**Phần I: Trắc Nghiệm (6đ 0.3 mỗi câu)**

**Câu 1:** Cho array **int**[] num = {2,3,9,0}; Khi đó num[2] sẽ có giá trị là:

1. **2 B. 3 C. 9 D. 0**

**Câu 2: public** **static** **void** main(String[] args) {

**int**[][] matrix = **new** **int**[10][10];

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

**for**(**int** j=0;j<matrix[i].length;j++) {

matrix[i][j] = i + j;

}

}

System.***out***.println(matrix[6][8]);

}

Đoạn code trên sẽ cho kết quả là:

1. **16 B. 14 C. 12 D. 48**

**Câu 3:** Cho regex"(FB|CD|DE|\\d)+" . String nào sẽ khớp với regex trên ?

1. **“**-4” **B. “** cd”  **C. “**FBDE\\d” **D. “**07CDDE”

**Câu 4:** Kết quả của phép tính

(Math.*pow*(8, 2)>Math.*pow*(3, 4))&&(Math.*pow*(232, 345)>Math.*pow*(137, 372))

Là:

1. **true B. false C. 0 D. 1**

**Câu 5: public** **static** **void** main(String[] args) {

**boolean** b = **true**;

**if**(b) {

System.***out***.print("Hello ");

b = **false**;

}

**else** **if**(!b) {

System.***out***.print("World");

}

}

Đoạn code trên sẽ cho kết quả là:

1. Hello  **B.** World

**C.** Hello World  **D.** Hello

World

**Câu 6: int** t = 12&40; biến t sẽ có giá trị là:

1. 52 **B.** 8 **C.** 12  **D.** 480

**Câu 7:** Cho một class sau:

**class** Adder{

Adder(){}

Adder(**int** i,**int** j){

**this**.i = i;

**this**.j = j;

}

**private** **int** i;

**private** **int** j;

**int** add() {

**return** i+j;

}

**int** add(**int** i, **int** j) {

**return** i+j;

}

}

Phương thức main:

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

System.***out***.println(**new** Adder(156,328).add());

System.***out***.println(**new** Adder().add(149,237));

}

Nếu ta chạy phương thức main thì sẽ cho ra kết quả là:

1. 484 **B.** 386

484484

**C.** 386  **D.** 484

386 386

**Câu 8:** Mệnh đề nào là KHÔNG đúng với constructor ?

1. Tất cả các constructor trong một class luôn cùng tên với nhau.
2. Constructor không thể là final
3. Constructor không thể là protected
4. Có thể không khởi tạo bất kỳ constructor nào trong một class.

**Câu 9:** Đoạn codeSystem.***out***.println("".compareTo("ProgrammingIsFun")); sẽ cho ra kết quả là:

1. 15  **B.** -16

**C.** -15 **D.** false

**Câu 10: import** java.util.ArrayList;

**import** java.util.Collections;

**public** **class** Test{

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

ArrayList<String> english = **new** ArrayList<>();

english.add("David");

english.add("Nick");

english.add("Zoe");

english.add("John");

ArrayList<String> math = **new** ArrayList<>();

math.add("Anna");

math.add("Jonathan");

math.add("Nick");

math.add("Robby");

System.***out***.println(Collections.*disjoint*(english, math));

}

}

Chương trình trên sẽ cho ra kết quả là:

1. **6 B. 1 C. false D. true**

**Câu 11:** Cho các class sau:

**class** C1{

**void** act() {

say();

}

**void** say(){

System.***out***.print(2);

}

}

**class** C2 **extends** C1{

**void** act() {

System.***out***.print(1);

**super**.act();

}

**void** say(){

System.***out***.print(3);

}

}

Phương thức main:

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

C2 c = **new** C2();

c.act();

}

Nếu ta chạy phương thức main thì sẽ cho ra kết quả là:

1. **123 B. 12 C. 132 D.13**

**Câu 12:** Cho đoạn code sau:

**public** **class** Test{

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

Book book1 = **new** Book("Harry Potter",400);

*changer*(book1);

System.***out***.printf("The book title is %s and the book page number is %d",book1.title,book1.pageNum);

}

**public** **static** **void** changer(Book b) {

b.title = "The Red Hair";

b.pageNum = 200;

}

}

**class** Book{

**public** Book(String title, **int** pageNum) {

**this**.title = title;

**this**.pageNum = pageNum;

}

String title;

**int** pageNum;

}

Đoạn code trên sẽ cho kết quả là:

