

LẬP TRÌNH JAVA

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG VỚI JAVA

Nguyễn Hoàng Anh

Email: nhanh@fit.hcmus.edu.vn

ĐH KHTN, 2011

Nội dung

- Khai báo lớp đối tượng
- Khai báo thuộc tính
- Định nghĩa các phương thức khởi tạo
- Định nghĩa các phương thức get/set
- Định nghĩa các phương thức nhập xuất console
- Định nghĩa các phương thức xử lý nghiệp vụ
- Định nghĩa các phương thức overload
- Định nghĩa các phương thức mức lớp (static)

KHAI BÁO LỚP

Khai báo lớp

```
1 package tenpackage;  
2  
3 public class TenLop {  
4     //Khai báo các thuộc tính  
5  
6     //Khai báo các phương thức  
7  
8 }
```

```
1 package bt1;  
2  
3 public class PhanSo{  
4     //Khai báo các thuộc tính  
5  
6     //Khai báo các phương thức  
7  
8 }
```

KHAI BÁO THUỘC TÍNH

Khai báo thuộc tính

```
1 package tenpackage;  
2  
3 public class TenLop {  
4     //Khai báo các thuộc tính  
5     private KieuDuLieu tenThuocTinh1;  
6     ...  
7     private KieuDuLieu tenThuocTinhN1;  
8     //Khai báo các phương thức  
9     ...  
10 }
```

Khai báo thuộc tính

```
1 package bt1;
2
3 public class PhanSo{
4     //Khai báo các thuộc tính
5     private int tuSo;
6     private int mauSo;
7     //Khai báo các phương thức
8     ...
9 }
```

Khai báo thuộc tính

```
1 package bt1;
2
3 public class MangSoNguyen{
4     //Khai báo các thuộc tính
5     private int [] arr;
6     //Khai báo các phương thức
7
8 }
```

```
1 package bt1;
2
3 public class MangPhanSo{
4     //Khai báo các thuộc tính
5     private PhanSo [] arr;
6     //Khai báo các phương thức
7
8 }
```




HỎI VÀ ĐÁP

PHƯƠNG THỨC KHỞI TẠO ĐỐI TƯỢNG

Định nghĩa phương thức khởi tạo đối tượng

```
1 package tenpackage;
2 public class TenLop{
3     //Khai báo các thuộc tính
4     private KieuDuLieu thuocTinh1; . . .
5     private KieuDuLieu thuocTinhN;
6     //Phương thức khởi tạo mặc định
7     public TenLop() {
8         this.thuocTinh1=Giá trị mặc định; . . .
9         this.thuocTinhN=Giá trị mặc định;
10    }
11    //Phương thức khởi tạo đầy đủ tham số
12    public TenLop(KieuDuLieu thuocTinh1,..., KieuDuLieu thuocTinhN) {
13        this.thuocTinh1=thuocTinh1; . . .
14        this.thuocTinhN=thuocTinhN;
15    }
16    //Phương thức khởi tạo sao chép
17    public TenLop(TenLop tl){
18        this.thuocTinh1=tl.thuocTinh1; . . .
19        this.thuocTinhN=tl.thuocTinhN;
20    }
21 }
```

Định nghĩa phương thức khởi tạo đối tượng

```
1 package bt1;
2 public class PhanSo{
3     //Khai báo các thuộc tính
4     private int tuSo;
5     private int mauSo;
6     //Phương thức khởi tạo mặc định
7     public PhanSo () {
8         this.tuSo=1;
9         this.mauSo=2;
10    }
11    //Phương thức khởi tạo đầy đủ tham số
12    public PhanSo(int tuSo, int mauSo){
13        this.tuSo=tuSo;
14        this.mauSo=mauSo;
15    }
16    //Phương thức khởi tạo sao chép
17    public PhanSo(PhanSo ps){
18        this.tuSo=ps.tuSo;
19        this.mauSo=ps.mauSo;
20    }
21 }
```

