY)

Giảng Viên: ThS. Giang Hào Côn

7.1. Khai báo và khởi tạo mảng 2 chiều

Mảng hai chiều (two dimensional array) trong Java dùng để lưu trữ dữ liệu dưới dạng ma trận gồm nhiều dòng và nhiều cột. Mảng 2 chiều lưu trữ các phần tử theo dạng bảng gồm dòng và cột.

	Column 1	Column 2	Column 3	Column 4
Row 1	<code>a[0][0]</code>	<code>a[0][1]</code>	<code>a[0][2]</code>	<code>a[0][3]</code>
Row 2	<code>a[1][0]</code>	<code>a[1][1]</code>	<code>a[1][2]</code>	<code>a[1][3]</code>
Row 3	<code>a[2][0]</code>	<code>a[2][1]</code>	<code>a[2][2]</code>	<code>a[2][3]</code>

7.2/ Cú pháp khai báo mảng 2 chiều

```
elementType[][] arrayRefVar;  
hoặc  
elementType arrayRefVar[][];
```

Trong đó:

- `ElementType` là kiểu dữ liệu.
- `arrayRefVar` là tên biến.

Ví dụ: Khai báo một biến mảng ma trận hai chiều lưu các giá trị kiểu `int`.

```
int[][] matrix;  
hoặc  
int matrix[][];
```

7.3/ Cấp phát vùng nhớ cho mảng 2 chiều

```
int[][] a;  
a = new int[3][4];
```

	Column 1	Column 2	Column 3	Column 4
Row 1	<code>a[0][0]</code>	<code>a[0][1]</code>	<code>a[0][2]</code>	<code>a[0][3]</code>
Row 2	<code>a[1][0]</code>	<code>a[1][1]</code>	<code>a[1][2]</code>	<code>a[1][3]</code>
Row 3	<code>a[2][0]</code>	<code>a[2][1]</code>	<code>a[2][2]</code>	<code>a[2][3]</code>

Chỉ số của dòng và cột của mảng 2 chiều trong Java bắt đầu từ **0**. Ví dụ, phần tử dòng 1, cột 1 là `a[0][0]`.

7.4/ Khởi tạo mảng 2 chiều

```
int[][] arr = {  
    {1, 2, 3, 0},  
    {4, 5, 6, 9},  
    {7, 2, 5, 5},  
};
```

Ở ví dụ này, tạo ra một mảng 2 chiều tên là **arr** và khởi tạo các phần tử của mảng trong dấu ngoặc nhọn {}. Mảng 2 chiều **arr** gồm 3 dòng và 4 cột.

