

氏名

点/102点

【問1】次はJavaに関する記述です。各記述が正しい場合は○を、誤っている場合は×を答えなさい。【各2計50】

- サブクラス型の変数にはスーパークラスのオブジェクトを代入できます
- JavaのすべてのクラスはObjectクラスのメンバーを継承します
- メソッドのオーバーライドはカプセル化を実現します
- 抽象メソッドとは処理本体が未定義のメソッドです
- 抽象メソッドの戻り値の型は指定しません
- 抽象クラスのオブジェクトは生成できません
- インタフェースはメンバーとして処理内容が定義されたメソッドをもつことができます
- インタフェースを実装してクラスを宣言します
- インタフェースをクラスと組み合わせることを拡張といいます
- 発生した例外に関する情報は例外クラスのオブジェクトにより保持されます
- 例外が発生したら、プログラムは中断ではなく常に強制終了します
- プログラムは例外を発生(送出)するコードを書くことができます
- ストリームとは異なる入出力機器から(ハ)のデータを統一的に扱う考え方で
- 文字ストリームは文字ベースのデータを読み書きするストリームです
- ストリームはラッパクラスを用いて構築されます
- コマンドライン引数の数は最大2個です
- コマンドライン引数はStringクラスの配列で受け取ることができます
- 1つのスレッドは1つの実行箇所をもつ一連の処理の流れです
- Javaはマルチスレッド処理を記述できます
- マルチスレッド処理では複数のスレッドが並行して実行されます
- 複数のスレッドはそれを開始した順番に終わるとは限りません
- 起動するスレッドの数は最大10個です
- GUIアプリケーションはWebブラウザ上で動作します
- GUIアプリケーションはボタンやチェックボックスなどのウィンドウ部品を配置できます
- マウスのクリックやドラッグなどのイベントを受け取る処理をイベント処理といいます

【問2】次はJavaで用いる用語です。各用語に関連する選択肢を1つずつ選び記号で答えなさい。【各2計16】

- | | |
|------------------------------|----------------|
| (1) オーバーライドの禁止 | (5) 例外処理 |
| (2) スレッド | (6) 抽象メソッド |
| (3) オブジェクトの属するクラスの検査 | (7) 標準入力 |
| (4) ウィンドウとウィンドウ部品を扱うためのライブラリ | (8) パッファ機能付き入力 |

■選択肢

- | | |
|--------------|--------------------|
| (A) final | (E) BufferedReader |
| (B) Runnable | (F) System.in |
| (C) try | (G) instanceof |
| (D) abstract | (H) AWT |

【問3】コマンドライン引数を用いたコード1を以下のように実行した場合の画面出力を正確に答えなさい。【各2計8】

- java Final3
- java Final3 0 1 2
- java Final3 1 2 0
- java Final3 2 0 1

■コード1

```
01: class Final3{
02:     public static void main(String[] args){
03:         if(args.length!=3){
04:             System.out.print("パラメータエラー");
05:         }else{
06:             int[] num= new int[args.length];
07:             for(int i=0;i<num.length;i++){
08:                 num[i]=Integer.parseInt(args[i]);
09:             }
10:             int sum=0;
11:             for(int i=0;i<num[0];i++){
12:                 sum+=num[i+1];
13:             }
14:             System.out.print(sum);
```

```
15:     }
16: }
17: }
```

【問4】次はスレッドを用いたコード2とその実行結果です。空欄を埋めなさい。【各2計8】

■コード2

```
01: class Sum extends 【 1 】{
02:     private int from, to, sum;
03:     public Sum(int f, int t){
04:         from=f;
05:         to=t;
06:     }
07:     public 【 2 】run(){
08:         sum=0;
09:         for(int i=from;i<=to;i++) sum+=i;
10:     }
11:     public int getSum(){
12:         return sum;
13:     }
14: }
15: class Final4{
16:     public static void main(String[] args){
17:         Sum th1=new Sum(1, 5);
18:         Sum th2=new Sum(6, 10);
19:         th1. 【 3 】 ();
20:         th2. 【 3 】 ();
21:         int sum=0;
22:         try{
23:             th1.join();
24:             th2.join();
25:             sum=th1.getSum()+th2.getSum();
26:         } 【 4 】 (InterruptedException e){
27:             sum=-1;
28:         }
29:         System.out.println("総計 "+sum);
30:     }
31: }
```

