

変数

問1

次の実行結果となるように、①に入れるコードとして正しいものを選びなさい。

【実行結果】

A

【ソースコード】

```
public class Test {  
  
    public static void main(String[] args) {  
  
        String str = ①;  
  
        System.out.println(str);  
  
    }  
}
```

1. A
2. {A}
3. 'A'
4. "A"

正解:4

解説

『東京ITスクールJavaテキスト 上巻』第4章/型/型とは

Java総合試験（問題数:30問、想定所要時間:60分）

扱う文字が1文字であっても、String型の変数には文字列として代入されます。

選択肢3は文字リテラルですので、コンパイルエラーとなります。

したがって、選択肢4が正解です。

問2

次の実行結果となるように、①に入れるコードとして正しいものを選びなさい。

【実行結果】

200

【ソースコード】

```
public class Test {  
  
    public static void main(String[] args) {  
  
        int num1 = 100;  
  
        ①  
  
        System.out.println(num1);  
  
    }  
}
```

1. System.out.println(200);
2. num1 = 200;
3. int num1 = 200;
4. num2 = 200;

正解:2

解説

『東京ITスクールJavaテキスト 上巻』第4章/変数を使う/値の代入

実行結果では「200」が1行のみ出力されています。

そのため、選択肢1の処理を①に記述するのは不適切です。

また、同じ名前の変数は同一のメソッド内では宣言できません。そのため、選択肢3は不正解です。

選択肢4の処理では、変数num2の型が宣言されていないため、コンパイルエラーとなります。

選択肢2の内容が今回は適切です。

したがって、選択肢2が正解です。

演算子

問3

次のソースコードをコンパイル、実行したい。想定される結果として正しいものを選びなさい。

【ソースコード】

```
public class Test {  
  
    public static void main(String[] args) {  
  
        String number1 = "100";  
  
        int number2 = 500;  
  
        number2 -= 100;  
    }  
}
```

```
        System.out.println(number1 + number2);  
    }  
}
```

1. 500
2. 100500
3. 100400
4. 600

正解:3

解説

『東京ITスクールJavaテキスト 上巻』第6章/演算子の種類/文字列連結演算子

変数number1は文字列型です。

文字列型の変数と数値型の変数を「+」演算子で結ぶと、この演算は加算ではなく、文字列連結として行われます。

今回のソースコードでは文字列「100」と数値「400」の文字列連結となるため、結果は「100400」となります。

したがって、選択肢3が正解です。

問4

次のソースコードを実行して、下記の実行結果(3÷4の計算結果)を出力したい。この場合、ソースコード中の①に記述すべき内容はどれか。正しいものを選びなさい。

【実行結果】

0.75

【ソースコード】

```
public class Test {  
  
    public static void main(String[] args) {  
  
        int number1 = 3;  
  
        int number2 = 4;  
  
        ①  
  
        System.out.println(answer);  
  
    }  
}
```

1. int answer = (double) number1 / number2;
2. int answer = number1 / number2;
3. double answer = (double) number1 / number2;
4. double answer = number1 / number2;

正解:3

解説

『東京ITスクールJavaテキスト 上巻』第6章/型変換/サイズが小さい型に代入する

double型からint型への代入は不可となるため、選択肢1は不正解です。

また、選択肢2は出力結果が0になってしまうため不正解です。

「(double) number1 / number2」では、まずnumber1の型がdouble型にキャストされます。

その結果、number1とnumber2の割り算の商は「0.75」となります。

したがって、選択肢3が正解です。

なお、割り算の結果を代入する先の変数がdouble型であっても、int型同士の割り算が実行された時点で、小数点以下の値が切り捨てられるため、選択肢4は不正解です。

配列

問5

次のソースコードをコンパイル、実行したい。想定される結果として正しいものを選びなさい。

【ソースコード】

```
public class Test {  
  
    public static void main(String[] args) {  
  
        String[] strArray = {"-*", "-*", "-*", "-*"};  
  
        System.out.print(strArray[1]);  
  
        System.out.print(strArray[2]);  
  
        System.out.print(strArray[3]);  
  
        System.out.print(strArray[4]);  
  
    }  
}
```