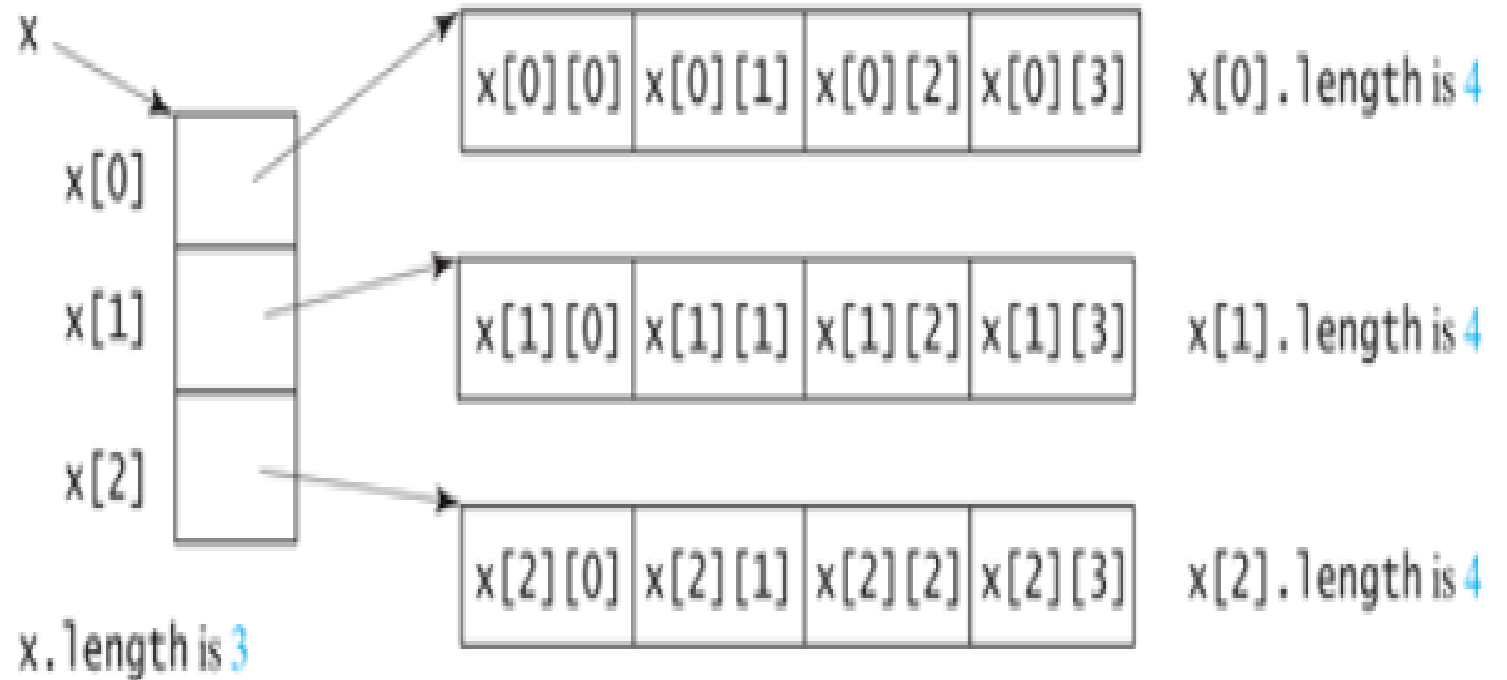
Ngoài ra, chúng ta cũng có thể **khởi tạo mảng 2 chiều** bằng cách **gán giá trị cho từng phần tử** trong mảng thông qua chỉ số dòng và chỉ số cột.

```
int[][] arr = new int[3][4];  
arr[0][0]=1; arr[0][1]=2; arr[0][2]=3; arr[0][3]=0;  
arr[1][0]=4; arr[1][1]=5; arr[1][2]=6; arr[1][3]=9;  
arr[2][0]=7; arr[2][1]=2; arr[2][2]=5; arr[2][3]=5;
```

7.5/ Kích thước của mảng 2 chiều trong Java

Mảng hai chiều là một mảng mà mỗi phần tử là một mảng một chiều.

```
x.length;  
x[0].length;  
x[1].length;  
x[2].length;
```



7.6/ Mảng gồ ghề (jagged array) trong Java

Mỗi dòng của một mảng 2 chiều trong Java là một mảng đơn. Vì vậy, mỗi dòng có thể có độ dài khác nhau. Một mảng như vậy được gọi là **mảng gồ ghề (jagged array)**.

```
int[][] a = {  
    {1, 2, 3},  
    {4, 5, 6, 9},  
    {7},  
};
```

	Column 1	Column 2	Column 3	Column 4
Row 1	1 a[0][0]	2 a[0][1]	3 a[0][2]	
Row 2	4 a[1][0]	5 a[1][1]	6 a[1][2]	9 a[1][3]
Row 3	7 a[2][0]			

