

BÀI TẬP JAVA NÂNG CAO

Thông tin liên hệ:

Website: <https://amzuni.com>

Facebook: <https://www.facebook.com/amzuni>

Youtube: <https://www.youtube.com/channel/UCBMgtQYuwvRBPiA297S23vA>

1. Lớp và đối tượng

Tạo project

Name Project: ClassAndObject

- **Package:** advance.dev

- **Class:** MainApp

Viết chương trình thực hiện các nhiệm vụ sau:

a) Tạo lớp Student lưu trữ các thông tin sau:

- Tên
- Tuổi
- Địa chỉ
- Số điện thoại
- Điểm trung bình

b) Trong Lớp MainApp:

Viết phương thức input() nhập mảng 10 sinh viên vào từ bàn phím, sử dụng lớp Scanner. *Hướng dẫn sử dụng lớp Scanner ở cuối trang*

- Viết phương thức print() in ra mảng danh sách 10 sinh viên đã nhập

Đưa project lên github

2. Hàm khởi tạo, getter, setter

Tạo project

Name Project: GetterSetter

- **Package:** advance.dev

- **Class:** MainApp

Viết chương trình thực hiện các nhiệm vụ sau:

a) Tạo lớp Student lưu trữ các thông tin sau:

- Tên
- Tuổi
- Địa chỉ
- Số điện thoại
- Điểm trung bình

Lưu ý: mức độ truy nhập các thuộc tính là private

b) Viết phương thức get và set dữ liệu cho các thuộc tính. Kiểm tra tính đúng đắn của dữ liệu

- Đối với thuộc tính tuổi phải là số nguyên lớn hơn 0.
- Đối với điểm trung bình phải là số thực lớn hơn bằng 0.

c) Trong Lớp MainApp:

Viết phương thức input() nhập mảng 10 sinh viên vào từ bàn phím, sử dụng lớp Scanner. *Hướng dẫn sử dụng lớp Scanner ở cuối trang*

- Viết phương thức print() in ra mảng danh sách 10 sinh viên đã nhập
- Viết hàm sắp xếp mảng sinh viên theo thứ tự điểm trung bình tăng dần

Đưa project lên github

3. Overloading, package

Tạo project

Name Project: OverLoading

- **Package:** advance.dev

- **Class:** MainApp

Viết các dòng code sau.

Đưa project lên github

4. Kế thừa, lớp cơ sở, inner class

Tạo project

Name Project: Inheritance

- **Package:** advance.dev

- **Class:** MainApp

Viết chương trình thực hiện các nhiệm vụ sau:

a) Tạo lớp cơ sở Person lưu trữ các thông tin sau:

- Tên
- Tuổi
- Số điện thoại

b) Tạo lớp Student kế thừa lớp Person lưu trữ các thông tin sau

- Điểm Toán
- Điểm Lý
- Điểm Hóa
- Mã sinh viên
- Lớp
- Viết hàm tính điểm trung bình: `diemTrungBinh()`

c) Tạo lớp Teacher kế thừa lớp Person lưu trữ các thông tin sau

- Mã giáo viên
- Hệ số lương
- Viết hàm tính lương `tinhLuong()`

d) Khai báo mảng 10 Person: `Person persons = new Person[10];`

- Viết hàm nhập `input(Person[] persons)` vào mảng persons gồm 4 Teacher và 6 Student
- Viết hàm xuất `print(Person[] persons)` ra mảng persons
- Viết hàm `findTeacher(Person[] persons)` tìm giáo viên nào có lương cao nhất
- Viết hàm tìm sinh viên nào có điểm trung bình cao nhất

Đưa project lên github

5. Đa hình, abstract class, final class

Name Project: Abstracts

- **Package:** advance.dev

- **Class:** MainApp

a) Xây dựng lớp cơ sở trừu tượng Shape chứa 2 phương thức trừu tượng

- chuVi() : tính chu vi của hình
- dienTich(): tính diện tích của hình

b) Xây dựng 3 lớp kế thừa lớp cơ sở Shape

Lớp Triangle (Tam giác):

