

開始日時	2023年 06月 5日(月曜日) 02:55
状態	終了
完了日時	2023年 06月 5日(月曜日) 03:06
所要時間	11 分 34 秒
素点	13 / 13
評点	100 / 100

問題 1

完了

1 / 1

インターフェイスを使用する理由

クラス設計において抽象クラスの代わりにインターフェイスを使用する理由として適切なものを選びなさい。

- ☐ a. staticもしくはfinalではないフィールドを宣言したい場合
- ☐ b. メンバーのアクセスレベルの規定にpublic以外の修飾子を使用したい場合
- ☒ c. 関連しないクラスが実装を提供することが予想される場合
- ☐ d. 関連するクラス間で実装コードを共有したい場合
- ☒ e. 多重継承のメリットを享受したい場合

問題 2

完了

1 / 1

パッケージ構文

以下の選択肢より、パッケージの宣言が正しく行われている記述のコードを1つ選びなさい。

- ☐ a. 

```
import java.io.*;
package aaa {
    public class Sample {}
}
```
- ☐ b. 

```
import java.io.*;
package aaa (
    public class Sample {}
);
```
- ☒ c. 

```
package aaa;
import java.io.*;
public class Sample {}
```
- ☐ d. 

```
import java.io.*;
package aaa;
public class Sample {}
```
- ☐ e. 

```
package aaa {
    import java.io.*;
    public class Sample {}
}
```

## 問題 3

完了

1 / 1

## インターフェイスの抽象メソッド

以下のプログラムの実行結果を下の選択肢より選びなさい。

```
interface Rideable {
    String getGait();
}

class Camel implements Rideable {
    int weight = 2;

    public static void main(String[] args) {
        new Camel().go(8);
    }

    void go(int speed) {
        ++speed;
        weight++;
        int walkrate = speed * weight;
        System.out.print(walkrate + getGait());
    }
    @Override
    String getGait() {
        return " mph, lope";
    }
}
```

- ☐ a. 18 mph, lope
- ☐ b. 24 mph, lope
- ☒ c. コンパイルエラーで実行できない。
- ☐ d. 実行時例外が発生する。
- ☐ e. 16 mph, lope
- ☐ f. 27 mph, lope

## 問題 4

完了

1 / 1

## 構文記述問題

MainクラスはISampleインターフェイスとITestインタフェースを**実装**したい。

以下のクラス宣言の「      」に入る単語を1つ記述してください。

```
public class Main 「      」 ISample, ITest { }
```

※スペースは不要です。

答え:

問題 5

完了

1 / 1

## インターフェイスのメンバ

次の中から、インタフェースに定義できないものを選びなさい。

- ☐ a. abstract void setData(String data);
- ☐ b. public void setData(String data);
- ☒ c. abstract String b;
- ☐ d. void setData(String data);
- ☒ e. private String c = "C";
- ☐ f. public String a = "A";

問題 6

完了

1 / 1

## インターフェイスと抽象メソッド

以下のインターフェイスを作成した。

```
package smalltest;
public interface NewInterface {
    public boolean isValid();
}
```

上記インターフェイスを実装したSmallTestクラスを以下のように作成したところ、コンパイルエラーが出てしまった。

```
package smalltest;
public class SmallTest implements NewInterface {
    @Overload
    public boolean isValid() {
        // なんらかの処理
        return true;
    }
}
```

コンパイルエラーが出た原因として、以下の理由が正しい場合は○を、正しくない場合は×を選択してください。

### コンパイルエラーが出た理由

インターフェイスにて定義された抽象メソッドと、@Overrideアノテーションにてオーバーライドされたメソッドとではシグネチャが異なり、SmallTestクラスのisValid()メソッドは、public void isValid()として定義しなければならないため。

1つ選択してください:

- ☐ ○
- ☒ ×

## 問題 7

完了

1 / 1

## 抽象クラスと抽象メソッド

抽象クラスと抽象メソッドに関する説明として正しいものはどれか。

以下の選択肢より選んでください。

- ☐ a. 抽象クラスを継承するサブクラスは必ず抽象クラスでない具象クラスとして宣言する必要があり、抽象メソッドをオーバーライドしなければならない
- ☐ b. 抽象メソッドは必ずしも抽象クラスを継承した具象クラスにてオーバーライドしなくてもよい
- ☐ c. 抽象メソッドが宣言されたクラスは、暗黙的に抽象クラスとなる
- ☒ d. 抽象クラスをインスタンス化することはできない
- ☐ e. 抽象クラスには、抽象メソッドでない具象メソッドを宣言することはできない