7.7/ Kỹ thuật lập trình với mảng 2 chiều trong Java

1. Nhập xuất mảng 2 chiều trong Java

```
public class Main {  
    public static void main(String[] args)  
    {  
        int[][] matrix = new int[3][4];  
        Scanner input = new Scanner(System.in);  
        System.out.println("Nhập " + matrix.length + " dòng và " + matrix[0].length + " cột của matrix:");  
        for(int row = 0; row < matrix.length ; row++) {  
            for(int column = 0; column < matrix[row].length; column++) {  
                matrix[row][column] = input.nextInt();  
            }  
        }  
  
        System.out.println("Xuất matrix có " + matrix.length + " dòng và " + matrix[0].length + " cột:");  
        for(int row = 0; row < matrix.length ; row++) {  
            for(int column = 0; column < matrix[row].length ; column++) {  
                System.out.print(matrix[row][column] + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

Kết Quả

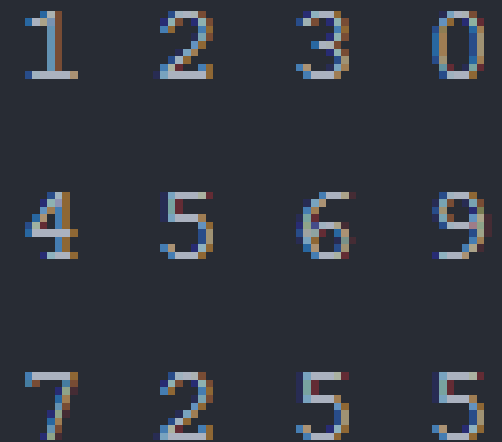
```
Nhap 3 dòng và 4 cột của matrix:  
1 2 3 0  
4 5 6 9  
7 2 5 5  
  
Xuat matrix có 3 dòng và 4 cột:  
1 2 3 0  
4 5 6 9  
7 2 5 5
```


7.7/ Kỹ thuật lập trình với mảng 2 chiều trong Java

2. Xuất mảng 2 chiều sử dụng vòng lặp for-each

```
public class Main {  
    public static void main(String[] args) {  
        int[][] matrix = {  
            {1, 2, 3, 0},  
            {4, 5, 6, 9},  
            {7, 2, 5, 5}  
        };  
        //vòng lặp for-each đầu tiên truy cập đến mảng 1 chiều chứa các dòng  
        for (int[] innerArray: matrix) {  
            //vòng lặp for-each truy cập đến từng phần tử trong các dòng  
            for(int data: innerArray) {  
                System.out.print(data + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

Kết Quả



1	2	3	0
4	5	6	9
7	2	5	5

3. Tính tổng các phần tử trong mảng 2 chiều

```
public class Main {  
    public static void main(String[] args) {  
        int[][] matrix = {  
            {1, 2, 3, 0},  
            {4, 5, 6, 9},  
            {7, 2, 5, 5}  
        };  
        int total = 0;  
        for(int row = 0; row < matrix.length; row++) {  
            for(int column = 0; column < matrix[row].length; column++) {  
                total += matrix[row][column];  
            }  
        }  
        System.out.println("Tong cac phan tu trong matrix = " + total);  
    }  
}
```

Kết Quả

Tong cac phan tu trong matrix = 49

4. Tính tổng các phần tử trong mảng 2 chiều theo cột

```
public class Main {  
    public static void main(String[] args) {  
        int[][] matrix = {  
            {1, 2, 3, 0},  
            {4, 5, 6, 9},  
            {7, 2, 5, 5}  
        };  
        for(int column = 0; column < matrix[0].length; column++){  
            int total = 0;  
            for(int row = 0; row < matrix.length; row++){  
                total += matrix[row][column];  
            }  
            System.out.println("Sum for column " + column + " is " + total);  
        }  
    }  
}
```

Kết Quả

```
Sum for column 0 is 12  
Sum for column 1 is 9  
Sum for column 2 is 14  
Sum for column 3 is 14
```

5. Tính tổng các phần tử trên đường chéo chính của ma trận

Đường chéo chính của một ma trận A bao gồm các phần tử $A_{i,j}$ với $i=j$. Ba ma trận sau đây đều có đường chéo chính được biểu thị bởi các số 1 màu đỏ:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

7.7/ Kỹ thuật lập trình với mảng 2 chiều trong Java

5. Tính tổng các phần tử trên đường chéo chính của ma trận

Bên dưới là một ma trận được biểu diễn bằng mảng 2 chiều. **Các phần tử màu đỏ** nằm trên đường chéo chính của ma trận.

