**+ Hằng số ( Bắt đầu bằng final)**

Câu lệnh: Final Kiểu dữ liệu = giá trị;

Vidu: final int x=100;

**+ Cách nhập dữ liệu từ bàn phím**

Cách khai báo: Public static void man(String[] args){

Scanner scaner =new Scanner(String.in) }

Thêm next trước các kiểu dữ liệu ( nextInt)

**+ Chuyển đổi kiểu dữ liệu**

* Kiểu ngầm định (gán cho 1 biến của biến khác )

Int x =100;

Ép kiểu sang float : float y=x;

* Kiểu tường minh ( độ chính xác ) int b=(int) a+5

Float x=8,7f; ( thêm f là float)

Ép kiểu sang Int y= (int) e;

**+ Các phép toán một ngôi ( + ,-,++,--,!)**

* + cộng một ngôi
* - trừ một ngôi
* ++ tăng giá trị của một biến lên 1
* -- giảm giá trị của một biến lên 1
* ! nghịch đảo true/fail ( sai thành đúng, đúng thành sai)
* Nếu đặt trước biến sẽ tăng/giảm lên rồi mới thực hiện đoạn code đó
* Nếu đặt sau biên sẽ tăng/giảm sau khi thực hiện đoạn code
* Phep a:5
* Phep ++a: 6
* Phep a++: 6
* Phep a:7
* Phep --a: 6
* Phep a--: 6
* Phep a:5

**+ Các phép so sánh (==, !=,>,<,>=,<=, && và,||hoặc)**

System.***out***.println( a+"="+b+":"+(a!=b));

System.***out***.println( a+"="+b+":"+(a>=b));

System.***out***.println( a+"="+b+":"+(a>b));

System.***out***.println( a+"="+b+":"+(a<b));

System.***out***.println( "Ca hai so la so chan"+(a%2==0 && b%2==0));

System.***out***.println( "Ca its nhaat 1 so chan"+(a%2==0 || b%2==0));

**+ Toán tử điều kiện**

(bloolean-expression)?if-true-return-this: if-false-return-this;

String ketqua=(a%2==0)? ”So chan”:”Sole”;

**+ Lớp Math và các hàm toán học**

* .e,pi
* Abs: tính giá trị tuyệt đối
* Ceil: tìm ra giá trị nguyên nhỏ nhất
* Pow: hàm số mũ lũy thừ
* Random: con số ngẫy nhiên
* Round: làm tròn
* Sqrt: Căn bậc 2
* Floor: lấy hàm nhỏ nhất
* System.***out***.println(" Gia tri tuyet doi"+ Math.*abs*(a));
* //Tính chu vi
* chuvi = 2\*Math.***PI***\*r;
* System.***out***.println(" Dien tich chu vi la:"+ Math.*round*(chuvi));
* dientich = Math.***PI*** \* Math.*pow*(r,2);//(Math.PI \* r\*r) cx đc
* System.***out***.println(" Dien tich la:"+ Math.*round*(dientich));

**+ Mảng trong lập trình có 2 cách**

Kiểu dữ liệu [] biến; Double[] mang1

Kiểu dữ liệu biến[]; String mang2[];

**public** **class** hihu {

**public** **static** **void** main(String[] args) {

Double[] mang1;

mang1= **new** Double[5];( Khởi tạo số lượng 5 phần tử / Điều kiện chỉ cho nhập 10 ký tự)

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

**for** (**int** i=0;i< mang1.length; i++) { // mang1.length lấy ra độ dài của mảng

System.out.println(" Nhap vao " + i + "=");

mang1[i]= sc.nextDouble();

}

**double** tong =0;

**for** (**int** i = 0; i < mang1.length; i++) {

tong +=mang1[i];

}

System.out.println(" Tong ="+ tong);

String mang2[];

mang2= **new** String[] {"Thanh","Thi"," Hai"};// Nêu ra các ký tự nhập vào []

// không cần nhập từ bàn phím

**for** ( **int** i=0; i<mang2.length;i++) {

System.***out***.println(" Gia trị la" + mang2[i] );// mang2[i] i là vị trí phần tử trong mảng bắt đầu từ 0