1. -*--*--*
2. -*--*--*-
3. -*--*-

4. 実行時に例外が送出される

正解:4

解説

『東京ITスクールJavaテキスト 上巻』第7章/配列の準備/要素の呼び出し

配列のインデックスは0から始まります。

問題文の「String[] strArray = {"-*", "-*", "-*", "-*"};」のように配列が宣言、初期化された場合は、インデックスは0～3までということになります。

したがって、System.out.print(strArray[4]);を実行すると、存在しないインデックス4番の要素を参照しようとするため、例外が送出されるはずです。

選択肢4が正解です。

問6

次のコードをコンパイル、実行したい。想定される結果として正しいものを選びなさい。

【ソースコード】

```
public class Test {  
  
    public static void main(String[] args) {  
  
        int[] numbers1 = new int[3];  
  
        int[] numbers2;  
  
  
        numbers1[0] = 10;
```

Java総合試験（問題数:30問、想定所要時間:60分）

```
numbers1[1] = 30;

numbers1[2] = 80;

numbers2 = numbers1;

numbers2[2] = 5;

System.out.println(numbers1[2]);

}

}
```

1. 10
2. 30
3. 80
4. 5

正解:4

解説

『東京ITスクールJavaテキスト 上巻』第7章/配列の準備/配列変数間での代入

配列変数間での代入の問題です。

変数numbers1の値を変数numbers2に代入することで、2つの変数が同一の配列を参照している状態となります。

その後、変数numbers2を使用して要素の値を操作すると、同じ配列を参照しているので、変数numbers1の要素の値にもその操作結果が反映されます。

変数numbers2を使用してインデックス2番の要素に「5」を代入し、変数numbers1で同一のインデックスの要素を参照すると、同様に「5」という値が取得できます。

したがって、選択肢4が正解です。

分岐処理

問7

次のソースコードをコンパイル、実行したい。実行結果として変数bの値が4として出力されるとき
【ある整数値】はどれか。正しいものを選びなさい。

【ソースコード】

```
public class Test {  
  
    public static void main(String[] args) {  
  
        int a = 【ある整数値】;  
  
        int b = 0;  
  
        if (a < 0 || 10 < a) {  
  
            b = 1;  
  
        } else if (a % 2 == 0 && a % 3 == 0) {  
  
            b = 2;  
  
        } else if (a % 3 == 0 || a % 5 == 0) {  
  
            b = 3;  
  
        } else if (!(a < 3)) {  
  
            b = 4;  
  
        } else {  
  
            b = 5;  
  
        }  
    }  
}
```

```
    }  
  
    System.out.println(b);  
  
}  
  
}
```

1. 2
2. 6
3. 7
4. 10

正解:3

解説

『東京ITスクールJavaテキスト 上巻』第6章/演算子/算術演算子

『東京ITスクールJavaテキスト 上巻』第8章/条件と関係演算子/関係演算子とは

『東京ITスクールJavaテキスト 上巻』第8章/if～else if～else文/if～else if～else文とは

bの値が4になる条件は「3以上である」です。

この時点では選択肢1が不正解であることが分かります。

また、bの値が4となるaの値は、1～3つ目の各条件を全て満たさない数値となります。

具体的には、「0以上で10以下である」、「6の倍数ではない」、「3の倍数ではなく、かつ5の倍数でもない」という条件を全て満たす数値です。

したがって、該当する値は7のみとなるため、選択肢3が正解です。

問8

次のソースコードをコンパイル、実行したい。想定される結果として正しいものを選びなさい。

【ソースコード】

```
public class Test {  
  
    public static void main(String[] args) {  
  
        int number = 2;  
  
        switch (number) {  
  
            case 1:  
  
                System.out.print("A");  
  
            case 2:  
  
                System.out.print("B");  
  
            case 3:  
  
                System.out.print("C");  
  
            default:  
  
                System.out.print("D");  
  
        }  
  
    }  
}
```

