

BÀI TẬP TRẮC NGHIỆM OOP

PHẦN 1.

Câu 1. Một lớp trong Java có thể có bao nhiêu lớp cha?

- A. 1
- B. 3
- C. 4
- D. 2

Câu 2. Khai báo lớp nào dưới đây là đúng?

- A. `public class default { }`
- B. `protected inner class engine { }`
- C. `final class outer { }`
- D. A, B, C đều sai

Ans: D

Câu 3. Cho lệnh `Circle x = new Circle()`. Phát biểu nào sau đây là chính xác nhất

- A. x chứa 1 tham chiếu tới 1 đối tượng Circle
- B. x chứa 1 giá trị nguyên
- C. x chứa 1 đối tượng của kiểu Circle
- D. Gán giá trị của Circle cho x

Câu 4. Giả list là một đối tượng kiểu ArrayList. Để hiển thị các phần tử trong list, ta sử dụng cú pháp nào sau đây:

- A. `System.out.println(list);`
- B. `Iterator itr = list.iterator(); while (itr.hasNext()) { System.out.print(itr.next() + ", "); }`
- C. `for (String str : list) System.out.println(list);`
- D. Các cách trên đều đúng.

Ans: D

Câu 5. Tính đa hình trong java là:

- A. Đối tượng của lớp con có thể tham chiếu đến đối tượng của lớp cha
- B. Xây dựng một lớp mới từ 1 lớp khác
- C. Đối tượng của lớp cha có thể tham chiếu đến đối tượng của lớp con
- D. Một lớp có thể chứa lớp khác

Câu 6. là một cấu trúc để xác định các đối tượng cùng loại.

- A. Class (lớp)

- B. Object (đối tượng)
- C. Method (phương thức)
- D. Data fields (thuộc tính)

Câu 7. Phương thức next () của lớp Scanner dùng để làm gì?

- A. Đọc 1 chuỗi (1 từ)
- B. Đọc 1 ký tự
- C. Đọc 1 dòng
- D. Đọc 1 số nguyên

Câu 8. Ngôn ngữ Java được phát triển bởi

- A. James Gosling và nhóm phát triển tại Sun Microsystems
- B. Mark Zuckerberg và nhóm phát triển tại Facebook
- C. Steve Jobs và nhóm phát triển tại Apple
- D. Bill Gates và nhóm phát triển tại Microsoft

Ans A

Câu 9. Một đối tượng là một thể hiện của

- A. Lớp
- B. Chương trình
- C. Phương thức
- D. Dữ liệu

Câu 10 Đối tượng lắng nghe (Listener) đối với TextField là

- A. TextListener
- B. ItemListener
- C. ActionListener
- D. Không có đối tượng nào ở trên

Câu 11 Phương thức dùng để gán cho nội dung Label là

- A. setText(String)
- B. setTextLabel("noi dung")
- C. setLabelText("noi dung")
- D. setLabel("noi dung")

Câu 12 Chọn phát biểu ĐÚNG về mảng (array)

- A. Biểu thức array.length được sử dụng để xác định số phần tử trong mảng
- B. Mảng có thể lưu trữ các phần tử thuộc nhiều kiểu dữ liệu khác nhau
- C. Chỉ số của mảng có thể sử dụng kiểu số thực (float, double)
- D. Các phần tử của mảng không thể là tham số của phương thức

Câu 13 Java cung cấp một số từ khóa để làm thay đổi dòng điều khiển dựa trên các điều kiện. Vậy từ khóa nào làm dừng hoạt động vòng lặp trong cùng chứa nó và thực hiện lệnh tiếp ngay sau vòng lặp đó

- A. stop
- B. halt
- C. exit
- D. break

ans: D

Câu 14 Từ khóa dùng để định nghĩa một lớp mới là?

- A. method
- B. object
- C. new
- D. class

Câu 15 interface là gì?

- A. Một interface là một tập hợp các hằng số và các phương thức trừu tượng.
- B. Một interface là một lớp mà một lớp con có thể mở rộng.
- C. Một interface là một tập hợp các thành phần GUI.
- D. Một interface là một tập hợp các phương thức có phạm vi truy xuất public.

Ans: A

Câu 16 Đặc trưng hướng đối tượng nào cho phép bạn xây dựng các lớp mới từ các lớp hiện có.

- A. Tính kế thừa
- B. Tính đa hình
- C. Tính đóng gói
- D. Tính trừu tượng

Câu 17 Định nghĩa lớp trừu tượng nào sau đây là hợp lệ?

- A. abstract class A { abstract void unfinished(); }
- B. class A { abstract void unfinished() { } }
- C. class A { abstract void unfinished(); }
- D. class A { abstract void unfinished(); }

Câu 18 Để tạo một label có nhãn text, bạn dùng lệnh nào dưới đây?