}

}

}

**+ Lệnh Try catch**

**try** {

System.***out***.println(" Nhap vao so nguyen");// Nhập đúng số nguyên thì sẽ chạy đúng

n=sc.nextInt();

} **catch** (Exception e) { // Nếu nhập không phải số nguyên thì sẽ hiện thông báo này

System.***out***.println(" Nhap dữ liệu không đúng");

} **finally**{// có lỗi hay không có lỗi cũng xuất ra đoạn này - Câu lệnh này có haowjc k

System.***out***.println("Mặc định");

}

System.***out***.println(" Gia trị nhap la"+ n);

**+ Xây dựng lớp và phương thức khỏi tạo Constructor**

* Các thuộc tính (biến)
* Các hành vi (hàm)

**+ Bài tập về Hàm**

**public** **class** TTTTTBB {// Viết hàm viết ở trong Class

// hàm trả về kết quả, và hàm không trả về kết quả

// Public , protected, private:

// Tên hàm - Tên kiểu dữ liệu - Tên biến ( kiểu dữ liệu a, Kiểu dữ liệu b)[public int tinhtong (int a, int b)]

**public** **int** tinhtong (**int** a, **int** b) {// hàm tính tong

**return** a+b;

}

**public** **int** tinhhieu (**int** a, **int** b) {// hàm tính hieu

**return** a+b;

}

**public** **static** **void** main(String[] args) {

**int** a=1;

**int** b=20;

TTTTTBB m = **new** TTTTTBB(); // Phải tạo đối tượng m là cửa để

//để truy cập vào từng hàm bên trong của lớp class TTTTTBB

**int** sum =m.tinhtong(a,b);

**int** sub =m.tinhhieu(a,b);

System.***out***.println(" Sum " + sum);

System.***out***.println(" Sub "+ sub);

}

}

**+ Bài tập 2 veef Hàm**

**public** **class** CheckInput {

// Viết 1 hàm trả vê gtri int nếu mà người dùng nhập chú thích sai yêu cầu nhập

// lại

**public** **int** Check() {

**int** number = 0;

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

**while** (**true**) {

**try** {

System.***out***.println(" Đưa ra");

number = Integer.*parseInt*(sc.nextLine());

**break**;

} **catch** (Exception e) {

System.***out***.println(" Đưa ra sai gia tri");

}

}

**return** number;

}

**public** **int** Checkpostsitive() {

**int** number = 0;

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

**while** (**true**) {

**try** {

System.***out***.println(" Đưa ra postsitive");

number = Integer.*parseInt*(sc.nextLine());

**if**(number>0)

**break**;

} **catch** (Exception e) {

System.***out***.println(" Đưa ra Đung gia tri");

}

}

**return** number;

}

**public** **int** Checknagative() {

**int** number = 0;

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

**while** (**true**) {

**try** {

System.***out***.println(" Đưa ra nagative");

number = Integer.*parseInt*(sc.nextLine());

**if**(number<0)

**break**;

} **catch** (Exception e) {

System.***out***.println(" Đưa ra Đung gia tri");

}

}

**return** number;

}

**public** **int** tinhtong() {

**int** a = Checkpostsitive();

**int** b= Checkpostsitive();

**return** a+b;

}

**public** **static** **void** main(String[] args) {

CheckInput m = **new** CheckInput();

**int** number = m.Check();

System.***out***.println(" Number" + number);

**int** Post = m.Checkpostsitive();

System.***out***.println("Post Number" + number);

**int** Navi = m.Checknagative();

System.***out***.println("Navi Number" + number);

System.***out***.println("Navi Number" + m.tinhtong());

}

}

**+ Mảng thường**

**public** **class** Array\_mang {

**public** **static** **void** main(String[] args) {

// array binh thuong

// arrayList

**int**[] arr= {1,2,4,5,6,9,17};

//for i

**for** (**int** i = 0; i <= arr.length-1; i++) {

System.***out***.println( arr[i]);