- Lưu thông tin độ dài 3 cạnh a, b, c
- Thực thi 2 phương thức chuVi() và dienTich()

Lớp Circle (Hình tròn):

- Lưu thông tin bán kính r
- Thực thi 2 phương thức chuVi() và dienTich()

Lớp Rectangle (Hình chữ nhật):

- Lưu thông tin độ dài 2 cạnh a, b
- Thực thi 2 phương thức chuVi() và dienTich()

c) Trong class MainApp

- Khai báo mảng 10 phần tử Shape shapes = new Shape[10]
- Viết phương thức input(Shape[] shapes) nhập giá trị cho mảng shapes 3 hình tròn, 3 hình chữ nhật, 4 hình tam giác
- Viết phương thức print(Shape[] shapes) xuất ra các hình vừa nhập
- Viết phương thức tìm hình có diện tích lớn nhất
- Viết phương thức tìm diện tích lớn nhất của mỗi hình tròn, chữ nhật, tam giác

Đưa project lên github

6. Lớp Object

Name Project: Objects

- **Package:** advance.dev

- **Class:** MainApp

Viết chương trình thực hiện các yêu cầu sau

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

```
Object ten = sc.next();  
Object tuoi = sc.nextInt();  
Object diem = sc.nextDouble();
```

Tìm hiểu sự khác biệt giữa kiểu dữ liệu primitive và Object

Đưa project lên github

7. Interface

Name Project: Interfaces

- **Package:** advance.dev

- **Class:** MainApp

a) Xây dựng lớp cơ sở trừu tượng Shape chứa 2 phương thức trừu tượng

- chuVi() : tính chu vi của hình
- dienTich(): tính diện tích của hình

b) Xây dựng interface IShape chứa 2 phương thức

- input(Shape[] shapes)
- print(Shape[] shapes)

c) Xây dựng 3 lớp kế thừa (extends) lớp cơ sở Shape và thực thi (implements) lớp IShape

Lớp Triangle (Tam giác):

- Lưu thông tin độ dài 3 cạnh a, b, c
- Thực thi 2 phương thức chuVi() và dienTich()

Lớp Circle (Hình tròn):

- Lưu thông tin độ dài 3 cạnh a, b, c
- Thực thi 2 phương thức chuVi() và dienTich()

Lớp Rectangle (Hình chữ nhật):

- Lưu thông tin độ dài 3 cạnh a, b, c
- Thực thi 2 phương thức chuVi() và dienTich()

d) Trong class MainApp

- Khai báo mảng 10 phần tử Shape shapes = new Shape[10]

- Viết phương thức `input(Shape[] shapes)` nhập giá trị cho mảng shapes 3 hình tròn, 3 hình chữ nhật, 4 hình tam giác
- Viết phương thức `print(Shape[] shapes)` xuất ra các hình vừa nhập
- Viết phương thức tìm hình có diện tích lớn nhất