	Column 1	Column 2	Column 3	Column 4
Row 1	<div>1</div> <div>a[0][0]</div>	<div>2</div> <div>a[0][1]</div>	<div>3</div> <div>a[0][2]</div>	<div>0</div> <div>a[0][3]</div>
Row 2	<div>4</div> <div>a[1][0]</div>	<div>5</div> <div>a[1][1]</div>	<div>6</div> <div>a[1][2]</div>	<div>9</div> <div>a[1][3]</div>
Row 3	<div>7</div> <div>a[2][0]</div>	<div>2</div> <div>a[2][1]</div>	<div>5</div> <div>a[2][2]</div>	<div>5</div> <div>a[2][3]</div>

5. Tính tổng các phần tử trên đường chéo chính của ma trận

```
public class Main {  
    public static void main(String[] args) {  
        int[][] matrix = {  
            {1, 2, 3, 0},  
            {4, 5, 6, 9},  
            {7, 2, 5, 5}  
        };  
        int total = 0;  
        for(int row = 0; row < matrix.length; row++) {  
            for(int column = 0; column < matrix[row].length; column++) {  
                if(row==column){  
                    total += matrix[row][column];  
                }  
            }  
        }  
        System.out.println("Tong cac phan tu tren duong cheo chinh cua matrix = " + total);  
    }  
}
```

6. Truyền mảng 2 chiều cho phương thức trong Java

```
public class Main {  
    public static void changeMatrix(int[][] arr){  
        arr[0][0]=1; arr[0][1]=0; arr[0][2]=0; arr[0][3]=0;  
        arr[1][0]=0; arr[1][1]=1; arr[1][2]=0; arr[1][3]=0;  
        arr[2][0]=0; arr[2][1]=0; arr[2][2]=1; arr[2][3]=0;  
    }  
    public static void main(String[] args) {  
        int[][] matrix = {  
            {1, 2, 3, 0},  
            {4, 5, 6, 9},  
            {7, 2, 5, 5}  
        };  
    }  
}
```

6. Truyền mảng 2 chiều cho phương thức trong Java

```
System.out.println("Matrix trước khi gọi hàm changeMatrix()");  
for(int row = 0; row < matrix.length ; row++) {  
    for(int column = 0; column < matrix[row].length ; column++) {  
        System.out.print(matrix[row][column] + " ");  
    }  
    System.out.println();  
}
```


6. Truyền mảng 2 chiều cho phương thức trong Java

```
changeMatrix(matrix);  
System.out.println("Matrix sau khi gọi hàm changeMatrix()");  
for(int row = 0; row < matrix.length ; row++) {  
    for(int column = 0; column < matrix[row].length ; column++) {  
        System.out.print(matrix[row][column] + " ");  
    }  
    System.out.println();  
}
```

7.7/ Kỹ thuật lập trình với mảng 2 chiều trong Java

7. Kết quả trả về của một phương thức có thể là một mảng 2 chiều

```
public class Main {  
    public static int[][] initMatrix(int row, int col){  
        int[][] matrix = new int[row][col];  
        matrix[0][0]=1; matrix[0][1]=0; matrix[0][2]=0; matrix[0][3]=0;  
        matrix[1][0]=0; matrix[1][1]=1; matrix[1][2]=0; matrix[1][3]=0;  
        matrix[2][0]=0; matrix[2][1]=0; matrix[2][2]=1; matrix[2][3]=0;  
        return matrix;  
    }  
}
```

7.7/ Kỹ thuật lập trình với mảng 2 chiều trong Java

7. Kết quả trả về của một phương thức có thể là một mảng 2 chiều

```
} public static void main(String[] args) {  
    int row = 3;  
    int col = 4;  
    int[][] matrix;  
    matrix = initMatrix(row, col);  
    System.out.println("Matrix duoc khoi tao:");  
    for(int i = 0; i < matrix.length ; i++) {  
        for(int j = 0; j < matrix[i].length ; j++) {  
            System.out.print(matrix[i][j] + " ");  
        }  
        System.out.println();  
    }  
}
```

Kết Quả

```
Matrix duoc khoi tao:  
1 0 0 0  
0 1 0 0  
0 0 1 0
```

7.8/ Kỹ thuật lập trình sao chép mảng trong Java

1. Sao chép mảng sử dụng toán tử gán

```
public class Main {  
    public static void main(String[] args) {  
        int [] numbers = {1, 2, 3, 4, 5, 6};  
        int [] copyArr = numbers;  
        System.out.print("Mang copyArr sao chep duoc: ");  
        for (int number: copyArr) {  
            System.out.print(number + " ");  
        }  
    }  
}
```

