

**Session 1+2****Exceptions and Assertions****&****java.lang Package****Phần I - Thực hiện trong 45 phút****1. Mục tiêu**

- Hiểu cấu trúc try-catch-finally, khởi tạo và sử dụng.
- Biết cách tạo một exception tùy chỉnh, cú pháp sử dụng throw.
- Hiểu về gói java.lang trong Java.
- Nắm khái niệm class Math và sử dụng các hàm trong đó.
- Nắm khái niệm class System và sử dụng các hàm trong đó.

**2. Thực hiện**

**Bài thực hành 1:** Viết một chương trình Java thực hành gọi một số hàm và biến trong class Math. Viết hàm tính toán diện tích hình tròn với các tham số và cách tính sử dụng Math

Bước 1: khởi tạo class tên là Bai1 trong package demo.jp2.lab01.phan1.

```
package demo.jp2.lab01.phan1;

/**
 *
 * @author minhvuvc
 */
public class Bai1 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

        //Bước 2: viết các dòng lệnh để kiểm thử các hàm trong class Math,
        //lưu ý đối với 1 số hàm nếu dùng với println sẽ có kết quả không
        //như ý muốn do việc tính toán làm tròn, sử dụng printf thay thế.
        System.out.println("random: " + Math.random());
        System.out.println("ceil: " + Math.ceil(Math.PI));
        System.out.println("floor: " + Math.floor(Math.PI));
        System.out.println("cbrt: " + Math.cbrt(-27));
        System.out.println("sqrt: " + Math.sqrt(4));
        System.out.printf("sin: %f", Math.sin(Math.PI));
    }
}
```

```

System.out.printf("\ncos: %f", Math.cos(Math.PI));
System.out.printf("\ntan: %f\n", Math.tan(Math.PI));
System.out.println("incrementExact: " + Math.incrementExact(6));
System.out.println("incrementExact (long): " + Math.incrementExact(9l));
System.out.println("negateExact: " + Math.negateExact(69));
System.out.println("nextDown: " + Math.nextDown(69));
System.out.println("toRadians: " + Math.toRadians(90) + " Because 1 PI = 180:
=> " + Math.PI / 2);
System.out.println("toDegrees: " + Math.toDegrees(Math.toRadians(90)));

//Bước 4: viết code gọi và sử dụng trong hàm main.
Bai1 bai1 = new Bai1();
bai1.dienTichHinhTron(2);
}

//Bước 3: viết hàm tính diện tích
private void dienTichHinhTron(float banKinh) {
    double dienTich = Math.pow(banKinh, 2) * Math.PI;
    System.out.println("Diện tích hình tròn là: " + dienTich);
}
}

```

Bước 2: viết các dòng lệnh để kiểm thử các hàm trong class Math, lưu ý đối với 1 số hàm nếu dùng với println sẽ có kết quả không như ý muốn do việc tính toán làm tròn, sử dụng printf thay thế.

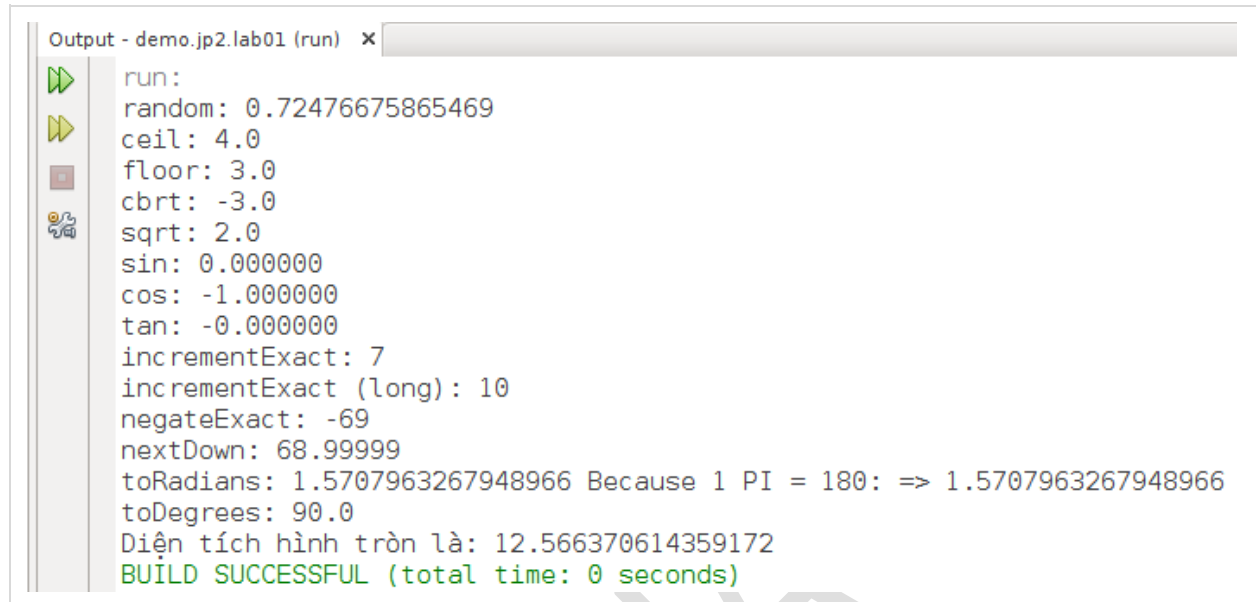
```

System.out.println("random: " + Math.random());
System.out.println("ceil: " + Math.ceil(Math.PI));
System.out.println("floor: " + Math.floor(Math.PI));
System.out.println("cbrt: " + Math.cbrt(-27));
System.out.println("sqrt: " + Math.sqrt(4));
System.out.printf("sin: %f", Math.sin(Math.PI));
System.out.printf("\ncos: %f", Math.cos(Math.PI));
System.out.printf("\ntan: %f\n", Math.tan(Math.PI));
System.out.println("incrementExact: " + Math.incrementExact(6));
System.out.println("incrementExact (long): " + Math.incrementExact(9l));
System.out.println("negateExact: " + Math.negateExact(69));
System.out.println("nextDown: " + Math.nextDown(69));
System.out.println("toRadians: " + Math.toRadians(90) + " Because 1 PI = 180:

```

```
=> " + Math.PI / 2);  
System.out.println("toDegrees: " + Math.toDegrees(Math.toRadians(90)));
```

Bước 5: run chương trình.



```
Output - demo.jp2.lab01 (run) x  
run:  
random: 0.72476675865469  
ceil: 4.0  
floor: 3.0  
cbrt: -3.0  
sqrt: 2.0  
sin: 0.000000  
cos: -1.000000  
tan: -0.000000  
incrementExact: 7  
incrementExact (long): 10  
negateExact: -69  
nextDown: 68.99999  
toRadians: 1.5707963267948966 Because 1 PI = 180: => 1.5707963267948966  
toDegrees: 90.0  
Diện tích hình tròn là: 12.566370614359172  
BUILD SUCCESSFUL (total time: 0 seconds)
```

**Bài thực hành 2:** Viết một chương trình Java cơ bản in ra màn hình thời gian hiện tại, copy dữ liệu từ mảng a sang mảng b và in dữ liệu mảng b, in ra màn hình tên của class hiện tại. Khai báo trong class 3 biến và in tên của các biến đó ra màn hình, in ra thông tin của class thông qua đối tượng Thông Tin lớp.

Tạo class " Bai2.java"

```
package demo.jp2.lab01.phan1;  
  
import java.lang.reflect.Field;  
import java.lang.reflect.Method;  
import java.text.Format;  
import java.text.SimpleDateFormat;  
import java.util.Calendar;  
import java.util.Date;  
  
/**  
 *  
 * @author minhvuvc  
 */  
public class Bai2 {  
  
    String tenSinhVien;  
    int tuoi;
```

```
float diemJava;

/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    Bai2 bai2 = new Bai2();
    //Bước 3: viết mã code in ra màn hình thời gian hiện tại.
    bai2.getTime();
    bai2.copyArray();

    //Bước 4: viết mã code in ra thông tin class.
    System.out.println("Tên class là: " + bai2.getClass().getName());
    Field arrField[] = bai2.getClass().getDeclaredFields();
    System.out.println("Bai2 có các biến: ");
    for (int i = 0; i < arrField.length; i++) {
        Field thuocTinh = arrField[i];
        System.out.println(thuocTinh.getName());
    }

    //Bước 5: viết mã code in ra danh sách hàm trong class.
    Method arrMethod[] = bai2.getClass().getDeclaredMethods();
    System.out.println("Bai2 có các hàm: ");
    for (int i = 0; i < arrMethod.length; i++) {
        Method thuocTinh = arrMethod[i];
        System.out.println(thuocTinh.getName());
    }
}

//Bước 2: viết mã code copy dữ liệu từ mảng a sang mảng b.
public void copyArray() {
    int arrSoNguyen[] = new int[]{1, 6, 5, 2, 8};
    int temp[] = new int[5];
    System.arraycopy(arrSoNguyen, 0, temp, 0, arrSoNguyen.length);
    for (int i = 0; i < temp.length; i++) {
        System.out.println("#" + temp[i]);
    }
}

//Bước 1: viết mã code in ra màn hình thời gian hiện tại.
public void getTime() {
    long time = System.currentTimeMillis();
    Date date = new Date(time);
    System.out.println("currentTimeMillis = " + time);
    Format format = new SimpleDateFormat("dd/MM/yyyy hh:mm:ss");
    System.out.println("Date = " + format.format(date));
}
```

```
}  
  
}
```

