SUSTech Mutual Aid Class

2022 Autumn
Java A Sample Exam

Question Setter: **Zacky Yang(杨宗奇)**

Verifier: **Yuhang Lin(林雨航)**

Publish Date: **2023/01/16**

This exam paper contains 5 parts and the score is 100 in total.

Part I. True or False(10 × 1 point = 10 points)

Please determine whether each statement is true(T) or false(F) and answer T/F accordingly.

1. Following the von Neumann architecture, modern computers consists of six logical units, one of which is Arithmetic and logic unit.

2. Java is both compiled and interpreted.

3. A program with only logic errors(wrong semantics) can go through compilation and runtime successfully, but the result are unexpected.

4. The symbol "!" can be used to invert logical values in java.

5. An Array a's length can be accessed by $a.length()$, and it cannot dynamically adjust its length.

6. Abstract class can have instance variables, but it cannot have constructors.

7. Short-circuit evaluation may occur when using operator & or |.

8. The protected fields of a class can be accessed only by its subclass.

9. Enums cannot have instance variables and constructors with arguments.

10. A class can both extend a class and implement interfaces, while an interface can only extend other interfaces.

Part II. Multiple-Choice Question(15×2 points $= 30$ points)

Note: Only one choice is correct

1. What package is the class *Object* in?

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

2. What is the relationship among the following?($A \subseteq B$ means B contains A)

A. JDK $\subseteq$ JRE $\subseteq$ JVM B. JRE $\subseteq$ JDK $\subseteq$ JVM

C. JVM $\subseteq$ JRE $\subseteq$ JDK D. JVM $\subseteq$ JDK $\subseteq$ JRE

3. What is the output of the following code?

```
int x=4;
switch (x)
{
    default: x+=x;
    case 1: x*=x;break;
    case 2: x-=x;
    case 3: x/=x;break;
}
System.out.println(x);
```

A. 64 B. 32

C. 4 D. Compile Error

4. What is the result of the following code?

```
int t=-1,k=5,total=120;
for(int i=1;i<=5;i++);
    t=t+1;
while(k>t){
    total/=--k;
}
System.out.println(total);
```

A. 5

B. 20

C. 120

D. Runtime Error

E. Compile Error

5. What are the values of *a.length* and *a*[1]*.length* respectively after executing the following code?

```java
int[][] a=new int[3][3];
int[][] b={{1,2,3},{4},{5},{6,7}};
int[] c={1,2};
b[1]=c;
a=b;
```

A. 3,3

B. 4,3

C. 4,1

D. 4,2

E. Compile Error

6. Which of the following operators has the lowest precedence?

A. /

B. =

C. +

D. %

7. What is the output of the following code?

```java
int a=7;
double x=2.8,y=4.7;
System.out.println( x+a%3*(int)(x+y)%2 );
```

A. 2.8

B. 3.8

C. 4

D. Compile Error

8. Which of the following can execute *test.class* when typing it in the command line?

A. javac test

B. javac test.class

C. java test

D. java test.class

9. Which of the following statements is **false** about enum types?

A. They extend Enum class, and have a common method compareTo.

B. They can only have private constructors.

C. They can be inherited by other classes.

D. The enum can only have finite number of objects.

10. Which of the statements is **true** about abstract classes?

A. Abstract classes cannot have constructors because they cannot be instantiated.

B. Abstract classes cannot be used to declare variables.

C. A class that extends an abstract class must be able to be instantiated.

D. Abstract class can have static methods and non-static concrete methods.

11. Which of the following is **true** about java interface?

A. Interfaces can have constructors, like abstract classes.

B. It is unnecessary to use keyword **abstract** to specify a method as an abstract method in interface.

C. A class can implement many interfaces, but an interface can only extend a single interface.

D. Interfaces cannot have private methods, but it can have protected methods.

12. which of the following about keyword **final** is **false**?