■実行結果

総計 55

【問5】単一継承に関する次の説明文の空欄を選択肢から選び答えなさい。重複して使用する場合もあります。【各2計10】

〔単一継承に関する説明文〕

単一継承とは、新しいクラスを宣言するときに、高々(1)つのクラスのメンバーを継承することである。一方、2つ以上のクラスのメンバーを継承することを(2)継承という。Javaではクラスは(3)継承であり、インタフェースは(4)継承である。このため、すでにスーパークラスを拡張して宣言されているクラスにおいて、スレッドを用いる場合はRunnable(5)を実装する。

■選択肢

- | | | | | | | | | | |
|---------|-------|----|-----|---------|------|----|----|----|----|
| 1 | 2 | 3 | クラス | オーバーライド | 拡張 | 単一 | 多重 | 実装 | 抽象 |
| インタフェース | ストリーム | 同期 | GUI | AWT | イベント | | | | |

【問6】GUIアプリケーションに関する次のコード3はマウスボタンを押した位置に鬼の面(oni.png,縦50ピクセル×横50ピクセル)の画像が表示され、離れた位置にさらに表示されます。この時、二つの画像の間は直線が描画されます。コードの空欄を選択肢から1つずつ選び、埋めてコードを完成させなさい。【各2計10】

■コード3

```
01: import java.awt.*;
02: import java.awt.event.*;
03: class Final6 extends Frame implements 【 1 】 {
04:     private Image oni;
05:     private int px,py,rx,ry;
06:     private boolean release=false;
07:     public Final6(){
08:         super("節分");
09:         Toolkit tk = getToolkit();
10:         oni = tk.getImage("oni.png");
11:         px=py=rx=ry=0;
12:         addMouseListener(this);
13:         【 2 】 (new SimpleWindowAdapter());
14:         setSize(250, 250);
15:         setVisible(true);
16:     }
17:     public void mouseClicked(MouseEvent e){}
18:     public void mouseEntered(MouseEvent e){}
19:     public void mouseExited(MouseEvent e){}
20:     public void mousePressed(MouseEvent e){
21:         px=e.getX();
22:         py=e.getY();
23:         release=false;
```

```
24:         【 3 】 ();
25:     }
26:     public void mouseReleased(MouseEvent e){
27:         rx=e.getX();
28:         ry=e.getY();
29:         release=true;
30:         【 3 】 ();
31:     }
32:     public void paint( 【 4 】 g){
33:         if(release){
34:             g.drawLine(px,py,rx,ry);
35:             g.drawImage(oni,rx-25,ry-25,this);
36:         }
37:         g.drawImage(oni,px-25,py-25,this);
38:     }
39:     public static void main(String[] args){
40:         Final6 obj = new Final6();
41:     }
42:     private class SimpleWindowAdapter 【 5 】 WindowAdapter{
43:         public void windowClosing(WindowEvent e){
44:             System.exit(0);
45:         }
46:     }
47: }
```

■選択肢

■実行例

Graphics MouseListener WindowListener
 Object interface paint repaint
 Frame final extends implements
 class abstract add addWindowListener



解答欄

【問1】

(1)	(2)	(3)	(4)	(5)
(6)	(7)	(8)	(9)	(10)
(11)	(12)	(13)	(14)	(15)
(16)	(17)	(18)	(19)	(20)
(21)	(22)	(23)	(24)	(25)

【問2】

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
-----	-----	-----	-----	-----	-----	-----	-----

【問3】

(1)	(2)	(3)	(4)
-----	-----	-----	-----

【問4】

(1)	(2)	(3)	(4)
-----	-----	-----	-----

【問5】

(1)	(2)	(3)	(4)	(5)
-----	-----	-----	-----	-----

【問6】

(1)	(2)	(3)	(4)	(5)
-----	-----	-----	-----	-----

お疲れ様でした!!