## 問題 8

完了

1 / 1

## abstract

次の中から、abstractで修飾**できないもの**をすべて選びなさい。

- ☐ a. メソッド
- ☒ b. 変数
- ☒ c. パッケージ
- ☐ d. クラス

## 問題 9

完了

1 / 1

## Java SE 8 のインターフェイス仕様

Java SE 8より実装された新しいインターフェイス(interface)の仕様に関する説明として正しいものはどれか。

以下の選択肢より選んでください。

- ☐ a. クラスに対してインターフェイスを複数実装することができなくなった。
- ☐ b. インターフェイスは単一の機能を保持するべきなので、抽象メソッドを1つのみ実装できるようにした。
- ☒ c. staticメソッドとdefaultメソッドが宣言できるようになり、インターフェイス自体に機能を持たせることが可能となった。
- ☐ d. 定数は列挙型で宣言する必要があり、インターフェイスでは定数を宣言できなくなった。
- ☐ e. インターフェイスにクラスを継承できるようになり、より詳しいインターフェイスの設計が可能となった。

## 問題 10

完了

1 / 1

## インターフェイスの実装

次のコードをコンパイル、実行したときの結果として、正しいものを選びなさい。

```
interface A {
    int test(int x, int y);
}
class B implements A {
    public int test(int a, int b) {
        return (a * b) / 2;
    }
}
class C implements A {
    public int test(int c, int d) {
        return (int)(3.2 * (c * d));
    }
}
class Main {
    public static void main(String[] args) {
        A[] array = {new B(), new C()};
        System.out.print(array[0].test(3, 2) + " ");
        System.out.println(array[1].test(3, 2));
    }
}
```

- ☐ a. Mainクラスの4行目 (System.out.print(array[0].test(3, 2) + " ");) でコンパイルエラーになる
- ☐ b. Cクラスのコンパイルに失敗する
- ☐ c. 「3 19」が表示され、例外がスローされる
- ☐ d. Bクラスの3行目 (return (a \* b) / 2;) でコンパイルエラーになる
- ☒ e. 「3 19」が表示される

## 問題 11

完了

1 / 1

## コンパイルと実行結果選択

次のコードをコンパイル、実行したときの結果として正しいものを選びなさい。

**Super.java**

```
public class Super {  
    static String name;  
    public void print() {  
        System.out.println("Super : " + name);  
    }  
}
```

**Sub.java**

```
public class Sub extends Super {  
    public void print() {  
        System.out.println("Sub : " + name);  
    }  
    public static void main(String[] args) {  
        Super a, b;  
        a = new Super();  
        b = new Sub();  
        a.name = "A";  
        b.name = "B";  
        a.print();  
        b.print();  
    }  
}
```

- ☐ a. 

Sub : B
Super : A
- ☐ b. 

Super : A
Super : B
- ☐ c. コンパイルできない
- ☐ d. 何も表示されないまま正常終了する
- ☐ e. 

Super : A
Sub : B
- ☒ f. 

Super : B
Sub : B
- ☐ g. 

Sub : A
Sub : A

## 問題 12

完了

1 / 1

## インターフェイスのメンバ

次のコードのコンパイルを成功させるには、以下のコードの「// Question!!!」にどのコードを追加することができるか。正しいものを2つ選びなさい。

```
public interface Sample {  
    // Question!!!  
}
```

- ☒ a. `void getName();`
- ☐ b. `String name;`
- ☒ c. `public void print();`
- ☐ d. `public static void sayHello(String name);`
- ☐ e. `private void setName(String name);`

## 問題 13

完了

1 / 1

## インターフェイスの多重実装

以下のコードに関する説明として、正しいものはどれか。

```
public interface A {  
    static void showA() {  
        System.out.println("A");  
    }  
    default void x() {  
        System.out.println("A::x");  
    }  
}
```

```
public interface B {  
    static void showB() {  
        System.out.println("B");  
    }  
    default void x() {  
        System.out.println("B::x");  
    }  
}
```

```
public class C implements A, B {  
    public static void main(String[] args) {  
        C app = new C();  
        app.x();  
    }  
}
```

以下の選択肢より選んでください。

- ☐ a.   
  
が表示される。
- ☐ b. 実行は可能だが結果は何も表示されない
- ☐ c.   
が表示される。
- ☒ d. メソッド x が競合するため、Cクラスでコンパイルエラーとなる
- ☐ e.   
が表示される。