A. Final classes cannot be inherited.

B. Abstract methods cannot be final.

C. Final instance variables can be assigned in a static method.

D. Methods in final class are naturely final methods.

13. Which of the following is valid?(We have known *Fruit* is the superclass of *Apple* and *Peach*)

A. Object a=new Apple(); Fruit b=(Object)a; Apple c=(Apple) b;

B. Object a=new Peach(); Fruit b=(Fruit)a; Apple c=(Apple) b;

C. Object a=new Peach(); Fruit b=(Peach)a; Peach c=(Peach)b;

D. Object a=new Fruit(); Peach b=(Peach)a; Fruit c=b;

14. What is the output of the following code?

```
public class A
{
    public int x=1;
    public int getX() {return x;}
    public void increaseX() {x+=2;}
}
class B extends A
{
    public int x=2;
    public int getX() {return x;}
    public void increaseX() {x+=3;}
    public static void main(String[] args)
    {
        A a=new A();
        B b=new B();
        System.out.printf("%d ",a.getX());
        a=b;
        b.increaseX();
        System.out.printf("%d ",a.x);
        System.out.print(a.getX());
    }
}
```

A. 1 5 1   B. 1 1 5   C. 1 5 5   D. 2 5 5

15. Which of the following assignment is valid?

A. List<Object> list1=new ArrayList<>();

ArrayList<Object> list2=list1;

B. ArrayList<Object> list1=new ArrayList<>();

List<Integer> list2=list1;

C. List<Integer> list1=new ArrayList<>();

List<Object> list2=list1;

D. List<Object> list1=new ArrayList<>();

List<Object> list2=list1;

Part III. Completion by Matching(15 × 1 point = 15 points)

We provide a list of possible answers, which contains correct ones. Please choose from them, **one word can only use once**.

Abstract, Assert, Break, Case, Class, Continue, Constructor, Default, Do, Encapsulation, Enum, Extends, Final, Finally, For, Heap, If-else, immutability , Implements, Import, Inheritance, Instance, Instanceof, Interface, Iteration, Local, New, Object, Package, Polymorphism, Private, Protected, Public Recursion, Return, Runtime, Semantic, Stack, Static, Super, Syntax, This, Throw, Try-catch, Void, While

1. ____________________ , inheritance and polymorphism are three main characteristics of Object-Oriented-Programming.

2. When we describe the inheritance of two interfaces in java code, we use keyword ____________ .

3. Variables declared inside the class but outside the methods are called ______________ variables.

4. ______________ is an operator used to ensure an object is an instance of a class or its subclass.

5. Keyword _______ is used to call a constructor to create an object.

6. _________ methods can be called with both class name and instance name.

7. We have two ways to do repetitive things in java, one is recursion, another is _
___________ .

8. Keyword _________ can call constructors or methods of superclass version.

9. ______________ fields can be accessed both by its subclasses and classes within the same package.

10. Keyword __________ is used to acquire access to other packages, and it should be written at the top of your code.

11. Keyword __________ is used to terminate methods and offer a value as a result of some methods.

12. When __________ error happens, we cannot go through the compilation successfully.

13. _________ statement is used to create branches in our algorithm.

14. _________________ of String means that String objects cannot be changed after they are created.

15. The values of primitive type variables are stored in ________ in memory space.

Part IV. Short Answer Questions(5 × 3 points = 15 points)

1. Please write down at least **9** operators in java language, including but not limited to arithmetic operators, logical operators, etc.

2. Please write down what type of class member uses dynamic binding, and describe how to access subclass objects' private instance variables using a superclass variable(All mentioned classes have the same declaration of instance variables, i.e. same modifier, type, and name).

3. Please select all valid variable names below.

QAQ, 100_java, string, $_$abc, xyz@163, int

4. List all the instance variables declared in the following that can be accessed by class C?(A, B, C are in the same packages)

```
class A
{
    protected int x;
    public int y;
    private int z;
}
class B extends A
{
    int p;
    public int q;
    private int r;
}
class C extends B
{
    private int g;
}
```

5. There is a generic method aims to compare two variables of the same type. However, there is a single bug with the following code. Please (1) write down the result of the following code, (2) explain what is wrong with the code, and (3) give a solution to the problem.

```
public class Test
{
    public static void main(String[] args)
    {
        Integer a=new Integer(1);
        Integer b=2;
        comparation(a,b);
    }
    public static <T> void comparation(T a,T b)
    {
            System.out.println(a.compareTo(b));
    }
}
```

Part V. Programming(3×10 points $= 30$ points)

1. [**Serpentine Matrix**] Given a series of $n \times m$ integer numbers, our task is to fill a $n \times m$ matrix. The filling rule is starting at the top left corner of the matrix, go downward to fill the column, then go rightward to fill the row, then upward, then leftward, if meeting boundary or filled grid, change the direction. Repeat the steps until the matrix is filled. Here is an example.