7.8/ Kỹ thuật lập trình sao chép mảng trong Java

2. Gán từng phần tử của mảng này vào mảng kia

```
public class Main {  
    public static void main(String[] args) {  
        int [] numbers = {1, 2, 3, 4, 5, 6};  
        int [] copyArr = new int[6];  
        //sao chép mảng numbers vào mảng copyArr  
        for (int i = 0; i < numbers.length; ++i) {  
            copyArr[i] = numbers[i];  
        }  
        System.out.print("Mang copyArr sao chép được: ");  
        for (int number: copyArr) {  
            System.out.print(number + " ");  
        }  
    }  
}
```

7.8/ Kỹ thuật lập trình sao chép mảng trong Java

3. Sao chép mảng sử dụng hàm `arraycopy()`

Trong Java, lớp `java.lang.System` có hỗ trợ hàm `arraycopy()` để sao chép mảng. Hàm `arraycopy()` cho phép sao chép một phần cụ thể của mảng nguồn sang mảng đích. Cú pháp của hàm `arraycopy()`:

```
arraycopy(Object src, int srcPos, Object dest, int destPos, int length)
```

Trong đó:

- `src` là mảng nguồn muốn sao chép
- `srcPos` là vị trí phần tử bắt đầu sẽ sao chép trên mảng nguồn
- `dest` là mảng đích muốn sao chép từ mảng nguồn sang mảng đích
- `destPos` là vị trí phần tử bắt đầu trên mảng đích mà mảng nguồn sao chép sang
- `length` là số phần tử muốn sao chép từ mảng nguồn sang mảng đích

7.8/ Kỹ thuật lập trình sao chép mảng trong Java

3. Sao chép mảng sử dụng hàm arraycopy()

```
public class Main {  
    public static void main(String[] args) {  
        int [] numbers = {1, 2, 3, 4, 5, 6};  
        int [] copyArr1 = new int[5];  
        int [] copyArr2 = new int[numbers.length];  
        System.arraycopy(numbers, 2, copyArr1, 1, 3);  
        System.out.print("Mang copyArr1 sao chep duoc: ");  
        for (int number: copyArr1) {  
            System.out.print(number + " ");  
        }  
        System.out.println();  
        System.arraycopy(numbers, 0, copyArr2, 0, numbers.length);  
        System.out.print("Mang copyArr2 sao chep duoc: ");  
        for (int number: copyArr2) {  
            System.out.print(number + " ");  
        }  
    }  
}
```

Kết Quả

```
Mang copyArr1 sao chep duoc: 0 3 4 5 0  
Mang copyArr2 sao chep duoc: 1 2 3 4 5 6
```

7.8/ Kỹ thuật lập trình sao chép mảng trong Java

4. Sao chép mảng sử dụng hàm copyOfRange()

```
public class Main {  
    public static void main(String[] args) {  
        int [] numbers = {1, 2, 3, 4, 5, 6};  
        //sao chép toàn bộ mảng numbers  
        int [] copyArr1 = Arrays.copyOfRange(numbers, 0, numbers.length);  
        System.out.print("Mang copyArr1 sao chép được: ");  
        for (int number: copyArr1) {  
            System.out.print(number + " ");  
        }  
        System.out.println();  
        //sao chép mảng numbers từ index 2 đến 5 (không bao gồm index=5)  
        int [] copyArr2 = Arrays.copyOfRange(numbers, 2, 5);  
        System.out.print("Mang copyArr2 sao chép được: ");  
        for (int number: copyArr2) {  
            System.out.print(number + " ");  
        }  
    }  
}
```

Kết Quả

```
Mang copyArr1 sao chép được: 1 2 3 4 5 6  
Mang copyArr2 sao chép được: 3 4 5
```


7.8/ Kỹ thuật lập trình sao chép mảng trong Java

5. Sao chép mảng 2 chiều

```
import java.util.Arrays;

public class Main {
    public static void main(String[] args) {
        int[][] source = {
            {1, 2, 3, 4},
            {5, 6},
            {0, 2, 42, -4, 5}
        };
        int[][] dest = new int[source.length][];
        for (int i = 0; i < dest.length; ++i) {
            dest[i] = new int[source[i].length];
            for (int j = 0; j < dest[i].length; ++j) {
                dest[i][j] = source[i][j];
            }
        }
        System.out.print("Mang 2 chieu sao chep duoc:");
        System.out.println(Arrays.deepToString(dest));
    }
}
```