Bước 6: chạy thử chương trình.

```
Output - demo.jp2.lab01 (run)  
  
run:  
currentTimeMillis = 1436779352014  
Date = 13/07/2015 04:22:32  
#1  
#6  
#5  
#2  
#8  
Tên class là: demo.jp2.lab01.phan1.Bai2  
Bai2 có các biến:  
tenSinhVien  
tuoi  
diemJava  
Bai2 có các hàm:  
copyArray  
getTime  
main  
BUILD SUCCESSFUL (total time: 0 seconds)
```

## Phần II - Thực hiện trong 75 phút

### 1. Mục tiêu

- Biết sử dụng String, StringBuilder và StringBuffer. Một số hàm thường dùng trong lớp String.
- Hiểu biết cách sử dụng biểu thức quy tắc (Regular Expression) để xác thực dữ liệu, tìm kiếm mẫu chung, nắm được cú pháp xây dựng biểu thức quy tắc.

### 2. Thực hiện

**Bài thực hành 1:** Viết một chương trình nhập tên của sinh viên vào một đối tượng StringBuilder, phân cách nhau bởi dấu #, chuyển đổi đối tượng sang String sau đó phân tách chuỗi thành mảng, in danh sách mảng.

Bước 1: Viết class Bai1 trong package demo.jp2.lab01.phan2

```
package demo.jp2.lab01.phan2;  
  
import java.util.Scanner;  
  
/**  
 *
```

```
* @author minhvuvc
*/
public class Bai1 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Bai1 bai1 = new Bai1();
        bai1.nhapDuLieu();
    }

    //Viết hàm nhập dữ liệu, các chuỗi nhập sẽ lưu trữ vào biến StringBuilder.
    public void nhapDuLieu() {
        StringBuilder dsSinhVien = new StringBuilder();
        for (int i = 0; i < 3; i++) {
            System.out.println("Mời nhập sinh viên thứ " + (i + 1));
            Scanner nhap = new Scanner(System.in);
            String str = nhap.nextLine();
            dsSinhVien.append(str + "#");
        }

        //Bước 2: chuyển đổi biến StringBuilder về kiểu String và phân tách thông qua
        ký tự "#"
        System.out.println("Danh sach sinh vien");
        String temp = dsSinhVien.toString();
        String arrSV[] = temp.split("#");
        for (int i = 0; i < arrSV.length; i++) {
            System.out.println((i + 1) + " " + arrSV[i]);
        }
    }
}
```

Bước 3: Chạy chương trình

Output - demo.jp2.lab01 (run)

```
run:
Mời nhập sinh viên thứ 1
vu tuan minh
Mời nhập sinh viên thứ 2
ngoc trinh
Mời nhập sinh viên thứ 3
thuy top
1 vu tuan minh
2 ngoc trinh
3 thuy top
BUILD SUCCESSFUL (total time: 16 seconds)
```

**Bài thực hành 2:** Viết một chương trình gồm 1 chuỗi chứa một số ký tự "nhạy cảm", sử dụng kiến thức về RegExp để thay thế thành "\*beep\*".

Bước 1: Viết class có hàm main tên Bai2

```
package demo.jp2.lab01.phan2;

import java.util.regex.Matcher;
import java.util.regex.Pattern;

/**
 *
 * @author minhvuvc
 */
public class Bai2 {
    private static String REGEX = "xxx";
    private static String INPUT = "xxx là một chuỗi ký tự, khi cần thể hiện tình"
        + " cảm thì viết xxx ở cuối thư";
    private static String REPLACE = "*beep*";

    public static void main(String[] args) {
        Pattern p = Pattern.compile(REGEX);
        // get a matcher object
        Matcher m = p.matcher(INPUT);
        INPUT = m.replaceAll(REPLACE);
        System.out.println(INPUT);
    }
}
```

Bước 2: chạy chương trình

Output - demo.jp2.lab01 (run)

```
run:
*beep* là một chuỗi ký tự, khi cần thể hiện tình cảm thì viết *beep* ở cuối thư
BUILD SUCCESSFUL (total time: 0 seconds)
```

**Bài thực hành 3:** Viết một chương trình xác thực tên lớp viết theo chuẩn ở Aptech. Ví dụ tên lớp hợp lệ như sau: C1506M.