$n = 3, m = 3, \quad series \quad of \quad numbers : 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9$

The filled matrix:

$$\begin{matrix} 1 & 8 & 7 \\ 2 & 9 & 6 \\ 3 & 4 & 5 \end{matrix}$$

We give the skeleton of the code, $int[] \quad a$ is the input series(**stored from** $a[1]$ **to** $a[n*m]$), $int[][] \quad res$ is the filled matrix(**stored from** $res[1][1]$ **to** $res[n][m]$), $boolean[][] \quad toFill$ is a two-dimensional boolean array with initial value $false$. Please fill the blanks to achieve its function.

(Hint: *dirx* and *diry* are arrays to control the direction of which element to fill next)

```
\\input part is omitted here
for(int i=1;i<=n;i++)
    for(int j=1;j<=m;j++)
        toFill[i][j]=__①__;
```

```
int dirx={1,0,-1,0},diry={0,1,0,-1};
int locx=0,locy=1,loca=0,dir=0;
while(__②__)
{
    loca++;
    locx+=dirx[dir];
    locy+=__③__;
    ____④____;
    toFill[locx][locy]=false;
    if(!toFill[locx+dirx[dir]][locy+diry[dir]])
        __⑤__;
}
for(int i=1;i<=n;i++)
    for(int j=1;j<=m;j++)
        System.out.printf("%d ",res[i][j]);
```

2. [**Longest Increasing Subsequence**] Given a series of n integer numbers $int[]\quad a$(**stored from** $a[1]$ **to** $a[n]$), the definition of subsequence of a is to choose some of the elements from the original series without disrupting the order(including the original series). For example, $\{1,3,5\}$ and $\{1,2,3,4,5\}$ are both the subsequence of $\{1,2,3,4,5\}$, but $\{1,3,2\}$ is not. A sequence is increasing means the next element is strictly larger than the previous one, i.e. $arr[i+1] > arr[i]$. Please fill the blanks to find the length of the longest increasing subsequence of a which stored in ans. (Math.max(int x,int y) returns the value of the larger one between x, y)

Hint:here $f[i]$ stores the maximum length of a subsequence of a that ended with $a[i]$

```
for(int i=1;i<=n;i++)
{
    f[i]=__①__;
    for(int j=1;__②__;j++)
        if(__③__)
```

```
            f[i]=Math.max(__④__);
    ans=Math.max(__⑤__);
}
```

3. [**Total War**] Immune System and coronavirus are engaged in a total war. We define two classes. Class *Virus* has (1)two fields: *isCoronavirus* of boolean type to represent whether the virus is a coronavirus, and *virulence*(毒性) of integer type.(2) a two-argument constructor to initialize the two fields respectively. Class *Human* has (1)two fields: *health* of integer type, and *temperature* of double type. (2) an one-argument constructor to initialize the *health* with the argument, and initialize the *temperature* with 37.0. (3) a non-static method named *contact* with an argument of Virus type and returned value of double type represents the temperature of the human after contacting the virus. If the virus is coronavirus, then the human will be infected if the *health* is less than the virus' *virulence* + 20, if not, the human will be infected if the *health* is less than the virus' *virulence*. If a human is infected, the *health* will decline 40 and the temperature will increase 1.5. If they are equal($health == virulence + 20$ for coronavirus, $health == virulence$ for non-coronavirus), the human's temperature remains the same, but the *health* will still decline 10. Otherwise, there's nothing happens, and you should only return the value of the temperature. For convenience, ignore the case that there will occur negative numbers, and we use public as the modifier of all fields.

Please write down the code of the two classes. If your code is correct, the following *main* method will output "37.0 38.5".

```
public static void main(String[] args)
{
    Human human=new Human(60);
    Virus virus1=new Virus(true,40);
    Virus virus2=new Virus(false,60);
    System.out.printf("%.1f ",human.contact(virus1));
    System.out.print(human.contact(virus2));
}
```