1. BCD
2. BD
3. ABCD
4. B

正解:1

解説

『東京ITスクールJavaテキスト 上巻』第8章/switch文/switch文の注意点

case文内にbreak文を記述しない場合、対象のcase文の条件を満たすと、そのcase文内の処理に加えて、更に下にあるcase文内の処理まで無条件で実行されます。

今回のソースコードでは、「case 2:」の条件を満たします。

すると、文字列「B」を標準出力した後に、続けて「C」、「D」も出力されます。

したがって、選択肢1が正解です。

反復処理

問9

次のソースコードの処理を意図的に長時間ループさせるには、①にどのような内容を記述すべきか。

正しいものを選びなさい。

【ソースコード】

```
public class Test {  
  
    public static void main(String[] args) {  
  
        for (①) {  
  
            System.out.println(i);  
  
        }  
    }  
}
```

```
}  
  
}
```

1. `int i = 0; i <= 1000; i++`
2. `int i = 0; i > 0; i--`
3. `int i = 0; i <= 0; i++`
4. `int i = 0; i <= 0; i--`

正解: 4

解説:

『東京ITスクールJavaテキスト 上巻』第9章/for文/for文とは

『東京ITスクールJavaテキスト 上巻』第9章/while文/無限ループ

for文で長時間ループを実行するには、for文の繰り返しの条件が常にtrueとなるように処理を書けばよいです。

条件「`int i = 0; i <= 0; i--`」では、ループカウンタ*i*の値が繰り返すごとに0から1ずつ減算されます。最終的にint型の最小値-1が実行され、変数*i*はオーバーフローにより最大値になります。その際にループをぬけますが、約20億回ループが実行されます。

したがって、選択肢4が正解です。

問10

次のソースコードをコンパイル、実行したい。想定される結果として正しいものを選びなさい。

【ソースコード】

```
public class Test {
```

```
public static void main(String[] args) {  
  
    int number = 1;  
  
    while (number > 0) {  
  
        ++number;  
  
        System.out.println(number);  
  
        if (number == 10) {  
  
            break;  
  
        }  
    }  
}
```

1. 数字が2から10まで出力される
2. 数字が1から10まで出力される
3. 2以上の数字が無限に出力される
4. 1以上の数字が無限に出力される

正解: 1

解説

『東京ITスクールJavaテキスト 上巻』第9章/while文/while文とは

『東京ITスクールJavaテキスト 上巻』第9章/処理の流れの変更/break文とは

このソースコードのwhile文では「変数numberの値が0より大きい」という条件を満たす限り、繰り返し処理を実行します。

while文のブロック内では、まず変数numberの値に1を加算しています。

変数numberの初期値は1ですから、1回目の繰り返し処理では、変数numberの値は2となり、その後に出力されます。

そのため、選択肢2と4は不正解です。

また、繰り返し処理にはif文が含まれており、変数numberの値が10である場合、break文により強制的に繰り返し処理が中断されます。

そのため、10までが画面に出力されることになります。

したがって、選択肢1が正解です。

フィールド、メソッド

問11

次のCarクラス中の各フィールドの値を、他のクラスのメソッドから呼び出し、直接値を代入したい。

値を代入する処理の記述として正しいものを選びなさい。

ただし、オブジェクトの参照情報が格納されている変数の名前は「car」とします。

【ソースコード】

```
public class Car {  
  
    int num;  
  
    double gas;  
  
}
```

1. car.num = 5248;

- `car.gas = 21.5;`
2. `car.num() = 5248;`
- `car.gas() = 21.5;`
3. `num = car(5248);`
- `gas = car(21.5);`
4. `Car.num = 5248;`
- `Car.gas = 21.5;`

正解: 1

解説:

『東京ITスクールJavaテキスト 下巻』第13章/クラスの宣言とオブジェクトの生成/フィールドを利用する

オブジェクトの参照情報が格納されている変数から、そのオブジェクトに属するフィールドを呼び出せます。その際には、「変数名.フィールド名」と記述します。

フィールドに値を直接代入するには「変数名.フィールド名 = 値」と記述します。

したがって、選択肢1が正解です。

なお、選択肢4はstatic変数に値を代入する場合の記述です。

今回のソースコードではnum、gasともにstatic修飾子が付与されていないため、インスタンス変数となります。

そのため、「クラス名.フィールド名」という形式で呼び出すことはできません。

問12

次のソースコードをコンパイル、実行したい。想定される結果として正しいものを選びなさい。

【ソースコード】

```
public class Test {  
  
    public static void main(String[] args) {  
  
        A a = new A();  
  
        int num = 0;  
  
        a.calc(3, 5);  
  
        System.out.println(num);  
  
    }  
}
```

```
public class A {  
  
    int calc(int num1, int num2) {  
  
        return num1 + num2;  
  
    }  
}
```

1. 0
2. 3
3. 5
4. 8

正解: 1

解説

『東京ITスクールJavaテキスト 下巻』第14章/メソッドの引数/情報を渡し、メソッドを呼び出す

『東京ITスクールJavaテキスト 下巻』第14章/メソッドの戻り値/戻り値の仕組み

calc()メソッドに第1引数として3、第2引数として5を渡しています。

calc()メソッドでは渡された2つの引数の加算をし、戻り値の8を呼び出し元に値を返しています。

しかし、main()メソッド内でcalc()メソッドを呼び出していますが、戻り値を変数に代入していないため、戻り値の値はcalc()メソッド実行後に消えてしまいます。

その後、変数numの値を標準出力していますが、この変数は初期値のままであるため、出力結果は「0」となります。

したがって、選択肢1が正解です。

戻り値をその後の処理でも利用したい場合は、忘れずに変数に代入するようにしましょう。

アクセス制限

問13

次のプログラムを記述したところ、コンパイルエラーが発生した。①～③の中で、コンパイルエラーが発生する処理を選びなさい。

ただし、A.javaとB.javaは同じパッケージ内に格納されているものとします。

【ソースコード】

```
public class A {  
  
    public String str;  
  
    private int num;  
  
    boolean flg;
```

```
}
```

```
public class B {  
  
    public static void main(String[] args) {  
  
        A a = new A();  
  
        a.str = "a"; // ①  
  
        a.num = 10; // ②  
  
        a.flg = true; // ③  
  
    }  
}
```

1. ①
2. ②
3. ③
4. ②と③

正解:2

解説

『東京ITスクールJavaテキスト 下巻』第15章/メンバへのアクセス制限/private修飾子

②でnumフィールドにアクセスしていますが、numフィールドのアクセス修飾子はprivateであるため、アクセスすることが出来ません。

したがって、選択肢2が正解です。

問14

次のプログラムにおいてnumフィールドのgetter、setterを定義する場合、①と②にそれぞれ記述すべき適切な処理の組み合わせを選びなさい。※①をgetter、②をsetterとする。

なお、getter、setterは他のどのクラスからもアクセスできるものとします。

【ソースコード】

```
public class A {  
  
    private int num;  
  
    ① {  
  
        return num;  
  
    }  
  
    ② {  
  
        this.num = num;  
  
    }  
  
}
```

1. ①:public int getNum() ②:public void setNum(int num)
2. ①:public int getNum() ②:public int setNum(int num)
3. ①:public void getNum() ②:public void setNum(int num)
4. ①:public void getNum() ②:public int setNum(int num)

正解: 1

解説

『東京ITスクールJavaテキスト 下巻』第15章/アクセス修飾子の利用/アクセサ

getterは他クラスから自クラスのフィールドの値を取得させるためのメソッドですので、戻り値が必要です。アクセス修飾子をpublicにしなければ、他クラスからのアクセスが出来なくなり、getterの意味を成しません。したがって、getterは「`public int getNum()`」が適切な記述となります。