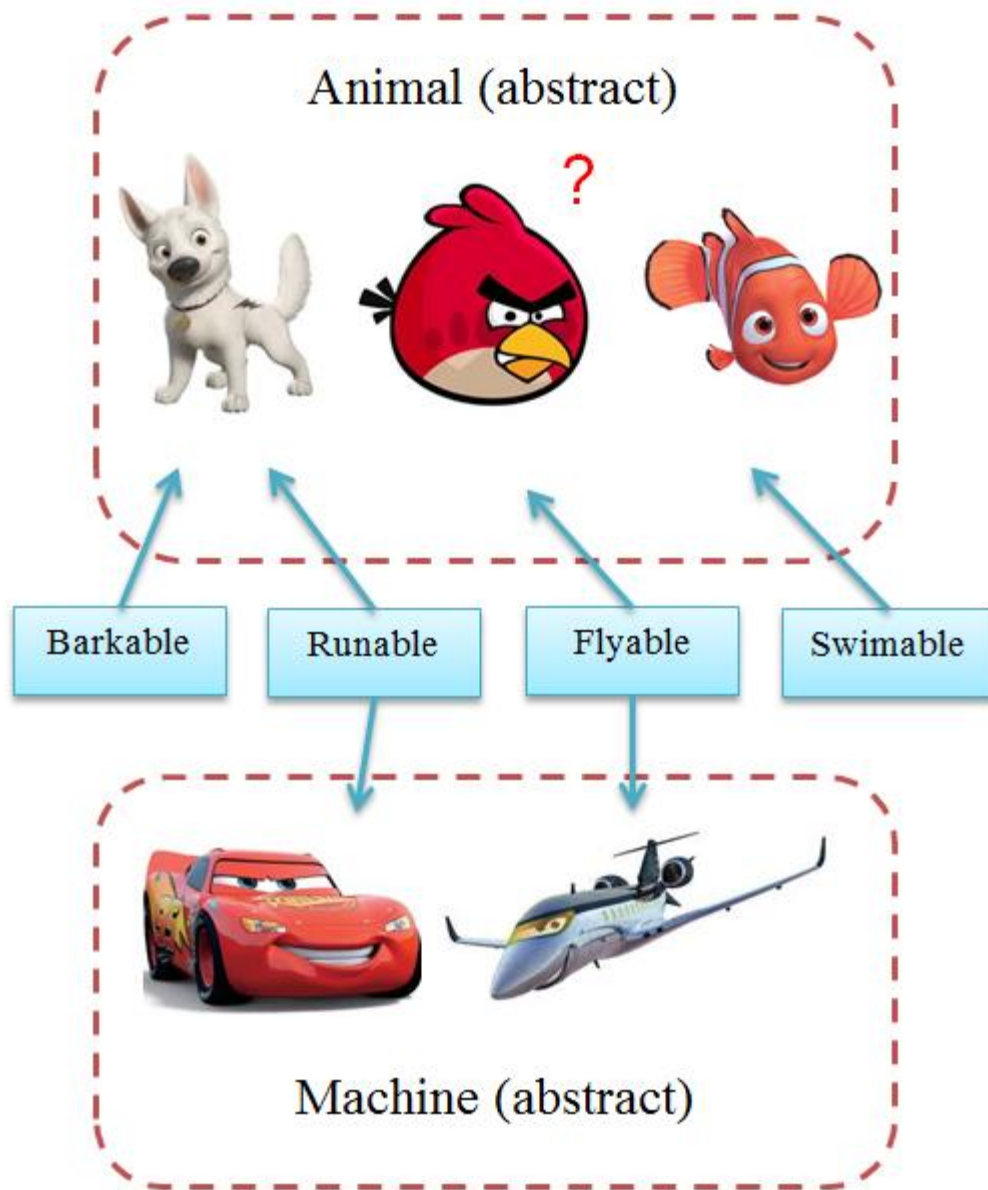
Viết phương thức tìm diện tích lớn nhất của mỗi hình tròn, chữ nhật, tam giác

Đưa project lên github

8. So sánh interface và abstract

Name Project: InterfaceAndAbstract

- **Package:** advance.dev
- **Class:** MainApp



a) Xây dựng các 2 lớp trừu tượng:

Lớp Animal với phương thức sau:

- eat()

Lớp Machine có phương thức:

- fuel() // nhiên liệu

b) Xây dựng lớp kế thừa lớp cơ sở Animal

- Lớp Dog
 - Lớp Fish
 - Lớp Bird
- c) Xây dựng lớp kế thừa lớp cơ sở Machine

- Lớp Car
- Lớp AirPlain

d) Xây dựng 3 interfaces

- Flyable có phương thức fly()
- Runnable có phương thức run();
- Swimable có phương thức swim();

e) Thực thi các interfaces

- Đối với Dog, Car thực thi (implements) lớp Runnable
- Đối với lớp Bird, AirPlan thực thi (implements) lớp Flyable
- Đối với lớp Fish thực thi lớp Swimable