Định nghĩa phương thức khởi tạo đối tượng

```
1 package bt1;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         PhanSo ps1=new PhanSo();
7         PhanSo ps2=new PhanSo(2,3);
8         PhanSo ps3=new PhanSo(ps2);
9     }
10
11 }
```

Định nghĩa phương thức khởi tạo đối tượng

```
1 package bt1;
2 public class MangPhanSo {
3     //Khai báo thuộc tính
4     private PhanSo[] arr;
5     //Phương thức khởi tạo mặc định
6     public MangPhanSo() {
7         this.arr = null;
8     }
9     //Phương thức khởi tạo có tham số
10    public MangPhanSo(PhanSo... arr) {
11        this.arr = new PhanSo[arr.length];
12        for (int i = 0; i < arr.length; i++) {
13            this.arr[i] = new PhanSo(arr[i]);
14        }
15    }
16    //Phương thức khởi tạo sao chép
17    public MangPhanSo(MangPhanSo mps) {
18        this.arr = new PhanSo[mps.arr.length];
19        for (int i = 0; i < mps.arr.length; i++) {
20            this.arr[i] = new PhanSo(mps.arr[i]);
21        }
22    }
23 }
```

Định nghĩa phương thức khởi tạo đối tượng

```
1 package bt1;
2 public class Main {
3     public static void main(String[] args) {
4         //Sử dụng phương thức khởi tạo mặc định
5         MangPhanSo mps1=new MangPhanSo();
6         //Sử dụng phương thức khởi tạo có tham số
7         MangPhanSo mps2=new MangPhanSo(new PhanSo(2,3));
8         MangPhanSo mps3=new MangPhanSo(new PhanSo(1,3),
9                                         new PhanSo(4,3),
10                                        new PhanSo(5,9));
11
12         PhanSo [] arr = new PhanSo[2];
13         arr[0]=new PhanSo(2,4);
14         arr[1]=new PhanSo(3,7);
15         MangPhanSo mps4=new MangPhanSo(arr);
16         //Sử dụng phương thức khởi tạo sao chép
17         MangPhanSo mps5=new MangPhanSo(mps3);
18     }
19 }
```

Định nghĩa phương thức khởi tạo đối tượng

```
1 package bt1;
2 public class MangSoNguyen {
3     //Khai báo thuộc tính
4     private int[] arr;
5     //Phương thức khởi tạo mặc định
6     public MangSoNguyen() {
7         this.arr = null;
8     }
9     //Phương thức khởi tạo có tham số
10    public MangSoNguyen(int... arr) {
11        this.arr = new int[arr.length];
12        for (int i = 0; i < arr.length; i++) {
13            this.arr[i] = arr[i];
14        }
15    }
16    //Phương thức khởi tạo sao chép
17    public MangSoNguyen(MangSoNguyen msn) {
18        this.arr = new int[msn.arr.length];
19        for (int i = 0; i < msn.arr.length; i++) {
20            this.arr[i] = msn.arr[i];
21        }
22    }
23 }
```


Định nghĩa phương thức khởi tạo đối tượng

```
1 package bt1;
2 public class Main {
3     public static void main(String[] args) {
4         //Sử dụng phương thức khởi tạo mặc định
5         MangSoNguyen msn1=new MangSoNguyen();
6         //Sử dụng phương thức khởi tạo có tham số
7         MangSoNguyen msn2=new MangSoNguyen(3);
8         MangSoNguyen msn3=new MangSoNguyen(3, 2, 8);
9         MangSoNguyen msn4=new MangSoNguyen(4, 9, 1, 5);
10        int [] arr = new int{3,4,7};
11        MangSoNguyen msn5=new MangSoNguyen(arr);
12        //Sử dụng phương thức khởi tạo sao chép
13        MangSoNguyen msn6=new MangSoNguyen(mps3);
14    }
15 }
```



