

## 汎用 Java プログラミング実習テスト 2022/6/14

解答は Google フォームでおこないます。

以下のリンクを右クリックし、「新しいウィンドウで開く」としてください。

<https://forms.gle/VLQjuV11CqqHeVQFA>

### 1 問題

次の  ~  にあてはまるものを解答群から選んでください。(各 2 点)

1. Java ではソースコードを作成したあと  してから実行する。
2.  されたものは  に変換されている。
3.  は仮想マシンである  が解釈してマシン語に置き換えられ実行される。
4. 変数を宣言する際は  が必要である。
5. メソッドを定義する際に "public void getName(){}" となっている場合の "public" を  という。
6. メソッドを定義する際に "public void getName(){}" となっている場合の "void" を  という。
7. メソッドを定義する際に "public void getName(){}" となっている場合の "getName()" を  という。
8. int 型の配列 hand を宣言する場合は "  hand" とする。
9. Java の配列は  である。
10. クラス Hero をインスタンス化する際は "  Hero()" とする。

表 1 解答群

(ア) アクセス修飾子	(イ) JVM	(ウ) new	(エ) バイトコード
(オ) メソッド名	(カ) コンパイル	(キ) Apache	(ク) int[ ]
(ケ) 基本データ型	(コ) 戻り値の型	(サ) 参照型	(シ) 型

## 2 問題

次の  ~  にあてはまるものを解答群から選んでください。(各 2 点)

1. クラスブロック内にある変数のことを  という。
2. クラス内にあるメソッドから、"name" という  にアクセスするには ".name" とする。
3.  のアクセス修飾子は基本的に  にする。これを"カプセル化"という。
4. それに対して"メソッド"は基本的に  にする。
5. カプセル化をおこなうと外部から  にアクセスできなくなるので、 の値を取得するため  メソッドを作成する。
6. カプセル化をおこなうと外部から  にアクセスできなくなるので、 に値を設定するため  メソッドを作成する。
7. Character クラスを元に Hero クラスを作りたい場合は Character クラスを  して作成すればよい。
8.  をおこなうためのキーワードは "Hero  Character" である。
9. 実装部分を持たない abstract なメソッドやクラスを  メソッド、 クラスという。
10. "try {} catch (Exception e) {}" を  という。

表 2 解答群

(ア) 継承	(イ) プロパティ	(ウ) private	(エ) public
(オ) this	(カ) extends	(キ) ゲッター	(ク) static
(ケ) セッター	(コ) フィールド	(サ) 例外処理	(シ) 抽象

### 3 問題

次の  ~  にあてはまるものを解答群から選んでください。同じ記号を選択してもよいです。  
(各 2 点)

リスト 1    Main.java

---

```
1   java.util.Random;                // クラスのインポート
2   java.util.Scanner;
3
4  public  Main {                // クラスの作成
5      public  void main( args) {    // メインメソッドの作成
6           num1 = new  ().nextInt(10);    // ランダムな整数値を生成して代入
7           num2 = new  ().nextInt(20);
8
9          System.out.println("文字 1 を入力してください");
10          str1 = new  (System.in).nextLine();    // ユーザーに文字列を入力してもらって代入
11         System.out.println("文字 2 を入力してください");
12          str2 = new  (System.in).nextLine();
13         System.out.println("数値 1 : " + num1);
14         System.out.println("数値 2 : " + num2);
15         System.out.println("文字 1 : " + str1);
16         System.out.println("文字 2 : " + str2);
17         System.out.println("何算をしますか ? ");
18         System.out.println("1:足し算  2:引き算  3:掛け算  4:割り算");
19         int select = new  (System.in).nextInt();    // ユーザーに数値を入力してもらって代入
20         if ( == 1) {                // ユーザーの入力値が 1 の場合
21             System.out.println("何と何を足しますか ? ");
22             System.out.println("5:数値 1 と数値 2   6:文字 1 と文字 2");
23             select = new  (System.in).nextInt();
24         }
25
26          (select) {                // 変数の中身によって分岐
27              2:                // 2 の場合
28                 System.out.println(minus(num1, num2));
29                  ;                //  を抜ける
30              3:                // 3 の場合
31                 System.out.println(multi(num1, num2));
32                  ;                //  を抜ける
33              4:                // 4 の場合
```

```

34     System.out.println(division(num1, num2));
35     12 ; // 10 を抜ける
36     11 5: // 5 の場合
37     System.out.println(plus(num1, num2));
38     12 ; // 10 を抜ける
39     11 6: // 6 の場合
40     System.out.println(plus(str1, str2));
41     12 ; // 10 を抜ける
42     13 : // 上記以外の場合
43     System.out.println("エラー");
44 }
45 }
46 public static 14 plus(int num1, int num2) { // 戻り値の型を考える
47     return num1 + num2;
48 }
49 public static 15 plus(String str1, String str2) { // 戻り値の型を考える
50     return str1 + str2;
51 }
52 public static int minus(int num1, int num2) {
53     return num1 - num2;
54 }
55 public static int multi(int num1, int num2) {
56     return num1 * num2;
57 }
58 public static int division(int num1, int num2) {
59     return num1 / num2;
60 }
61 }
62