Đưa project lên github

9. Sao chép, clone

Name Project: CopyClone

- **Package:** advance.dev

- **Class:** MainApp

Viết chương trình thực hiện các nhiệm vụ sau:

a) Tạo lớp Student lưu trữ các thông tin sau:

- Tên
- Tuổi
- Địa chỉ
- Số điện thoại
- Điểm trung bình

b) Lớp Student thực thi (implements) lớp Cloneable sao chép các thuộc tính của Student

c) Trong Lớp MainApp:

- Khởi tạo đối tượng Student std1. Sử dụng phương thức clone để sao chép đối tượng std1 sang Student std2. In ra màn hình phương thức toString() của 2 đối tượng std1 và std2.

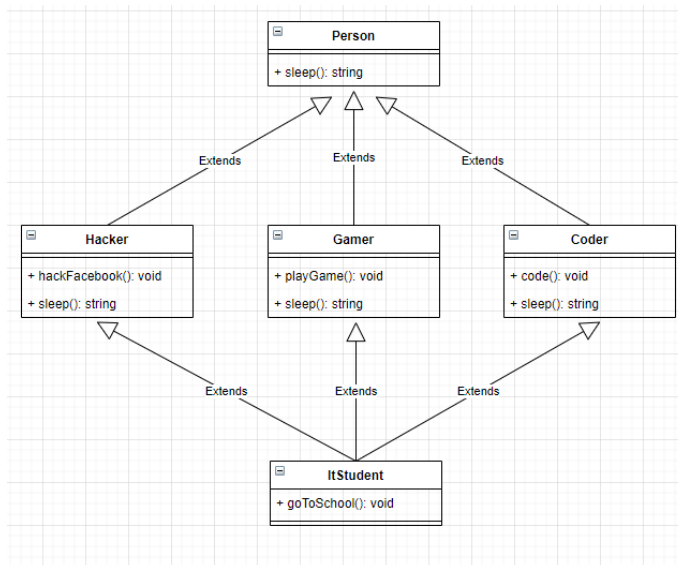
Đưa project lên github

10. Đa kế thừa interface

Name Project: InterfaceInheritance

- **Package:** advance.dev

- **Class:** MainApp



Tạo 3 lớp:

a) Tạo ba interface như sau:

```

public interface ICoder {
    String sleep(String time);
    void code();
}
public interface IGamer {
    String sleep(String time);
    void playGame();
}
public interface IHacker {
    String sleep(String time);
    void hackFacebook();
}
  
```

b) Lớp ItStudent sẽ implements cả ba interface đó:

```
public class ItStudent implements IHacker, IGamer, ICoder {  
    @Override  
    public void code() {  
    }  
  
    @Override  
    public void playGame() {  
    }  
  
    @Override  
    public void hackFacebook() {  
    }  
  
    @Override  
    public String sleep(String time) {  
        return time;  
    }  
}
```

c) Khởi tạo đối tượng ItStudent và gọi các phương thức của đối tượng ra trong phương thức main.

Đưa project lên github

11. Ngoại lệ, try catch

Name Project: TryCatch

- **Package:** advance.dev

- **Class:** MainApp

Viết chương trình thực hiện các nhiệm vụ sau

a) Nhập vào bàn phím hai số nguyên a, b. Khai báo số nguyên $c = a / b$. Sử dụng try catch để bắt lỗi trường hợp $b = 0$.

b) Nhập vào bàn phím một chuỗi a (String a). Sử dụng hàm Integer.parse(a) để chuyển chuỗi a sang số nguyên. Sử dụng try catch để bắt lỗi nhập vào không phải dạng số.