HỎI VÀ ĐÁP

GETTER / SETTER

Định nghĩa các phương thức get/set

```
1 package tenpackage;
2 public class TenLop{
3     //Khai báo các thuộc tính
4     private KieuDuLieu thuocTinh1; . . .
5     private KieuDuLieu thuocTinhN;
6     //Các phương thức khởi tạo đối tượng
7     ...
8     //Các phương thức get/set
9     public KieuDuLieu getThuocTinh1() {
10         return thuocTinh1;
11     }
12     public void setThuocTinh1(KieuDuLieu thuocTinh1) {
13         this.thuocTinh1 = thuocTinh1;
14     } . . .
15     public KieuDuLieu getThuocTinhN() {
16         return thuocTinhN;
17     }
18     public void setThuocTinhN(KieuDuLieu thuocTinhN) {
19         this.thuocTinhN = thuocTinhN;
20     }
21 }
```

Định nghĩa các phương thức get/set

```
1 public class PhanSo{
2     //Khai báo các thuộc tính
3     private int tuSo; private int mauSo;
4     //Phương thức khởi tạo đối tượng
5     . . .
6     //Các phương thức get/set
7     public int getTuSo() {
8         return tuSo;
9     }
10    public void setTuSo(int tuSo) {
11        this.tuSo = tuSo;
12    }
13    public int getMauSo() {
14        return mauSo;
15    }
16    public void setMauSo(int mauSo) {
17        if (mauSo != 0) {
18            this.mauSo = mauSo;
19        }
20    }
21 }
```

Định nghĩa các phương thức get/set

```
1 package bt1;
2
3 public class Main {
4
5     public static void main(String[] args) {
6
7         PhanSo ps1=new PhanSo();
8         System.out.println(ps1.getTuSo()
9                             + "/" + ps1.getMauSo());
10        ps1.setTuSo(3);
11        ps1.setMauSo(4);
12        System.out.println(ps1.getTuSo()
13                            + "/" + ps1.getMauSo());
14
15    }
16 }
```

Định nghĩa các phương thức get/set

```
1 public class MangPhanSo {
2     //Khai báo thuộc tính
3     private PhanSo[] arr;
4     //Phương thức khởi tạo mặc định
5     . . .
6     public PhanSo[] getArr() {
7         return arr;
8     }
9
10    public void setArr(PhanSo ... arr) {
11        this.arr = new PhanSo[arr.length];
12        for (int i = 0; i < arr.length; i++) {
13            this.arr[i] = new PhanSo(arr[i]);
14        }
15    }
16    . . .
17 }
```

Định nghĩa các phương thức get/set

```
1      . . .
2      public void set(int index, PhanSo value) {
3          if (index >= 0 && index < this.arr.length) {
4              //this.arr[index] = new PhanSo(value);
5              this.arr[index]=value;
6          }
7      }
8
9      public PhanSo get(int index) {
10         PhanSo ps = null;
11         if (index >= 0 && index < this.arr.length) {
12             //ps=new PhanSo(this.arr[index]);
13             ps = this.arr[index];
14         }
15         return ps;
16     }
17 }
```


Định nghĩa các phương thức get/set

```
1 public static void main(String[] args) {
2     MangPhanSo mps = new MangPhanSo(
3         new PhanSo(1, 3),
4         new PhanSo(4, 3),
5         new PhanSo(5, 9));
6     //Sử dụng get
7     PhanSo[] arr = mps.getArr();
8     int index = 1;
9     PhanSo ps = mps.get(index);
10    //Sử dụng set
11    PhanSo[] arr2 = new PhanSo[2];
12    arr2[0] = new PhanSo(2, 4);
13    arr2[1] = new PhanSo(3, 7);
14    mps.setArr(arr2);
15    index=0;
16    mps.set(index, new PhanSo(8, 5));
17 }
```