- A. new Labelled();
- B. new Label();
- C. new Label(text);

D. new Labelled(text);

ans: C

Câu 19 Để đọc 1 file văn bản (.txt), ta sử dụng luồng nào sau đây?

- A. output stream
- B. error stream
- C. file stream
- D. input stream

Ans: D

Câu 20 Phương thức final có thể bị ghi đè trong lớp con?

- A. Có thể
 - B. Chỉ khi có từ khóa super
 - C. Chỉ khi có từ khóa override
 - D. Không
-

PHẦN 2. XỬ LÝ NGOẠI LỆ

Question 1. Which of these keywords is not a part of exception handling?

- A. try
- B. finally
- C. thrown
- D. catch

ans: C

Question 2.

```
public class Foo
{
    public static void main(String[] args)
    {
        try
        {
            return;
        }
        finally
        {
            System.out.println( "Finally" );
        }
    }
}
```

- A. Finally
- B. Compilation fails.
- C. The code runs with no output
- D. An exception is thrown at runtime.

Ans: A

Question 3

What will be the output of the program?

```
try
{
    int x = 0;
    int y = 5 / x;
}
catch (Exception e)
{
    System.out.println("Exception");
}
catch (ArithmeticException ae)
```

```
{  
System.out.println(" Arithmetic Exception");  
}  
System.out.println("finished");
```

- A. finished
- B. runtime error.
- C. Compilation fails
- D. Exception.

Question 4

```
public class X  
{  
public static void main(String [] args)  
{  
try  
{  
badMethod();  
System.out.print("A");  
}  
catch (Exception ex)  
{  
System.out.print("B");  
}  
finally  
{  
System.out.print("C");  
}  
System.out.print("D");  
}  
public static void badMethod()  
{  
throw new Error(); /* Line 22 */  
}  
}
```

- A. ABCD
- B. Compilation fails.
- C. C is printed before exiting with an error message.
- D. BC is printed before exiting with an error message

Question 5.

```
public class X  
{  
public static void main(String [] args)  
{  
try
```

```

{
badMethod();
System.out.print("A");
}
catch (RuntimeException ex) /* Line 10 */
{
System.out.print("B");
}
catch (Exception ex1)
{
System.out.print("C");
}
finally
{
System.out.print("D");
}
System.out.print("E");
}
public static void badMethod()
{
throw new RuntimeException();
}
}

```

- A. BD
- B. BDE
- C. BD
- D. DE

Ans: B

Question 6. java.lang.NullPointerException is a

- A. Error
- B. runtime exception
- C. Compile time exception
- D. None

Ans: B

Question 7. FileNotFoundException

- A. Is a subclass/extends IOException
- B. Is a Compile time exception
- C. Found in java.io package
- D. All

Question 8. Exception and Error are direct subclasses of?

- A. .BaseException
- B. Throwable
- C. Object
- D. RuntimeException

Question 9. What block is always executed, independently of a exception being raised?

- A. throws
- B. finally
- C. catch
- D. throw

ans: B

Question 10. What keyword is used to explicitly raise a exception?

- A. catch
- B. throw
- C. throws
- D. raise

ans: B

Question 11 Which of these class is highest in hierarchy in java

- A. java.lang.Exception
- B. java.lang.Error
- C. java.lang.Throwable
- D. java.lang.Object

Question 12 Exception is found in which package in java

- A. java.lang
- B. java.util
- C. java.io
- D. .java

Question 13. Exception is a class/interface/abstract class/other?

- A. Class
- B. Interface
- C. Abstract class
- D. Other

Question 14


```
class Main {  
public static void main(String args[]) {  
try {  
throw 10;  
}  
catch(int e) {  
System.out.println("Got the Exception " + e);  
}  
}  
}
```

- A. Got the Exception 10
- B. Got the Exception 0
- C. Compiler Error
- D. none

ans: C

Question 15 Both class Error and class Exception are children of this parent:

- A. Throwable
- B. Catchable
- C. Runnable
- D. Problem

Ans: A

Question 16 What type of exception is thrown by parseInt() if it gets illegal data?

- A. ArithmeticException
- B. RuntimeException
- C. NumberFormatException
- D. NumberErro

Question 17 The code within the ----- block will always be executed whether or not an exception is thrown.

- A. try
- B. catch
- C. finally
- D. throw

Question 18 Which exception is thrown by read() method?

- A. IOException
- B. InterruptedException
- C. SystemException

D. SystemInputException

Question 19. From which problems is it possible for a program to recover?

- A. Errors
- B. Exceptions
- C. Both errors and exceptions
- D. Neither.