また、setterは他クラスから自クラスのフィールドに値を設定させるためのメソッドですので、戻り値は不要です。その代わり引数が必要になります。こちらも他クラスからのアクセスをさせるために、アクセス修飾子をpublicにする必要があり、「`public void setNum(int num)`」が適切な記述となります。

したがって、選択肢1が正解の組み合わせです。

オーバーロード

問15

`calc()`メソッドをオーバーロードする場合、オーバーロード出来ないメソッドの記述を選びなさい。

【ソースコード】

```
public class A {  
  
    void calc(int num) {  
  
        System.out.println(num * 3);  
  
    }  
  
}
```

1. `void calc()`
2. `void calc(String str)`
3. `int calc()`
4. `int calc(int num)`

正解:4

解説

『東京ITスクールJavaテキスト 下巻』第16章/メソッドのオーバーロード/オーバーロードとは

オーバーロードは異なる引数の型・個数・順番で同じメソッドを定義することであり、戻り値の型は関係がありません。

選択肢1、2、3は引数の型・個数に違いがあり、Aクラス内に追加することが可能です。

選択肢4は、戻り値のみに違いがあり、引数が同じであるためオーバーロードできません。

したがって、選択肢4が正解です。

問16

次のソースコードをコンパイル、実行したい。想定される結果として正しいものを選びなさい。

【ソースコード】

```
public class A {  
  
    void calc(int num) {  
  
        System.out.println("処理1");  
  
    }  
  
    void calc(String num) {  
  
        System.out.println("処理2");  
  
    }  
}
```

```
void calc(float num) {  
  
    System.out.println("処理3");  
  
}  
  
void calc(double num) {  
  
    System.out.println("処理4");  
  
}  
  
}  
  
public class B {  
  
    public static void main(String[] args) {  
  
        A a = new A();  
  
        a.calc("141");  
  
    }  
  
}
```

1. 処理1
2. 処理2
3. 処理3
4. 処理4

正解:2

解説

『東京ITスクールJavaテキスト 下巻』第16章/メソッドのオーバーロード/オーバーロードとは

『東京ITスクールJavaテキスト 下巻』第15章/アクセス修飾子の利用/アクセス修飾子の種類

『東京ITスクールJavaテキスト 下巻』第17章/コンストラクタの基本/コンストラクタの定義

コンストラクタに戻り値は定義することが出来ません。

また、アクセス修飾子はメソッドと同様に定義することが出来ます。

したがって、選択肢2が正解です。

問18

次のソースコードをコンパイル、実行したい。想定される結果として正しいものを選びなさい。

【ソースコード】

```
public class A {  
  
    public A(int num) {  
  
        System.out.print(num);  
  
    }  
  
}  
  
  
public class B {  
  
    public static void main(String[] args) {  
  
        A a = new A(2);  
  
        System.out.print(5);  
  
    }  
  
}
```

1. 2
2. 5
3. 25
4. 52

正解:3

解説

『東京ITスクールJavaテキスト 下巻』第17章/コンストラクタの基本/コンストラクタを呼び出す

コンストラクタは、オブジェクトの生成と同時に動作します。

このコードの場合、最初にコンストラクタの処理が動作することで2が出力され、その後呼び出し元に処理が戻り5が出力されます。

したがって、選択肢3が正解です。

staticメンバ

問19

次のソースコードをコンパイル、実行したい。想定される結果として正しいものを選びなさい。

【ソースコード】

```
public class Test {  
  
    public static void main(String[] args) {  
  
        for(int i = 0; i < 10; i++) {  
  
            new Car();  

```

```
    }  
  
    System.out.println(Car.sum);  
}  
}
```

```
public class Car {  
  
    static int sum = 0;  
  
    public Car() {  
  
        sum++;  
    }  
}
```