```

表 3 解答群

(ア) int	(イ) switch	(ウ) String[]	(エ) String
(オ) select	(カ) case	(キ) default	(ク) import
(ケ) class	(コ) Random	(サ) while	(シ) Scanner
(ス) exit	(セ) static	(ソ) abstract	(タ) break

## 4 問題

以下のコード中の  1  ~  15  にあてはまる適切な語句を、あとの解答群から選び、記号で答えてください。同じ記号を複数回使ってもいいです。(各 2 点)

リスト 2 Main.java

```
1 public class Main {
2     public static void main (String[] args) {
3         // ヒーロークラスのインスタンスを作成。名前は「ミナト」、HP は「100」
4          1 h = new Hero (  2 ,  3 );
5         // マタンゴクラスのインスタンスを作成。サフィックスは「A」
6          4 m = new Matango (  5 );
7         h.attack (  6 );           // ヒーローインスタンスの攻撃メソッドを実行
8         m.attack (  7 );           // マタンゴインスタンスの攻撃メソッドを実行
9         m.run ();
10        h.run ();
11         8 c = h;           // ヒーロークラスのインスタンスは親クラスの型にも入れることができる
12        System.out.println (c.getName());
13    }
14 }
```

リスト 3 Character.java

```
1 public  9 class Character {           // 抽象クラス
2     private String name;
3     public Character (String name) {
4         this.name = name;
5     }
6     public  9 void attack (Matango m);           // 抽象メソッド
7     public void run () {
8         System.out.println (this.name + "は逃げ出した!");
9     }
10    public String getName () {
11        return name;
12    }
13    public void setName (String name) {
14        this.name = name;
15    }
```

16 }

17

---

#### リスト 4 Hero.java

---

```
1 public class Hero 10 Character { // キャラクタークラスを継承
2     private int hp;
3     public Hero (int hp, String name) {
4         11 ( 12 ); // 親クラスのコンストラクタを呼び出す
5         13 .hp = hp; // hp フィールドに値を設定
6     }
7     public void attack ( 14 m) { // マタンゴインスタンスを受け取る
8         System.out.println ("お化けキノコを攻撃!");
9         System.out.println ("お化けキノコに 5 ポイントのダメージを与えた!");
10        m.setHp (m.getHp() - 5);
11    }
12    public int getHp () {
13        return hp;
14    }
15    public void setHp (int hp) {
16        this.hp = hp;
17    }
18 }
19
```

---

#### リスト 5 Matango.java

---

```
1 public class Matango {
2     private int hp = 10;
3     private char suffix;
4     public static 15 int EXP = 10; // 定数フィールドを設定
5     public Matango (char suffix) {
6         this.suffix = suffix;
7     }
8     public void attack (Hero h) {
9         System.out.println ("お化けキノコ" + this.suffix + "の攻撃!");
10        System.out.println ("勇者に 3 ポイントのダメージを与えた!");
11        h.setHp (h.getHp() - 3);
12    }
13    public void run () {
```

```

14      System.out.println("お化けキノコは逃げ出した!");
15  }
16  public int getHp () {
17      return hp;
18  }
19  public void setHp (int hp) {
20      this.hp = hp;
21  }
22  public char getSuffix () {
23      return suffix;
24  }
25  }

```

表 4 解答群

(ア) 'A'	(イ) h	(ウ) Hero	(エ) abstract
(オ) 100	(カ) this	(キ) m	(ク) super
(ケ) name	(コ) Matango	(サ) implements	(シ) "ミナト"
(ス) Character	(セ) final	(ソ) extends	(タ) fixed