

2021年度	後期理解度テスト2試験問題
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12日（水） 1時限目（1組2組 A11）
持ち込み	禁 止
情報工学科	年 組 学籍番号

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】

- (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 Thread 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ポリモーフィズムを実現します
- (4) 抽象メソッドは戻り値の型が常に void です
- (5) 抽象メソッドは仮引数を常に持ちません
- (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) インタフェースはコンストラクタをもちます
- (8) インタフェースを実装してクラスを宣言します
- (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) 例外クラスは Throwable クラスのメンバーを継承します
- (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（へ）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) 標準入力とはマウスのことです
- (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の数の上限は 10 です
- (18) コマンドライン引数は文字列の配列としてメインメソッドの仮引数に渡されます
- (19) 1 つのスレッドは 1 つの実行個所をもつ一連の処理の流れです
- (20) Java はシングルスレッド処理のみを記述できます
- (21) 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
- (22) 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
- (23) GUI アプリケーションは画像を描画することができます
- (24) GUI は Graphical User Interface の略です
- (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) 標準入力
- (3) 抽象メソッド
- (4) マウスのイベント処理
- (5) 例外の送出
- (6) オブジェクトの属するクラスの検査
- (7) スレッド
- (8) バッファ機能付き入力

■選択肢

- | | |
|--------------|--------------------|
| (A) final | (E) BufferedReader |
| (B) Runnable | (F) System.in |
| (C) throw | (G) instanceof |
| (D) abstract | (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する。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 0 1 2 [改行] |
| (3) >java Final03 a b c [改行] | (4) >java Final03 2 1 [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[] params=null;
06:                 int num=Integer.parseInt(args[0]);
07:                 if(num >= 1)
08:                     params=new String[num];
09:                 for(int i=0;i<params.length;i++)
10:                     params[i]=args[i+1];
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:     }
20: }
```

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、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】次はファイル入力を用いたコードです。入力ファイル in.txt 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
2	2	2	2
5	5	5	5
1	1	1	1
3	3	3	3
4	4	4	4

■コード 2

```
01: import java.io.*;
02: class Final04{
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 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 JApplet{
04:     private Label sw;
05:     private int state=0;
06:     private String[] msg={"遮断","通電"};
07:     public Final06(){
08:         super("スイッチ");
09:         Button bt=new Button("切り替え");
10:         sw=new Label(msg[state]);
11:         Panel pl=new Panel();
12:         pl.add(bt);
13:         pl.add(sw);
14:         add(pl);
15:         addWindowListener(new MyWindowListener());
16:         bt.addActionListener(new MyActionListener());
17:         setVisible(true);
18:     }
```

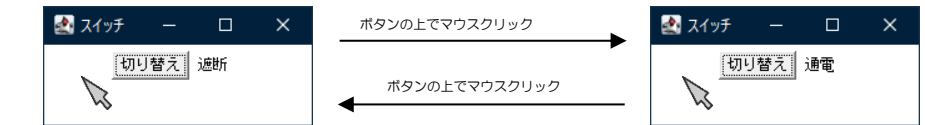
```
19:     class MyWindowListener extends WindowAdapter{
20:         public void windowClosing(WindowEvent e){
21:             System.exit(0);
22:         }
23:     }
24:     class MyActionListener implements ActionListener{
25:         public void actionPerformed(ActionEvent e){
26:             state = state == 0 ? 1 : 0;
27:             sw.setText(msg[state]);
28:         }
29:     }
30:     public static void main(String[] args){
31:         Final06 obj=new Final06();
32:     }
33: }
```

参考

クラス Panel：2 つのボタンのように複数の GUI 部品を配置するには Panel クラスを用います。Panel クラスは複数の GUI 部品を配置できるパネルです。このパネルに 2 つのボタンを配置し、ウィンドウに配置するには次のようにします。

```
Panel p = new Pane();
p.add(new Button("ボタン 1")); // パネルに 1 つ目のボタンを配置
p.add(new Button("ボタン 2")); // パネルに 2 つ目のボタンを配置
add(p); // Frame にパネルを配置
```

■実行結果



■選択肢

Frame Thread Runnable implements extends sw bt WindowListener WindowAdapter
MouseListener MouseAdapter MyWindowListener MyActionListener WindowEvent
MouseEvent ActionEvent

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!!