Định nghĩa các phương thức get/set

```
1 package bt1;
2 public class MangSoNguyen {
3     //Khai báo thuộc tính
4     private int[] arr;
5     //Phương thức khởi tạo mặc định
6     . . .
7     public int[] getArr() {
8         return arr;
9     }
10    public void setArr(int ... arr) {
11        this.arr = new int[arr.length];
12        for (int i = 0; i < arr.length; i++) {
13            this.arr[i] = rr[i];
14        }
15    }
16    . . .
17 }
```

Định nghĩa các phương thức get/set

```
1      . . .
2      public void set(int index, int value) {
3          if (index >= 0 && index < this.arr.length) {
4              this.arr[index]=value;
5          }
6      }
7
8      public Integer get(int index) {
9          Integer value = null;
10         if (index >= 0 && index < this.arr.length) {
11             value = this.arr[index];
12         }
13         return value;
14     }
15 }
```

Định nghĩa các phương thức get/set

```
1 public static void main(String[] args) {
2     MangSoNguyen mps = new MangSoNguyen( 3, 9, 5);
3     //Sử dụng get
4     int[] arr = mps.getArr();
5     int index = 1;
6     Integer value = mps.get(index);
7     //Sử dụng set
8     int[] arr2 = {9, 6, 8};
9     mps.setArr(arr2);
10    index=0;
11    mps.set(index, 7);
12 }
```



HỎI VÀ ĐÁP

NHẬP XUẤT CONSOLE

Định nghĩa các phương thức nhập xuất console

```
1 public class PhanSo{
2     //Khai báo các thuộc tính
3     private int tuSo;
4     private int mauSo;
5     . . .
6     public void nhap(String tieuDe) {
7         System.out.println(tieuDe);
8         Scanner scan = new Scanner(System.in);
9         System.out.print("Tử số:");
10        this.tuSo = Integer.parseInt(scan.nextLine());
11        System.out.print("Mẫu số:");
12        this.mauSo = Integer.parseInt(scan.nextLine());
13    }
14
15    public void xuat(String tieuDe){
16        System.out.println(tieuDe);
17        String str=String.format("%d/%d", this.tuSo, this.mauSo);
18        System.out.println(str);
19    }
20 }
21 }
```

Định nghĩa các phương thức nhập xuất console

```
1 package bt1;
2
3 public class Main {
4
5     public static void main(String[] args) {
6
7         PhanSo ps1=new PhanSo();
8         ps1.nhap("Nhập phân số 1");
9         ps1.xuat("Xuất phân số 1");
10
11        PhanSo ps2=new PhanSo();
12        ps2.nhap("Nhập phân số 2");
13        ps2.xuat("Xuất phân số 2");
14
15    }
16 }
```


Định nghĩa các phương thức nhập xuất console

```
1 package bt1;
2 public class MangPhanSo {
3     //Khai báo thuộc tính
4     private PhanSo[] arr;
5     . . .
6     public void nhap(String tieuDe){
7         Scanner scan=new Scanner(System.in);
8         System.out.println(tieuDe);
9         System.out.print("Số lượng phần tử:");
10        int n=Integer.parseInt(scan.nextLine());
11        this.arr=new PhanSo[n];
12        for(int i=0; i<this.arr.length; i++){
13            this.arr[i] = new PhanSo();
14            this.arr[i].nhap("Phần số thứ "+i);
15        }
16    }
17    . . .
18 }
```

Định nghĩa các phương thức nhập xuất console

```
1      . . .
2      public void xuat(String tieuDe){
3          System.out.println(tieuDe);
4          int n = this.arr.length;
5          System.out.println("Số lượng phần tử: " + n);
6          for(int i=0; i<this.arr.length; i++){
7              this.arr[i].xuat("Phần số thứ "+i);
8          }
9      }
10 }
```

Định nghĩa các phương thức nhập xuất console

```
1 public static void main(String[] args) {  
2  
3     MangPhanSo mps=new MangPhanSo();  
4     mps.nhap("Nhập mảng phân số");  
5     mps.xuat("Xuất mảng phân số");  
6  
7 }
```