1. 0
2. 9
3. 10
4. 11

正解:3

解説

『東京ITスクールJavaテキスト 下巻』第18章/static変数とstaticメソッド

Carクラスのオブジェクトを生成すると、コンストラクタの処理が実行され、そのたびにsumの値が1加算されます。

static変数の値はプログラム全体が実行中の間は保持されます。

今回はCarクラスのコンストラクタが10回呼び出されるため、初期値0のstatic変数sumに結果として10加算されることになります。

したがって、実行結果として「10」が出力されるため、選択肢3が正解です。

問20

下記のStudentクラスを作成するとコンパイルエラーが発生する。その原因として正しいものを選びなさい。

【ソースコード】

```
public class Student {  
  
    private static String job = "学生";  
  
    private String name;  
  
    private int age;  
  
    public static void greet() {  
  
        System.out.println("私の名前は" + name + "と言います。");  
  
        System.out.println("年齢は" + age + "歳です。");  
  
        System.out.println("現在は" + job + "です。");  
  
    }  
}
```

1. jobのアクセス修飾子がpublicではない
2. nameとageに初期値が代入されていない
3. greet()メソッド内でnameとageが呼び出されている
4. greet()メソッド内のjobに「this.」が付いていない

正解:3

解説

『東京ITスクールJavaテキスト 下巻』第18章/staticメソッドの注意点

static変数のアクセス修飾子はpublicでなければならないというルールはありません。

そのため、選択肢1は不正解です。

static変数でもインスタンス変数でも、必ず何らかの値をフィールド宣言時に代入しないといけないというわけではありません。

よって、選択肢2は不正解です。

なお、値の代入処理を書かない場合、String型(参照型)のフィールドにはnull値が、int型のフィールドには整数値0が暗黙的に代入されます。

greet()メソッドはstaticメソッドですので、staticメソッド内ではstatic変数を呼び出せます。

ただし、static変数はオブジェクトに紐づかない情報であるため、「オブジェクトが所有する」という意味を持つ「this」をstatic変数に付けることはできません。

そのため、選択肢4は不正解です。

staticメソッド内ではオブジェクトに紐づく情報を呼び出せないため、インスタンス変数は呼び出せません。

したがって、選択肢3が正解です。

継承

問21

下記のTestクラス中の①に記述できる処理はどれか。正しいものを選びなさい。

【ソースコード】

```
public class Test {  
  
    public static void main(String[] args) {  
  
        D d = new D();  
  
        ①  
  
    }  
}
```

```
public class A {  
  
    public void doA() {  
  
        System.out.println("a");  
  
    }  
}
```

```
public class B extends A {  
  
    private void doB() {  
  
        System.out.println("b");  
  
    }  
}
```

```
public class C extends A {
```

```
public void doC() {  
  
    System.out.println("c");  
  
}  
  
}
```

```
public class D extends B {  
  
    public void doD() {  
  
        System.out.println("d");  
  
    }  
  
}
```

1. 「d.doA()メソッド」と「d.doB()メソッド」
2. 「d.doB()メソッド」と「d.doC()メソッド」
3. 「d.doC()メソッド」と「d.doD()メソッド」
4. 「d.doA()メソッド」と「d.doD()メソッド」

正解: 4

解説

『東京ITスクールJavaテキスト 下巻』第20章/継承/継承とは

『東京ITスクールJavaテキスト 下巻』第20章/protected修飾子

各クラスの継承関係を確認すると、クラスAはクラスBとCのスーパークラス、クラスBはクラスDのスーパークラスであることが分かります。

クラスCとクラスDは継承関係にありません。

よって、クラスDのオブジェクトからはクラスCのメンバは呼び出せないため、選択肢2と3は不正解です。

Java総合試験（問題数:30問、想定所要時間:60分）

また、クラスB中のdoB()メソッドにprivateが付与されていることが分かります。

privateなメンバは継承関係の有無に関わらず他のクラスからは呼び出せません。

そのため、選択肢1は不正解です。

したがって、呼び出せるメソッドはdoA()メソッドとdoD()メソッドのみであるため、選択肢4が正解です。

問22

次のソースコードをコンパイル、実行したい。想定される結果として正しいものを選びなさい。

【ソースコード】

```
public class Test {  
  
    public static void main(String[] args) {  
  
        B b = new B();  
  
    }  
}
```

```
public class A {  
  
    public A() {  
  
        System.out.print(" a");  
  
    }  
  
}
```

```
public A(int a) {  
  
    System.out.print(a);  
  
}
```



```
}  
}
```

```
public class B extends A {  
  
    public B() {  
  
        super(10);  
  
        System.out.print("b");  
  
    }  
  
}
```

```
    public B(int b) {  
  
        super();  
  
        System.out.print(b);  
  
    }  
  
}
```