Đưa project lên github

12. Ném lỗi throw và throws

Name Project: ThrowAndThrows

- **Package:** advance.dev

- **Class:** MainApp

Viết chương trình thực hiện các nhiệm vụ sau

- Viết phương thức divide(int a, int b) trả về return a / b. Sử dụng throw và throws để ném lỗi. Gọi phương thức divide ở phương thức main
 - Viết phương thức parse(String a) sử dụng phương thức Integer.parse(a) để chuyển chuỗi a sang số nguyên. Sử dụng throw và throws để ném lỗi. Gọi phương thức parse trong phương thức main
- a, b là 2 số nguyên nhập vào bàn phím thông qua hàm Scanner. Hướng dẫn sử dụng lớp Scanner ở cuối trang*

Đưa project lên github

13. Tùy biến Exception

Name Project: Exceptions

- **Package:** advance.dev

- **Class:** MainApp

- Tạo lớp AgeException kế thừa từ lớp Exception.
 - Viết phương thức checkAge(int age) trong phương thức main. Ném lỗi AgeException nếu kiểm tra age <= 0
 - Gọi phương thức checkAge trong phương thức main.
- age nhập vào từ bàn phím thông qua lớp Scanner. Hướng dẫn sử dụng lớp Scanner ở cuối trang*

Đưa project lên github

14. Lớp Collections

Name Project: CollectionsClass

- **Package:** advance.dev

- **Class:** MainApp

Viết các dòng code sau.

Đưa project lên github

15.Iterator, Iterable, List, Queue, Set, Map

Name Project: CollectionsBase

- **Package:** advance.dev

- **Class:** MainApp

a) Xây dựng lớp cơ sở trừu tượng Shape chứa 2 phương thức trừu tượng

- chuVi() : tính chu vi của hình
- dienTich(): tính diện tích của hình

Thuộc tính

- name

b) Xây dựng interface IShape chứa 2 phương thức

- input(Shape[] shapes)
- print(Shape[] shapes)

c) Xây dựng 3 lớp kế thừa (extends) lớp cơ sở Shape và thực thi (implements) lớp IShape

Lớp Triangle (Tam giác):

- Lưu thông tin độ dài 3 cạnh a, b, c
- Thực thi 2 phương thức chuVi() và dienTich()

Lớp Circle (Hình tròn):

- Lưu thông tin độ dài 3 cạnh a, b, c
- Thực thi 2 phương thức chuVi() và dienTich()

Lớp Rectangle (Hình chữ nhật):

- Lưu thông tin độ dài 3 cạnh a, b, c
- Thực thi 2 phương thức chuVi() và dienTich()

d) Trong class MainApp

- Khai báo `List<Shape> shapes = new ArrayList<Shape>();`
- Viết phương thức `readFromFile(List<Shape> shapes)` nhập giá trị cho mảng từ tập tin ***input.txt***.

- Nội dung File *input.txt*

#circle1

3

#circle2

5

#circle3

6

7

8

#rectangle1

5

6

#rectangle2

6

9

#triangle1

3

3

6

#triangle2

6

8

9

Hướng dẫn đọc tập tin txt ở cuối trang

- Viết phương thức `print(List<Shape> shapes)` xuất ra các hình vừa nhập.
Sử dụng Iterator để duyệt
- Viết phương thức tìm hình có diện tích lớn nhất, sử dụng Iterator để duyệt
- Viết phương thức tìm diện tích lớn nhất của mỗi hình tròn, chữ nhật, tam giác. Sử dụng Iterator để duyệt

Đưa project lên github

16. LinkedList, ArrayList, Stack, Vector, PriorityQueue, HashMap

Name Project: CollectionsAdvance

- **Package:** advance.dev

- **Class:** MainApp

Thực hiện bài tập 15 thay ArrayList bằng LinkedList. Kiểm tra tốc độ xử lý với sự trợ giúp.

```
long begin = System.currentTimeMillis();  
//Code ...  
long end = System.currentTimeMillis();  
System.out.println(String.format("Thời gian xử lý: %d mili giây", end -  
begin));
```

Đưa project lên github

17. Lớp Arrays

Name Project: Arrays

- **Package:** advance.dev

- **Class:** MainApp

Đưa project lên github

18. Lớp Hashtable và Lớp Properties

Name Project: HashtableAndProperties

- **Package:** advance.dev

- **Class:** MainApp

Viết các dòng code sau.