Bước 1: viết class tên là Bai3 có hàm main

```
package demo.jp2.lab01.phan2;

import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

/**
 *
 * @author minhvuvc
 */
public class Bai3 {

    //Bước 2: Khai báo chuỗi regexp
    String regExp = "^C[0-9]{4}[GHIKML]$";

    //Bước 3: Viết hàm validate
    public void validateName() {
        do {
            System.out.println("Nhập tên của lớp");
            Scanner nhap = new Scanner(System.in);
            Pattern pattern = Pattern.compile(regExp);
            Matcher matcher = pattern.matcher(nhap.nextLine());
            if (matcher.matches()) {
                System.out.println("Tên hợp lệ");
                break;
            } else {
                System.out.println("Tên sai, nhập lại!");
            }
        } while (true);
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        //Bước 4: Gọi vào hàm main
    }
}
```



```
Bai3 bai3 = new Bai3();
bai3.validateName();
}
}
```

Bước 5: Chạy chương trình

```
Output - demo.jp2.lab01 (run)
run:
Nhập tên của lớp
cádf
Tên sai, nhập lại!
Nhập tên của lớp
c23222g
Tên sai, nhập lại!
Nhập tên của lớp
c1506m
Tên sai, nhập lại!
Nhập tên của lớp
C1506M
Tên hợp lệ
BUILD SUCCESSFUL (total time: 21 seconds)
```

### Phần III - Bài tập tự làm

1. Tính giá trị của các biểu thức sau với x được nhập từ bàn phím :

Bài 1:

$$2\sin 2x - 3\cos 2x + 5\sin x \cdot \cos x - 2$$

Bài 2:

$$\frac{\cos x - 2\sin x \cdot \cos x}{2\cos^2 x + \sin x - 1}$$

Bài 3:

$$\sin^2 x + \sin 2x - 2\cos^x$$

Bài 4:

$$4\sin^3 x + 3\sin^2 \cdot \cos x - \sin x - \cos^3 x$$

2. Làm bài tập theo yêu cầu sau:

Tạo class **Student** gồm các thuộc tính:

- rollNo (String - Gồm 8 ký tự bắt đầu phải là chữ C - vd: C1312001)
- fullname (String - Tối đa 64 ký tự)
- deskphone (String - Định dạng mã vùng 3 ký tự + dấu '.' + tối thiểu 6 và tối đa 7 số - vd: 043.667788).

- cellphone (String - Định dạng gồm toàn số, tối thiểu 10 số và tối đa 11 số).
- email
- yearOfBirth (int - nhập năm sinh - tối thiểu là 1970 và tối đa là 1996).
- `compile("^19([7-8][0-9])|(9[0-6])$");` thêm giải thích

Viết các hàm get, set cho lớp, implement lớp trên với interface Serializable và override hàm toString in ra thông tin của sinh viên.

Tạo mảng (dùng kiến thức Array hoặc Collection - List, ArrayList) để có thể lưu trữ dữ liệu sinh viên.

Tạo MainClass, tạo menu, tạo hàm cho phép nhập dữ liệu cho mảng Student ở trên.

Yêu cầu nhập dữ liệu phải bắt lỗi nhập liệu (theo RegExp).

Gợi ý: chuỗi xác thực email có thể như sau:

```
private static final String EMAIL_PATTERN =
    "^[_A-Za-z0-9-\\+]+(\\.[_A-Za-z0-9-]+)*@"
    "+[A-Za-z0-9-]+(\\.[A-Za-z0-9-]+)*(\\.[A-Za-z]{2,})$";
```

Học viên chủ động tự viết chuỗi xác thực, tập thử với các cú pháp, mẫu đã học trong giáo trình để hiểu bài học hơn. Tham khảo thêm ý nghĩa RegExp ở [trang sau](#).

3. Làm bài tập theo yêu cầu sau:

Viết chương trình nhập vào ngày tháng năm sinh của bạn theo cú pháp: *mm-dd-yyyy* hoặc *mm/dd/yyyy*

Xác thực dữ liệu nhập vào có đúng là kiểu ngày tháng năm sinh theo định dạng yêu cầu hay không?

Tham khảo [tại đây](#).

4. Làm bài tập sau:

Viết chương trình xác thực dữ liệu nhập vào là 1 số điện thoại bất kỳ (có 9 đến 11 số).

5. Làm bài tập theo yêu cầu:

Một số điện thoại cố định của Hà Nội có dạng sau:

043xxxxxx

Trong đó x là 1 chữ số bất kỳ.

Viết chương trình nhập vào 1 số điện thoại và xác thực xem đó có phải là số điện thoại cố định ở Hà Nội hay không.

6. Làm bài tập theo yêu cầu:

Dùng Class Math sinh số ngẫu nhiên trong khoảng 0-90

Ứng với số ngẫu nhiên N sinh ra hiển thị kết quả :

$$A = B.\sin(N) + \cos(N)/C$$

Trong đó :

B : Tổng không gian nhớ của hệ thống tính theo MB

C : Tổng không gian nhớ trống của hệ thống tính theo MB