1. a10
2. 10b
3. b10
4. ab

正解: 2

解説

『東京ITスクールJavaテキスト 下巻』第20章/継承/明示的なスーパークラスのコンストラクタ呼び出し

Java総合試験（問題数:30問、想定所要時間:60分）

AクラスはBクラスのスーパークラスです。

Bクラスのコンストラクタを実行する際は、事前にAクラスのコンストラクタが実行されます。

main()メソッドではBクラスのオブジェクト生成時に引数なしのコンストラクタを呼び出しています。

Bクラスの引数なしのコンストラクタ内では、super(10)という処理でAクラスの引数ありのコンストラクタを呼び出しています。

その後、Bクラスのコンストラクタ内独自の処理が実行されます。

よって、コンストラクタは「A(int a)→B()」という流れで実行されます。

したがって、コンソール上には「10」、「b」の順に出力されるため、選択肢2が正解です。

インターフェイス

問23

下記のHumanインターフェイスをコンパイルすると、コンパイルエラーが発生する。インターフェイスブロック内でエラーの原因となる文は何行あるか。正しい行数を選びなさい。

【ソースコード】

```
public interface Human {  
  
    int LEG_NUMBER = 2;  
  
    int ARM_NUMBER;  
  
    public Human();  
  
    public void walk();  
  
}
```

1. 1行

- 2. 2行
- 3. 3行
- 4. 4行全て

正解:2

解説

『東京ITスクールJavaテキスト 下巻』第22章/インターフェイス/インターフェイスの定義

コンパイルエラーの原因となるのは、2行目の「`int ARM_NUMBER;`」と3行目の「`public Human();`」です。

インターフェイス内のフィールドは全て `static final` が暗黙的に付与されます。いわゆる「定数」という扱いとなります。

そのため、インターフェイス内のフィールドには宣言時に必ず初期値を指定する必要があります。

よって、2行目の「`int ARM_NUMBER;`」は誤りです。

インターフェイスからオブジェクトは生成できません。

そのため、インターフェイス内でコンストラクタを定義することは仕様上できません。

よって、3行目の「`public Human();`」も誤りです。

その他の文は正しい記述です。

したがって、選択肢2が正解です。

問24

以下のクラスをコンパイルしようとする、コンパイルエラーが発生する。コンパイルエラーの原因として正しいものを選びなさい。

【ソースコード】

```
public class Test {  
  
    public static void main(String[] args) {  
  
        B b = new B();  
  
    }  
}
```

```
public interface A {  
  
    public void doA();  
}
```

```
public class B implements A {  
  
    public void doB() {  
  
    }  
}
```

1. doA()メソッドに「{ }」のブロックが記載されていない
2. main()メソッド内で、doB()メソッドが呼び出されていない
3. Bクラス中でdoA()メソッドが定義されていない
4. doB()メソッドの「{ }」内に具体的な処理が記載されていない

正解:3

解説

『東京ITスクールJavaテキスト 下巻』第22章/インターフェイス/インターフェイスの実装

インターフェイス内のメソッドは全て抽象メソッドとして扱われるため、「{ }」を記述できません。

そのため、選択肢1は不正解です。

また、実装クラスのオブジェクトを生成したら、そのクラス内のメンバを呼び出さなければならないというルールはありません。

よって、選択肢2は不正解です。

doB()メソッドのブロック内には何も処理が記載されていませんが、文法上は問題ないため、コンパイルエラーの発生原因にはなりません。

そのため、選択肢4は不正解です。

インターフェイス内で抽象メソッドが定義されている場合、そのインターフェイスを実装したクラス内では抽象メソッドに対する具象メソッド(具体的な処理を記述できるメソッド)を定義する必要があります。