}

//Tính tổng các phần tử trong mảng

// System.out.println("\_\_\_\_\_");

//

// int Sum=0;

// for (int i = 0;i < arr.length-1; i++) {

// Sum = Sum+arr[i];

//

// }

// System.out.println(" In ra" +Sum );

// for object

System.***out***.println("\_\_\_\_\_\_\_");

**for**(**int** item : arr) { // item là gọi ra từng giá trị trong mảng arr

System.***out***.println(item);

}

}

}

**+ Mảng ArrayList**

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.Iterator;

**public** **class** Array\_list {

**public** **static** **void** main(String[] args) {

// Tên hàm ArrayList

ArrayList<Integer> arr =**new** ArrayList<>(); // giống cú pháp Scanner: Nhưng k phải là nhaatpj từ bản phím)

// Thêm phần tử vào trong arr

arr.add(10);

arr.add(5);

arr.add(3);

arr.add(15);

arr.add(50);

arr.add(-30);

//arr = {10,5,3,15,50,-30}

// Kiểu for i

**for** (**int** i=0; i< arr.size(); i++) {

System.***out***.println(arr.get(i));// (arr.get(i) dùng cho Arraylist

}

System.***out***.println("\_\_\_\_\_");

//Kiểu for object

**for** (Integer a : arr) {

System.***out***.println( a);

}

// Hàm Collections là hàm xắp xếp

Collections.*sort*(arr); //Sắp xếp tăng dần

**for** (Integer o : arr) {

System.***out***.println( o);

}

}

}

**+ Biến local** được khai báo trong phương thức hoặc hàm

Public void getInfo()

{

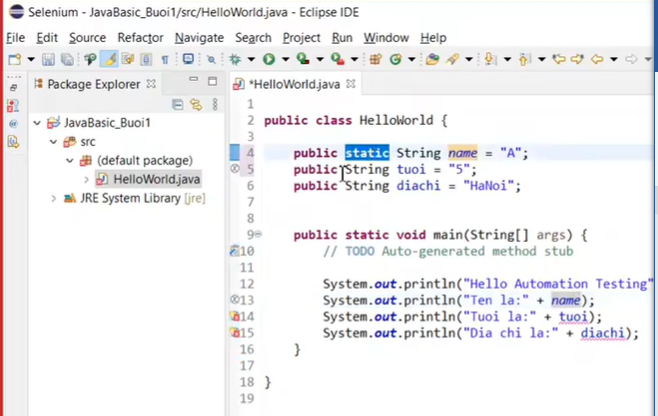
Int a =10 // Gọi là biến local

}

**+ Biến toàn cục** nằm trong Class nằm ngoài hàm

Public String name = “ Tuoi” // Biến toàn cục

**+ Biến static** chỉ được gọi những biến static



+ Tìm 1 đoạn seacrh tìm ra những bài k nằm trong từ khóa search đó

* Element\_value : 1 đối trượng trên trang web (lấy Xpath)
* getText() : Lấy đoạn test tương ứng với element\_value
* String Mess=element\_value.getText()

**+ Khởi tạo mảng một chiều**

Datatype <tên mảng>[] = new datatype[size];

Kiểu dữ liệu <tên mảng>[] = new kiểu dữ liệu[độ dài của mảng là số nguyên]

* Phải cùng Kiểu dữ liệu

Cách 1:

* **int** a[] = {12,33,4,53};
* **for**( **int** i : a) {
* // i là vị trí của mảng đó
* // i : a là lấy ra in ra từng vị trí của mảng đó
* // không giới hạn từ đâu tới đâu mà in ra tất cả
* System.***out***.println(i);

Cách 2:

**int** a[] = **new** **int**[3];

a[0]=10;

a[1]=14;

a[2]=15;

**for** (**int** i = 0; i < a.length; i++) {

System.***out***.println(a[i]);