Ans: B

Question 20. class exception_handling

```
{ public static void main(String args[])
{ try
{ int a, b;
b = 0;
a = 5 / b;
System.out.print("A");
} catch(ArithmeticException e)
{ System.out.print("B");
}
finally
{ System.out.print("C");
}
}
}
```

- A. A
- B. B
- C. AC
- D. BC

PHẦN 3.

DESIGN PATTERN QUIZ

Question 1. Which text below describes the Template Method pattern best?

- A. Encapsulates interchangeable behaviours and uses delegation to decide which one to use.
- B. Simplifies the interface of a group of classes.
- C. Subclasses decide how to implement steps in an algorithm.
- D. Changes the interface of one or more classes.
- E. Allows (a group of) objects to be notified when some state changes.

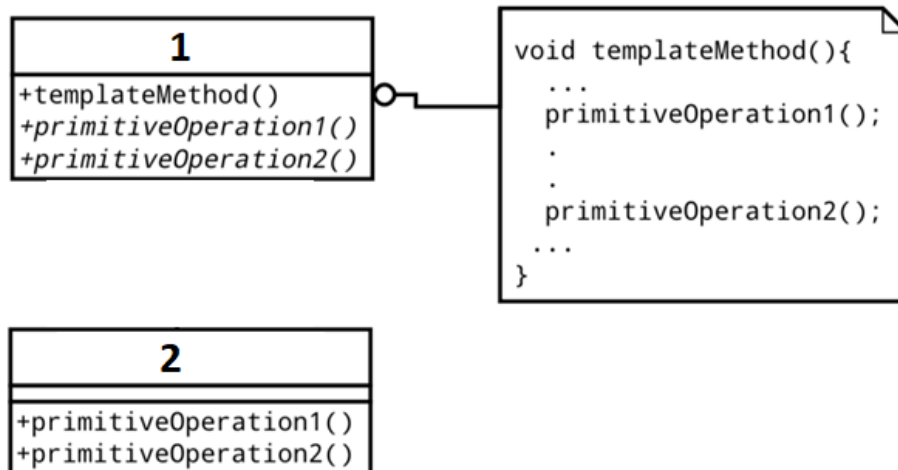
Ans: C

Question 2. What is not a correct statement for the Hollywood Principle?

- A. High-level components control when and how low-level components participate in the computation.
- B. It is a way to prevent dependency rot.
- C. A low-level component never calls a high-level component directly.
- D. Call us, we don't call you.

Ans: D

Question 3. Have a look at the class diagram for the Template Method pattern. What are the appropriate class names?



- A. 1. ConcreteClass 2. AbstractClass
- B. 1. AbstractClass 2. ConcreteClass

Ans: B

Question 4. Which text below describes the Strategy Pattern best?

- A. Encapsulates interchangeable behaviours and uses delegation to decide which one to use.
- B. Simplifies the interface of a group of classes.
- C. Provides a way to traverse a collection of objects without exposing the collection's implementation.
- D. Changes the interface of one or more classes.
- E. Client treats collections of objects and individual objects uniformly.

Question 5. Identify the aspects of your application that vary and separate them from what stays the same. What does that mean?

- A. Define a super class for what varies and let what stays the same extend that super class.
- B. Take what varies and "encapsulate" it so it won't affect the rest of the code.
- C. Put what varies and what stays the same together. Do it so that starting with an abstract class, all other classes inherit in a clever way.
- D. The number of classes should be identical with the number of interfaces.

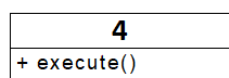
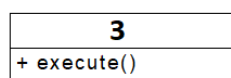
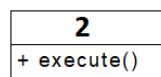
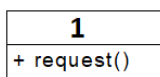
Question 6. What is a proper design principle?

- A. Use abstract classes to implement what varies.
- B. Put what varies and what stays the same together. Do it so that starting with an abstract class, all other classes inherit in a clever way.
- C. Program to an interface, not an implementation.
- D. The number of classes should be identical with the number of interfaces.

Question 7. Favour composition over inheritance. What does that mean?

- A. HAS-A can be better than IS-A.
- B. IS-A can be better than HAS-A.
- C. Do not use abstract classes.
- D. The number of classes should be identical with the number of interfaces.

Question 8. Have a look at the class diagram for the Strategy pattern. What are the appropriate names for the classes?

