

_____点/100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) 抽象メソッドは戻り値の型が常にvoidです
- (5) 抽象メソッドの仮引数の数は0個または1個です
- (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) インタフェースのメンバーの数は偶数である必要があります
- (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) 例外が発生したら、例外処理の有無にかかわらず常に強制終了します
- (12) プログラムは独自の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(ハ)のデータを統一的に扱う考え方です
- (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) ストリームはラッパクラスを用いて構築されます
- (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クラスの配列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開始した順番に処理を終えます
- (21) 複数のスレッドの間で処理の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を描画することはできません
- (23) GUIアプリケーションはWebブラウザ上で動作します
- (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) ウィンドウのオープンやクローズを表すイベントは受け取ることはできません

【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) クラスの拡張の禁止 | (5) 例外の送出 |
| (2) 標準入力 | (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 |
| (3) 同期 | (7) スレッド |
| (4) マウスのイベント処理 | (8) ファイル入力 |

■選択肢

- | | |
|------------|-------------------|
| (A) final | (E) FileReader |
| (B) Thread | (F) System.in |
| (C) throw | (G) synchronized |
| (D) AWT | (H) MouseListener |

【問3】例外処理を含むコード1を次のようにコマンドライン引数を指定して実行した場合の画面出力を正確に答えなさい。ここで、[改行]はエンターキーを押すことを意味します。また、何も表示されない場合は「なし」と解答欄に記入しなさい。【各2計8】

- (1) >java Final03 [改行] (2) >java Final03 a b c [改行]
 (3) >java Final03 2 a b c [改行] (4) >java Final03 2 a [改行]

■コード1

```
01: class Final03{
02:     public static void main(String[] args){
03:         if(args.length >= 1){
04:             try{
05:                 String paras="";
06:                 int num=Integer.parseInt(args[0]);
07:                 for(int i=0;i<num;i++){
08:                     paras+=args[i+1];
09:                 }
10:                 System.out.println(paras);
11:             }catch(NullPointerException e){
12:                 System.out.println("例外 1");
13:             }catch(IndexOutOfBoundsException e){
14:                 System.out.println("例外 2");
15:             }catch(NumberFormatException e){
16:                 System.out.println("例外 3");
17:             }
18:         }
19:     }
```

参考 (JDK Web ドキュメントより)

NullPointerException 例外: オブジェクトが必要な場合に、アプリケーションがnull を使おうとするとスローされます。
 NumberFormatException 例外: アプリケーションが文字列を数値型に変換しようとしたとき、文字列の形式が正しくない場合にスローされます。

【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】次はファイル入力を用いたコード3と入力ファイルin.txtの内容です。入力ファイルを次のようにした場合の画面出力を正確に答えなさい。【各2計8】

- | | | | |
|--------|-------|-------|-------|
| (1) -1 | (2) 0 | (3) 4 | (4) 6 |
| 0 | 1 | 3 | 5 |
| 1 | 2 | 2 | 4 |
| 2 | 3 | 1 | 3 |
| 3 | 4 | 0 | 2 |
| 4 | 5 | -1 | 1 |

■コード3

```
01: import java.io.*;
02: class Final05{
03:     public static void main(String[] args) throws IOException{
04:         BufferedReader br;
05:         br=new BufferedReader(new FileReader("in.txt"));
06:         int num=Integer.parseInt(br.readLine());
07:         int i,sum=0;
08:         for(i=0;i<num;i++){
09:             String data;
10:             if((data=br.readLine())==null)break;
11:             sum+=Integer.parseInt(data);
12:         }
13:         if(i==num){
14:             System.out.println(sum);
15:         }else{
16:             System.out.println("err");
17:         }
18:         br.close();
19:     }
20: }
```

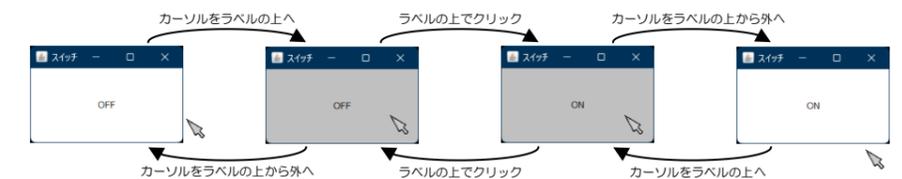
【問6】GUIアプリケーションに関する次のコード4を実行したら、実行結果のような動作をします。選択肢から1つずつ選び、空欄を埋めなさい。【各2計10】

■コード4

```
01: import java.awt.*;
02: import java.awt.event.*;
03: class Final06 extends [ 1 ]{
04:     private Label sw;
05:     private boolean on=false;
06:     private String[] msg={"OFF","ON"};
```

```
07:     public Final06(){
08:         super("スイッチ");
09:         sw=new Label(msg[on==false?0:1]);
10:         sw.setAlignment(Label.CENTER);
11:         sw.addMouseListener(new MyMouseListener());
12:         addWindowListener(new MyWindowListener());
13:         [ 2 ](sw);
14:         setSize(200,150);
15:         setVisible(true);
16:     }
17:     class MyWindowListener [ 3 ] WindowAdapter{
18:         public void windowClosing(WindowEvent e){
19:             System.exit(0);
20:         }
21:     }
22:     class MyMouseListener extends [ 4 ]{
23:         public void mouseClicked(MouseEvent e){
24:             on=!on;
25:             sw.setText(msg[on==false?0:1]);
26:         }
27:         public void mouseEntered(MouseEvent e){
28:             sw.setBackground(Color.lightGray);
29:         }
30:         public void mouseExited(MouseEvent e){
31:             sw.setBackground(Color.WHITE);
32:         }
33:     }
34:     public static void [ 5 ](String[] args){
35:         Final06 obj=new Final06();
36:     }
37: }
```

■実行結果



■選択肢

Frame Thread Runnable implements extends MouseListener MouseAdapter
 WindowAdapter Final06 MouseEvent WindowEvent start run paint main

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)
------	-----	-----	-----	-----

【問6】	(1)	(2)	(3)	(4)	(5)
------	-----	-----	-----	-----	-----