Định nghĩa các phương thức nhập xuất console

```
1 package bt1;
2 public class MangSoNguyen {
3     //Khai báo thuộc tính
4     private int [] arr;
5     . . .
6     public void nhap(String tieuDe){
7         Scanner scan=new Scanner(System.in);
8         System.out.println(tieuDe);
9         System.out.print("Số lượng phần tử:");
10        int n=Integer.parseInt(scan.nextLine());
11        this.arr=new int[n];
12        for(int i=0; i<this.arr.length; i++){
13            System.out.print("Phần tử thứ "+ i + ":");
14            this.arr[i]=Integer.parseInt(scan.nextLine())
15        }
16    }
17    . . .
}
```

Định nghĩa các phương thức nhập xuất console

```
1      . . .
2      public void xuất(String tieuDe){
3          System.out.println(tieuDe);
4          int n = this.arr.length;
5          System.out.println("Số lượng phần tử: " + n);
6          for(int i=0; i<this.arr.length; i++){
7              System.out.print("Phần số thứ "+i+ ":");
8              System.out.println(this.arr[i]);
9          }
10     }
11 }
```

Định nghĩa các phương thức nhập xuất console

```
1 public static void main(String[] args) {  
2  
3     MangSoNguyen msn=new MangSoNguyen();  
4     msn.nhap("Nhập mảng số nguyên");  
5     msn.xuat("Xuất mảng số nguyên");  
6  
7 }
```



HỎI VÀ ĐÁP

XỬ LÝ NGHIỆP VỤ

Định nghĩa các phương thức xử lý nghiệp vụ

```
1 public class PhanSo{
2     //Khai báo các thuộc tính
3     private int tuSo;
4     private int mauSo;
5     . . .
6     public PhanSo cong(PhanSo ps){
7         PhanSo kq=new PhanSo();
8         kq.tuSo=this.tuSo*ps.mauSo+this.mauSo*ps.tuSo;
9         kq.mauSo=this.mauSo*ps.mauSo;
10        return kq;
11    }
12    . . .
13 }
```

Định nghĩa các phương thức xử lý nghiệp vụ

```
1 package bt1;
2 public class Main {
3     public static void main(String[] args) {
4
5         PhanSo ps1=new PhanSo();
6         ps1.nhap("Nhập phân số 1");
7         ps1.xuat("Xuất phân số 1");
8
9         PhanSo ps2=new PhanSo();
10        ps2.nhap("Nhập phân số 2");
11        ps2.xuat("Xuất phân số 2");
12
13        PhanSo kq=ps1.cong(ps2);
14        kq.xuat("Tổng hai phân số");
15    }
16 }
```

Định nghĩa các phương thức xử lý nghiệp vụ

```
1 package bt1;
2 public class MangPhanSo {
3     //Khai báo thuộc tính
4     private PhanSo[] arr;
5     . . .
6     public PhanSo tong() {
7         PhanSo s=this.arr[0];
8         for(int i=1; i<this.arr.length; i++){
9             s=s.cong(this.arr[i]);
10        }
11    }
12    . . .
13 }
```

Định nghĩa các phương thức xử lý nghiệp vụ

```
1 public static void main(String[] args) {  
2  
3     MangPhanSo mps=new MangPhanSo();  
4     mps.nhap("Nhập mảng phân số");  
5     mps.xuat("Xuất mảng phân số");  
6     PhanSo s=mps.tong();  
7     s.xuat("Tổng mảng phân số");  
8  
9 }
```

Định nghĩa các phương thức xử lý nghiệp vụ

```
1 package bt1;
2 public class MangSoNguyen {
3     //Khai báo thuộc tính
4     private int[] arr;
5     . . .
6     public int tong() {
7         int s=0;
8         for(int i=0; i<this.arr.length; i++){
9             s=s + this.arr[i];
10        }
11    }
12    . . .
13 }
```