Kết Quả

Mang 2 chieu sao chep duoc:[[1, 2, 3, 4], [5, 6], [0, 2, 42, -4, 5]]

STRING CLASS

Giảng Viên: ThS. Giang Hào Côn

Dữ liệu chuỗi – **String**

Chuỗi là tập hợp các ký tự được cấp phát tuần tự, liên tục trong bộ nhớ. Đây là loại dữ liệu sử dụng phổ biến trong chương trình. Trong Java, chuỗi được xem là một loại “**đối tượng**”, do đó, java cung cấp một class cùng với các phương thức phục vụ cho việc xử lý đối với dữ liệu thuộc loại này với tên “**String**” class thuộc namespace: **java.lang**

Có thể khai báo để sử dụng chuỗi theo một trong các cách sau:

+ Khai báo trực tiếp

```
String tenTruong = “Nguyen Tat Thanh”;
```

Dữ liệu chuỗi – **String**

+ Khai báo để chuyển 1 chuỗi từ mảng ký tự

```
char data[] = {'N', 'g', 'u', 'y', 'e', 'n', ' ', 'T', 'a', 't', ' ', 'T', 'h', 'a', 'n', 'h'};
```

```
String str = new String(data);
```

N	g	u	y	e	n		T	a	t		T	h	a	n	h
---	---	---	---	---	---	--	---	---	---	--	---	---	---	---	---

Length = ? ; vị trí đầu tiên ? ; vị trí thứ 7 ? ; vị trí thứ 12 ? ;

Nhập - Xuất chuỗi

Có thể sử dụng phương thức **nextLine** của đối tượng scanner như sau. VD: *Khai báo biến hoTen chứa chuỗi đồng thời tiến hành nhập dữ liệu cho biến này ta làm như sau:*

```
Scanner nhap = new Scanner(System.in);
```

```
System.out.print("Ho va ten:\t");
```

```
String hoTen = nhap.nextLine();
```

Nhập - Xuất chuỗi

Sử dụng phương thức **print** hoặc **println** để in nội dung của 1 đối tượng **String** ra màn hình giống như các loại dữ liệu khác (*Có thể sử dụng kết hợp với **escape character***)

+ Xuất nội dung của biến **hoTen** lên màn hình, kết hợp với **placeholder** cùng **escape character** ta làm như sau:

```
System.out.print("Sinh viên: %s \t chuyên ngành CNTT \n Lớp:  
18DTH2D ", hoTen);
```

Các hàm liên quan đến chuỗi

- So sánh chuỗi

public boolean **equals**(Object anObject)

public boolean **equalsIgnoreCase**(String anotherString)

public boolean **startsWith**(String prefix)

public boolean **endsWith**(String suffix)

- Chuyển đổi

public String **toUpperCase**()

public String **toLowerCase**()

public String **length**()

Các hàm liên quan đến chuỗi

■ Cắt chuỗi

public String **trim**()

public String **substring**(int beginIndex, int endIndex)

public String[] **split**(String regex, int limit)

Bài tập Thảo Luận

Câu 01: Tìm hiểu chức năng của các phương thức

- Cho một số ví dụ liên quan đến việc so sánh:
 - ❖ **equals**
 - ❖ **startsWith**
 - ❖ **endsWith**
- Ví dụ liên quan đến việc ứng dụng **split** 1 chuỗi
- Ví dụ liên quan đến: **LowerCase**, **UpperCase**, **Trim** và **substring**

Câu 02:

- Viết phương thức **nhập** mảng 2 chiều
- Viết phương thức **xuất** mảng 2 chiều
- Viết phương thức tính tích các phần tử trên đường chéo chính
- Viết phương thức đếm số phần tử trong mảng 2 chiều bằng giá trị x, biết rằng x được nhập từ bàn phím.
- Viết phương thức liệt kê các phần tử phía trên đường chéo chính.
- Viết phương thức tính tổng các số chẵn chia hết cho 4.