Đưa project lên github

19. Lớp String, StringBuffer, StringTokenizer

Name Project: StringAdvance

- **Package:** advance.dev

- **Class:** MainApp

Viết các dòng code sau.

Đưa project lên github

20. Kỹ thuật Serialization, Deserialization

Name Project: Serialization

- **Package:** advance.dev

- **Class:** MainApp

Viết các dòng code sau.

Đưa project lên github

21. Threading và Multithreading

Name Project: Threading

- **Package:** advance.dev

- **Class:** MainApp

Viết các dòng code sau.

Đưa project lên github

THÔNG TIN BỔ SUNG:

1. Nhập thông tin từ bàn phím

Cách 1:

```
/* Sử dụng thư viện cần thiết */

import java.io.*;

/* Sử dụng phương thức System.in làm tham số cho phương thức khởi tạo
của InputStreamReader */

InputStreamReader isr = new InputStreamReader(System.in);

/* Sử dụng bộ đệm dữ liệu BufferedReader để lưu dữ liệu đọc */

BufferedReader br = new BufferedReader(isr);

/* có thể thực hiện được mà không cần lưu vào bộ đệm, khi đó bạn chỉ có
thể đọc một byte tại một thời điểm và vào một biến kiểu int, vì phương
thức read () trả về kiểu int:

InputStreamReader isr = new InputStreamReader(System.in);
int a = isr.read();

*/

try
{
    /* đọc một dòng từ bộ đệm; phương thức readLine () yêu cầu xử lý một
    lỗi có thể xảy ra khi nhập từ bảng điều khiển trong một khối thử */

    String line = br.readLine();

    //chuyển đổi một chuỗi thành một số

    int number = Integer.parseInt(line);
}

catch (NumberFormatException e) //KHÔNG phải là một số được
nhập, bạn không thể chuyển đổi thành kiểu int, bắt lỗi

{

    System.out.println("Không phải là số nguyên");

}

catch (IOException e)

{
```



```

        System.out.println("Lỗi nhập từ bàn phím");
    }
}

```

Cách 2:

```

package binh.dev;

import java.util.Scanner;

/**
 *
 * @author TranThanhBinh
 */

public class MainApp {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a = sc.nextInt();
        double b = sc.nextDouble();
        String str = sc.nextLine();
    }
}

```

2. Hiển thị giá trị số định dạng

```

/* Sử dụng thư viện cần thiết */
import java.text.*;

/* Sử dụng class, ví dụ định dạng 5 chữ số hàng thập phân */
NumberFormat formatter = NumberFormat.getNumberInstance();
formatter.setMaximumFractionDigits(5);
System.out.println(formatter.format(number));

```

3. Ví dụ sử dụng các phương thức toán học trong lớp Math

```

Math.pow(10, k);           //tính 10 mũ k
Math.exp(x);               //tính e mũ x
Math.sqrt(x);              //tính căn bậc 2 của x
Math.sin(x);               //tính sin x

```

4. Đọc tập tin text

```
try {
    FileInputStream fstream = new FileInputStream("input.txt");
    BufferedReader br = new BufferedReader(new InputStreamReader(fstream));

    String strLine;
    //Đọc từng dòng từ tập tin input.txt
    while((strLine = br.readLine()) != null) {
        System.out.println(strLine);
    }
} catch (Exception e) {
    // TODO: handle exception
}

```

Ví dụ: <https://github.com/binhdev/CollectionDemo>