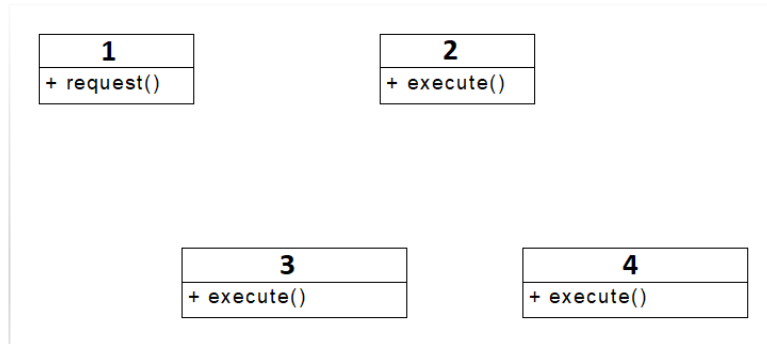


- A. 1. Context 2. Strategy 3. ConcreteStrategyB 4. ConcreteStrategyA

B. 1. Strategy 2. StrategyInterface 3. StrategyInterfaceImplA 4. StrategyInterfaceImplB

Ans: A

Question 9. Have a look at the class diagram for the strategy pattern. What is the relation between class 1 and class 2 based on the methods and the previous question?



- A. Class 2 is-a class 1.
- B. Class 2 has-a class 1.
- C. Class 1 has-a class 2.
- D. Class 1 is-a class 2.

Ans: C

Question 10. Which text below describes the Observer Pattern best?

- A. Allows (a group of) objects to be notified when some state changes.
- B. Provides a way to traverse a collection of objects without exposing the collection's implementation.
- C. Changes the interface of one or more classes.
- D. Simplifies the interface of a group of classes.
- E. Encapsulates interchangeable behaviours and uses delegation to decide which one to use.

Question 11. Which statement is correct?

- A. The Observer Pattern defines a many-to-many relationship.
- B. The Observer Pattern defines a one-to-many relationship.
- C. The Observer Pattern defines a one-to-one relationship.
- D. The Observer Pattern defines a many-to-one relationship.

Question 12. Regarding the Observer Pattern, which statement is NOT correct?

- A. We can add new observers at any time.
- B. We never need to modify the subject to add new types of observers.
- C. We can reuse subjects or observers independently of each other.
- D. Changes to either the subject or an observer will always affect the other

Question 13. Loosely coupled designs can handle change because

- A. they minimize the interfaces between objects.
- B. they minimize the interactions between objects.
- C. they minimize the interdependency between objects.
- D. they minimize the inheritance between objects.

Question 14. Which text below describes the Abstract Factory Pattern best?

- A. It changes the interface of one or more classes.
- B. It encapsulates interchangeable behaviours and uses delegation to decide which one to use.
- C. It simplifies the interface of a group of classes.
- D. It provides a way to traverse a collection of objects without exposing the collection's implementation.
- E. It provides an interface for creating families of related or dependent objects without specifying their concrete classes.

Question 15. Which text below describes the Factory Method Pattern best?

- A. Encapsulates interchangeable behaviours and uses delegation to decide which one to use.
- B. Separates the construction of a complex object from its representation.
- C. Subclasses decide which concrete classes to create.
- D. Changes the interface of one or more classes.
- E. Simplifies the interface of a group of classes.

Ans: B

Question 16. What statement is wrong?

- A. A factory method is abstract.
- B. A factory method handles object creation and encapsulates it in a subclass.
- C. A factory method informs the client what concrete product is actually created.
- D. A factory method returns a product that is typically used within methods defined in the superclass.
- E. A factory method might be parametrized (or not).

Ans: C

Question 17. What statement about the Dependency Inversion Principle is correct?

- A. Inverse the class inheritance via interfaces.
- B. Have abstract classes implement interfaces.
- C. Depend upon abstractions.
- D. Do depend upon concrete classes.

Question 18. What is the primary purpose of the Observer design pattern?

- A. To define a family of algorithms and make them interchangeable.
- B. To provide an interface for creating families of related or dependent objects.
- C. To define a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.
- D. To ensure a class has only one instance and provide a global point to access it.

Question 19. Which behavioral design pattern defines a way to encapsulate a request as an object, thereby allowing for parameterization of clients with different requests, queuing of requests, and logging of the parameters?

- A. Observer Pattern
- B. Strategy Pattern
- C. Command Pattern
- D. State Pattern

Question 20. What problem does the Chain of Responsibility pattern address?

- A. It provides an interface for creating families of related or dependent objects without specifying their concrete classes.
- B. It defines a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.
- C. It avoids coupling the sender of a request to its receiver by giving more than one object a chance to handle the request.
- D. It defines a family of algorithms, encapsulates each algorithm, and makes the algorithms interchangeable.

Question 21. Which behavioral design pattern allows an object to alter its behavior when its internal state changes, and the object will appear to change its class?

- A. Observer Pattern
- B. State Pattern
- C. Command Pattern
- D. Visitor Pattern

Ans: B

=====