1. The book title is The Red Hair and the book page number is 400
2. The book title is Harry Potter and the book page number is 200
3. The book title is The Red Hair and the book page number is 200
4. The book title is Harry Potter and the book page number is 400

**Câu 13:** Cho đoạn code sau:

**public** **class** Test{

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

MyFunction act = (**int** i)->{**return** i/2;};

MyFunction act2 = (**int** i)->{**return** i/5;};

System.***out***.print(act.act(19)+act2.act(79));

}

}

**interface** MyFunction{

**public** **int** act(**int** i);

}

Đoạn code trên sẽ cho kết quả là:

1. **24 B. 25.3 C. 25 D.Exception xảy ra**

**Câu 14:** Mệnh đề nào là KHÔNG đúng với interface ?

1. Các phương thức trong interface không thể là protected.
2. Ta có thể implements nhiều interface tại một class.
3. Interface không có constructor.
4. Interface có thể extends một class nào đó.

**Câu 15: Cho các class và interface như sau:**

**interface** I{

**default** **void** show() {

System.***out***.println("Hello");

}

}

**class** A{

**public** **void** show() {

System.***out***.println("World");

}

}

**class** C **extends** A **implements** I {}

Phương thức main:

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

C c = **new** C();

c.show();

}

Nếu ta chạy phương thức main thì sẽ cho ra kết quả là:

1. Hello **B.** Hello

World

**C.** World **D.** Exception xảy ra

**Câu 16:** Nếu ta chạy phương thức main thì sẽ cho ra kết quả là:

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

String s1 = "Java";

String s2 = "Java";

String s3 = **new** String("Java");

String s4 = **new** String("Java");

System.***out***.println(s1==s2);

System.***out***.println(s2==s3);

System.***out***.println(s3==s4);

}

1. **false B. true**

**true false**

**true false**

**C. false D. true**

**false true**

**false true**

**Câu 17: Cho một class sau:**

**class** Bird{

**public** **void** sound(){}

**static** **class** Dove **extends** Bird{

**protected** **void** sound(){System.***out***.println("coo coo");}

}

}

Nếu đoạn code trên có lỗi compile-time xảy ra thì compiler sẽ gạch dưới ở đâu ?

1. **sound() của class Bird B. sound() của class Dove**
2. **protected hay public D. Không có lỗi xảy ra**

**Câu 18: Cho đoạn code sau:**

**public** **class** Test{

@SuppressWarnings("static-access")

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

C1 c = **new** C2();

c.*m1*();

}

}

**class** C1{

**static** **void** m1() {

System.***out***.println("Hello");

}

}

**class** C2 **extends** C1{

**static** **void** m1() {

System.***out***.println("World");

}

}

Đoạn code trên sẽ cho kết quả là:

**A. Hello B. World**

**C. Không có gì xảy ra D. Exception xảy ra**

**Câu 19:**

**import** java.io.File;

**import** java.util.Scanner;

**public** **class** Test{

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

@SuppressWarnings("resource")

Scanner input = **new** Scanner(**new** File("abc.txt"));

}

}

Biết rằng có tồn tại file abc.txt. Nếu đoạn code trên có lỗi compile-time xảy ra thì compiler sẽ gạch dưới ở đâu ?

1. Cảinput và **new** Scanner(**new** File("abc.txt"))
2. **new** Scanner(**new** File("abc.txt"))

**C.** input

**D.** Không có lỗi nào xảy ra.

**Câu 20: Cho đoạn code sau:**

**public** **class** Test{

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

**try** {

*p*();

System.***out***.println("Programming");

}

**catch**(Exception e){

System.***out***.println(e.getLocalizedMessage());

}

}

**static** **void** m() {**throw** **new** NullPointerException("Java");}

**static** **void** n() {*m*();}

**static** **void** p() {*n*();}

}

Đoạn code trên sẽ cho kết quả là:

**A. Java B. Programming**

**Programming Java**

**C. Java D. Programming**

**Phần II: Tự luận (4đ)**

**Câu 1: Hãy nêu định nghĩa về DAO design pattern ? (1đ)**

**Câu 2: Hãy vẽ sơ đồ UML về Strategy design pattern ? (1đ)**

**Câu 3: Trên bảng ta có các số 1/80, 2/80, 3/80, …, 80/80. Ta được phép xóa hai số a, b ngẫu nhiên rồi sau đó thêm vào số a+b-2ab lên trên bảng. Thực hiện quá trình này đến khi trên bảng chỉ còn đúng một số X. Hãy tìm ra số X đó bằng lập trình hoặc bằng toán học. (2đ)**