}

**+ Protected:** được truy cập bên trong package và ngoài package phải kế thừa

**+ Private :** Chỉ truy cập trong 1 lớp

**+ Tính đóng gói:**

**+ Kế thừa Extends**

* **class** Bike {
* **void** run() {
* System.***out***.println("Chạy");
* }
* }
* **public** **class** motorbike **extends** Bike {
* **void** run() {
* System.***out***.println("Chạy khoảng 40km/giờ");
* }
* **public** **static** **void** main(String[] args) {
* // **TODO** Auto-generated method stub
* Bike bk = **new** motorbike();
* System.***out***.println("Ví dụ về tính đa hình:");
* bk.run();
* }
* }

**+ Nạp Chồng**

* **package** Buoi1;
* **class** adder{
* // Nạp chồng tạo ra nhiều 3 Tên trùng, Nhưng khác nhau đối số hoặc kiểu dữ liệu đó
* **static** **int** add (**int** a, **int** b) {
* **return** a+b;
* }
* **static** **int** add (**int** a, **int** b, **int** c) {
* **return** a+b+c;
* }
* **static** String add (String a, String b,String c) {
* **return** a+b+c;
* }
* }
* **public** **class** Napchong {
* **public** **static** **void** main(String[] args) {
* System.***out***.println(" Vi du nap chồng");
* System.***out***.println(adder.*add*(11, 11));
* System.***out***.println(adder.*add*(11, 11, 11));
* System.***out***.println(adder.*add*("Le", "Thi", "Hai"));
* }

**+ Cách xác định Locator của Element**

**findElement()** và **findElements()** được cung cấp bởi lớp WebDriver và WebElement.

* **findElement()**: trả về một đối tượng WebElement
* **findElements()**: trả về một danh sách các đối tượng thuộc WebElement
* **Thuộc tính**: id, name, classname, linktext, partialLinkText ( là đường link có thể rút gọn tên), CssSelector,tagname
* **Css Selectors:** color, background, background-image, type của input, playhoder
* **Xpath:** Tương đối (//), tuyệt đối(/)
* **HTML**:a, h1-h6, div,span,section,header,body,img,i,b,hr,
* **package** Buoi1;
* **import** org.openqa.selenium.By;
* **import** org.openqa.selenium.WebDriver;
* **import** org.openqa.selenium.WebElement;
* **import** org.openqa.selenium.chrome.ChromeDriver;
* **public** **class** Locator {
* **public** **static** **void** main(String[] args) **throws** InterruptedException {
* System.*setProperty*("webdriver.chrome.driver", "D:\\Java\\chromedriver.exe");
* WebDriver driver = **new** ChromeDriver();
* //1 - Maximize browser của mình
* driver.manage().window().maximize();
* //2 - Đi đến 1 url
* driver.navigate().to("https://anhtester.com");
* System.***out***.println(driver.getTitle());

* // Đối tượng của WebElement
* // By.id là tìm theo id
* // button, input, link, dropdown, dialog,header\_page
* WebElement button\_Login= driver.findElement(By.*id*("btn-login"));
* button\_Login.click();
* Thread.*sleep*(2000);
* WebElement email= driver.findElement(By.*name*("email"));
* //sendKeys nhập giá trị
* email.sendKeys("haimoms2001@gmail.com");
* Thread.*sleep*(2000);
* WebElement password= driver.findElement(By.*name*("password"));
* password.sendKeys("Haile@2001");
* Thread.*sleep*(2000);
* //Cách 1
* WebElement button\_Loginxacnhan= driver.findElement(By.*id*("login"));
* button\_Loginxacnhan.click();
* // Css
* //a[id='btn\_dangbai']: a là thẻ css : [id='btn\_dangbai'] thuộc tính
* // Cách 2
* driver.findElement(By.*cssSelector*("a[id='btn\_dangbai']")).click();
* //a là thẻ css, @ đại diện cho thuộc tính , id=’btn-dangbai’ thuộc tính
* driver.findElement(By.*xpath*("//a[@id=’btn-dangbai’]")).click();
* // Chạy chờ khoảng 2s thì tắt
* Thread.*sleep*(2000);
* // Chạy song thì đóngThoát hẳn Browser
* driver.quit();
* }
* }