Định nghĩa các phương thức xử lý nghiệp vụ

```
1 public static void main(String[] args) {  
2  
3     MangSoNguyen msn=new MangSoNguyen();  
4     msn.nhap("Nhập mảng số nguyên");  
5     msn.xuat("Xuất mảng số nguyên");  
6     int s=msn.tong();  
7     System.out.println("Tổng mảng số nguyên: "+ s);  
8  
9 }
```



HỎI VÀ ĐÁP

OVERLOAD

Định nghĩa các phương thức overload

```
1 public class PhanSo{
2     private int tuSo;
3     private int mauSo;
4     . . .
5     public PhanSo cong(PhanSo ps){
6         PhanSo kq=new PhanSo();
7         kq.tuSo=this.tuSo*ps.mauSo+this.mauSo*ps.tuSo;
8         kq.mauSo=this.mauSo*ps.mauSo;
9         return kq;
10    }
11    public PhanSo cong(int k){
12        PhanSo kq=new PhanSo();
13        kq.tuSo=this.tuSo+this.mauSo*k;
14        kq.mauSo=this.mauSo;
15        return kq;
16    }
17 }
```

Định nghĩa các phương thức overload

```
1 package bt1;
2 public class Main {
3     public static void main(String[] args) {
4
5         PhanSo ps1=new PhanSo(2, 3);
6         PhanSo ps2=new PhanSo(3, 4);
7
8         PhanSo kq1=ps1.cong(ps2);
9         kq1.xuat("Tổng hai phân số ps1 và ps2");
10
11        PhanSo kq2=ps1.cong(40);
12        kq2.xuat("Tổng hai phân số ps1 và 40");
13
14    }
15 }
```

Định nghĩa các phương thức overload

```
1 package bt1;
2 public class MangPhanSo {
3     private PhanSo[] arr;
4     . . .
5     public PhanSo tong() {
6         PhanSo s=this.arr[0];
7         for(int i=1; i<this.arr.length; i++) {
8             s=s.cong(this.arr[i]);
9         }
10    }
11    public PhanSo tong(MangPhanSo mps) {
12        PhanSo s1=this.tong();
13        PhanSo s2=mps.tong();
14        PhanSo kq=s1.tong(s2);
15        return kq;
16    }
17 }
```

Định nghĩa các phương thức overload

```
1 public static void main(String[] args) {  
2  
3     MangPhanSo mps1=new MangPhanSo();  
4     mps1.nhap("Nhập mảng phân số");  
5     PhanSo s1=mps1.tong();  
6     s1.xuat("Tổng mảng phân số 1");  
7  
8     MangPhanSo mps2=new MangPhanSo();  
9     mps2.nhap("Nhập mảng phân số");  
10    PhanSo s2=mps1.tong(mps2);  
11    s2.xuat("Tổng mảng phân số 1 & 2");  
12 }
```

Định nghĩa các phương thức overload

```
1 package bt1;
2 public class MangSoNguyen {
3     private int[] arr;
4     . . .
5     public PhanSo tong() {
6         int s=0;
7         for(int i=0; i<this.arr.length; i++) {
8             s=s + this.arr[i];
9         }
10    }
11    public int tong(MangSoNguyen msn) {
12        int s1=this.tong();
13        int s2=msn.tong();
14        int kq=s1 + s2;
15        return kq;
16    }
17 }
```

Định nghĩa các phương thức overload

```
1 public static void main(String[] args) {
2
3     MangSoNguyen msn1=new MangSoNguyen();
4     msn1.nhap("Nhập mảng số nguyên 1");
5     int s1=msn1.tong();
6     System.out.println("Tổng mảng số nguyên 1" + s1);
7
8     MangSoNguyen msn2=new MangSoNguyen();
9     msn2.nhap("Nhập mảng số nguyên 2");
10    int s2=msn1.tong(msn2);
11    System.out.println("Tổng mảng số nguyên 1&2"+ s2);
12 }
```



