

2023年度	後期理解度テスト2試験問題 解答例		
科目名	JavaプログラミングⅡ		
担当教員	石原真紀夫		
実施日付	1月15日（月）	1時限目	09:00～（1組2組/C棟B1F）
持ち込み	禁 止		
情報工学科	年	組	学籍番号
氏名			

点／106 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】

- (1) 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
- (2) 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
- (3) メソッドのオーバーライドはカプセル化を実現します
- (4) 抽象メソッドとは戻り値の型の宣言を持たないメソッドです
- (5) 抽象メソッドは仮引数を持たなくてもよいです
- (6) インタフェースはメンバーとして抽象メソッドをもちます
- (7) 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
- (8) 既存のインタフェースを拡張して新しいインタフェースを宣言できます
- (9) 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
- (10) 例外は例外クラスにより表現されます
- (11) 例外はコンパイル時に、文法の間違いにより発生するエラーです
- (12) プログラマは例外を発生させるコードを書くことはできません
- (13) ストリームとは異なる入出力機器からのノへのデータを統一的に扱う考え方で
- (14) 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
- (15) 標準入力とはディスプレイのことです
- (16) コマンドライン引数は半角スペースで区切り複数個与えることができます
- (17) コマンドライン引数は String[] 型で受け取ることができます
- (18) コマンドライン引数の数の上限は 10 です
- (19) Java はマルチスレッド処理を記述できます
- (20) 複数のスレッドは常にそれを開始した順番に終わります
- (21) 複数のスレッドは互いに同期をとることはできません
- (22) GUI アプリケーションは線や円などのグラフィックスを描画することができます
- (23) GUI は Graphical User Interface の略です
- (24) GUI アプリケーションではマルチスレッドを利用できません
- (25) 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
- (2) アダプタクラス
- (3) 抽象クラス
- (4) オブジェクトの属するクラスの検査
- (5) 例外の送出
- (6) ウィンドウとウィンドウ部品を扱うためのライブラリ
- (7) スレッド
- (8) バッファ機能付き入力

■選択肢

- | |
|--------------------|
| (A) instanceof |
| (B) Runnable |
| (C) BufferedReader |
| (D) AWT |
| (E) final |
| (F) abstract |
| (G) throw |
| (H) MouseAdapter |

【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 10】

- (1) java Final03
- (2) java Final03 2 1 3
- (3) java Final03 4 3 2
- (4) java Final03 2 1 2 3
- (5) java Final03 0

■コード 1

```
01: class Final03{
02:     public static void main(String[] args){
03:         boolean flg=true;
04:         if(args.length==0){
05:             System.out.print("引数がありません");
06:         }else{
07:             int[] num=new int[args.length];
08:             for(int i=0;i<num.length;i++){
09:                 num[i]=Integer.parseInt(args[i]);
```

```
10:         }
11:         int sum=0;
12:         for(int i=0;i<num[0];i++){
13:             try{
14:                 sum+=num[i+1];
15:             }catch(ArrayIndexOutOfBoundsException e){
16:                 System.out.println("引数が足りません");
17:                 flg=false;
18:                 break;
19:             }
20:         }
21:         if(flг)System.out.print(sum);
22:     }
23: }
24: }
```

【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。選択肢から 1 つずつ選び、空欄を埋めなさい。【各 2 計 10】

■コード 2

```
01: class Company{
02:     private int sum=0;
03:     public 1 void add(int a){
04:         sum+=a;
05:     }
06:     public int getSum(){return sum;}
07: }
08: class Driver extends 2 {
09:     private Company cmp;
10:     public Driver(Company c){
11:         cmp=c;
12:     }
13:     public void 3 (){
14:         for(int i=0;i<3;i++){
15:             cmp.add(40);
16:         }
17: }
18: class Final04{
19:     public static void main(String[] args){
20:         boolean flg=true;
21:         Company cmp = new Company();
22:         Driver dr1 = new Driver(cmp);
23:         Driver dr2 = new Driver(cmp);
24:         dr1. 4 ();
25:         dr2. 4 ();
26:         try{
27:             dr1.join();
28:             dr2.join();
29:         } 5 (InterruptedException e){
30:             flg=false;
31:         }
32:         if(flг) System.out.println("利益"+cmp.getSum()+"円");
33:         else System.out.println("例外が発生しました");
34:     }
35: }
```

■実行結果

利益 240 円

■選択肢

sleep start run Runnable Thread join paint throw throws
IOException synchronized static try catch finally

【問 5】スレッドに関する次の説明文の空欄を選択肢から選びなさい。【各 2 計 10】
〔スレッドに関する説明文〕

スレッドの利用は、（ 1 ）クラスを拡張する方法と（ 2 ）インタフェースを実装する方法がある。いずれも継承される（ 3 ）メソッドを（ 4 ）して、新しいスレッドで処理したいコードを記述します。（ 1 ）クラスを用いた場合のスレッドの起動は、拡張したサブクラスのオブジェクトを生成し、継承した（ 5 ）メソッドを実行すればよい。

■選択肢

カプセル化 オーバーライド オーバーロード ポリモーフィズム Thread Object
Graphics Runnable MouseListener run() paint() main() start() join()

【問 6】GUI アプリケーションに関する次のコード 3 を実行したら、実行結果のような動作をした。選択肢から 1 つずつ選び、空欄を埋めなさい。【各 2 計 10】

■コード 3

```
01: import java.awt.*;
02: import java.awt.event.*;
03: class Final06 extends 1 {
04:     private Label sw;
05:     private int state=0;
```

```
06:     private String[] msg={"OFF","弱","中","強"};
07:     private Color[] clr={Color.BLACK,Color.BLUE,Color.GREEN,Color.RED,};
08:     public Final06(){
09:         super("ヒータ");
10:         sw=new Label(msg[state]);
11:         sw.setForeground(clrs[state]);
12:         add(sw);
13:         setSize(210,100);
14:         setVisible(true);
15:         addWindowListener(new MyWindowListener());
16:         sw.addMouseListener(new 2 ());
17:     }
18:     class MyWindowListener 3 WindowAdapter{
19:         public void windowClosing(WindowEvent e){
20:             System.exit(0);
21:         }
22:     }
23:     class MyMouseListener extends MouseAdapter{
24:         public void mouseClicked( 4 e){
25:             state++;
26:             if(state==4)state=0;
27:             sw.setText(msg[state]);
28:             sw.setForeground(clrs[state]);
29:         }
30:     }
31:     public static void main(String[] args){
32:         Final06 obj=new 5 ();
33:     }
34: }
```

■実行結果



■選択肢

Frame Thread Runnable implements extends MouseListener MyMouseListener
MyWindowListener Final06 MouseEvent WindowEvent start run paint main

解答欄

(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) E	(2) H	(3) F	(4) A	(5) G	(6) D	(7) B	(8) C
-------	-------	-------	-------	-------	-------	-------	-------

【問 3】

(1) 引数がありません	(2) 4	(3) 引数が足りません	(4) 3	(5) 0
--------------	-------	--------------	-------	-------

【問 4】

(1) synchronized	(2) Thread	(3) run	(4) start	(5) catch
------------------	------------	---------	-----------	-----------

【問 5】

(1) Thread	(2) Runnable	(3) run()	(4) オーバーライド	(5) start()
------------	--------------	-----------	-------------	-------------

【問 6】

(1) Frame	(2) MyMouseListener	(3) extends	(4) MouseEvent	(5) Final06
-----------	---------------------	-------------	----------------	-------------

お疲れ様でした!!