HỎI VÀ ĐÁP

PHƯƠNG THỨC MỨC LỚP (STATIC)

Định nghĩa các phương thức mức lớp (static)

```
1 public class PhanSo{
2     private int tuSo;
3     private int mauSo;
4     . . .
5     public static PhanSo parsePhanSo(String str){
6         PhanSo kq=new PhanSo();
7         String [] s=str.split("/");
8         kq.tuSo=Integer.parseInt(s[0]);
9         kq.mauSo=Integer.parseInt(s[1]);
10        return kq;
11    }
12 }
13 }
```

Định nghĩa các phương thức mức lớp (static)

```
1 package bt1;
2 public class Main {
3     public static void main(String[] args) {
4
5         PhanSo ps1=PhanSo.parsePhanSo("2/3");
6         PhanSo ps2=PhanSo.parsePhanSo("3/4");
7
8         PhanSo kq1=ps1.cong(ps2);
9         kq1.xuat("Tổng hai phân số ps1 và ps2");
10
11        PhanSo kq2=ps1.cong(40);
12        kq2.xuat("Tổng hai phân số ps1 và 40");
13
14    }
15 }
```

Định nghĩa các phương thức mức lớp (static)

```
1 package bt1;
2 public class MangPhanSo {
3     private PhanSo[] arr;
4     . . .
5     public static MangPhanSo parseMangPhanSo(String str) {
6         MangPhanSo mps=new MangPhanSo();
7         String []s=str.split(";");
8         mps.arr=new PhanSo[s.length];
9         for(int i=0; i<s.length; i++){
10             mps.arr[i]=PhanSo.parsePhanSo(s[i].trim());
11         }
12         return mps;
13     }
14 }
```

Định nghĩa các phương thức mức lớp (static)

```
1 public static void main(String[] args) {  
2  
3     MangPhanSo mps1=MangPhanSo.parseMangPhanSo(  
4         "1/2;2/3;3/4");  
5     PhanSo s1=mps1.tong();  
6     s1.xuat("Tổng mảng phân số 1");  
7  
8     MangPhanSo mps2=MangPhanSo.parseMangPhanSo(  
9         "4/3;7/6;9/2");  
10    mps2.nhap("Nhập mảng phân số");  
11    PhanSo s2=mps1.tong(mps2);  
12    s2.xuat("Tổng mảng phân số 1 & 2");  
13 }
```

Định nghĩa các phương thức mức lớp (static)

```
1 package bt1;
2 public class MangSoNguyen {
3     private int[] arr;
4     . . .
5     public static MangSoNguyen parseMangSoNguyen(String str) {
6         MangSoNguyen msn=new MangSoNguyen();
7         String []s=str.split(";");
8         msn.arr=new int[s.length];
9         for(int i=0; i<s.length; i++){
10             msn.arr[i]=Integer.parseInt(s[i].trim());
11         }
12         return msn;
13     }
14 }
```

Định nghĩa các phương thức mức lớp (static)

```
1 public static void main(String[] args) {
2
3     MangSoNguyen msn1=MangPhanSo.parseMangSoNguyen (
4         "2;3;4");
5     int s1=msn1.tong();
6     System.out.println("Tổng mảng số nguyên 1:" + s1);
7
8     MangSoNguyen msn2=MangPhanSo.parseMangSoNguyen (
9         "4;7;9");
10    int s2=msn1.tong(msn2);
11    System.out.println("Tổng mảng số nguyên 1&2:"+s2);
12 }
```



HỎI VÀ ĐÁP

Tài liệu tham khảo

- Nguyễn Hoàng Anh, Tập slide bài giảng và video môn Lập Trình Java, ĐH KHTN, 2010
- The Java Language Specification Third Edition (2005)



HỎI VÀ ĐÁP