したがって、選択肢3が正解です。

パッケージ

問25

パッケージ「test.pkg1」内に含まれるクラスがある。このクラスのpackage文として正しいものを選びなさい。

1. package pkg1;
2. package /test/pkg1;
3. package test.pkg1;
4. package test¥pkg1;

正解:3

解説

『東京ITスクールJavaテキスト 下巻』第24章/パッケージ/パッケージとは

『東京ITスクールJavaテキスト 下巻』第24章/パッケージ/異なるパッケージのクラスを使う

階層構造になっているパッケージに対するpackage文を記述する際は、最上位階層から過不足なくパッケージを指定する必要があります。

そのため、選択肢1は不正解です。

また、各パッケージの間には「.(ピリオド)」を記述します。

そのため、選択肢2と4は不正解です。

したがって、選択肢3が正解です。

問26

下記のパッケージ構造内に存在するTestクラスをインポートするimport文として正しいものを選びなさい。

pkg

├─ sub1

└─ sub2

 ├─ sub3

 └─ sub4

 └─ Test.java

1. import pkg.Test;

2. `import pkg.sub1.sub4.Test;`
3. `import pkg.sub2.sub4.Test;`
4. `import pkg.sub2.sub3.sub4.Test;`

正解：3

解説

『東京ITスクールJavaテキスト 下巻』第24章/インポート/インポートとは

サブパッケージ内のクラスをインポートする際は、`import`文で最上位階層から過不足なくパッケージの階層を明記する必要があります。

そのため、選択肢1は不正解です。

パッケージを指定する際はパッケージ同士の階層構造を正しく記述する必要があります。

そのため、選択肢2と4は不正解です。

したがって、選択肢3が正解です。

例外

問27

以下のソースコード中の①～②に適切な記述を入れて、コンパイルエラーが発生しないソースコードを完成させたい。

①～②に記述する内容の組み合わせとして、正しいものを選びなさい。

【ソースコード】

```
import java.io.BufferedReader;
```

```
import java.io.IOException;
```

```
import java.io.InputStreamReader;

public class Test {

    public static void main(String[] args) {

        try {

            BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

            String message = reader.readLine();

            System.out.println(message);

        } catch (① e) {

            System.out.println("・・・");

        } ② {

            System.out.println("・・・");

        }

    }

}
```

1. ①:Exception ②:finally
2. ①:IOException ②:final
3. ①:NumberFormatException ②:finally
4. ①:Exception ②:final

正解: 1

解説

『東京ITスクールJavaテキスト 下巻』第25章/例外クラス

『東京ITスクールJavaテキスト 下巻』第25章/例外処理/try-catch-finally文

catchブロックの後に、catchブロックまたはfinallyブロックを続けて記述できます。

今回のソースコード中の②では例外クラスの引数を指定する箇所がないため、②にはfinallyが入ることが分かります。

選択肢2と4では、②にfinallyではなくfinalと記述することになっているため、コンパイルエラーが発生します。

そのため、選択肢2と4は不正解です。

また、今回のソースコードでは、BufferedReaderクラスのreadLine()メソッドを実行しているため、IOExceptionをcatchするか、throwsする必要があります。

ソースコード中ではthrow分は記述されておらず、catchブロックが記述されているため、catchするということが分かります。

catchする場合、catchブロックの引数には送出される例外クラス、またはそのクラスのスーパークラスである例外クラスを指定します。

NumberFormatExceptionは例外クラスではありますが、IOExceptionとは継承関係にないため、コンパイルエラーが発生します。

そのため、選択肢3は不正解です。

IOExceptionはExceptionクラスを継承している関係にあります。

したがって、選択肢1が正解です。

問28

下記のTestクラス中のmain()メソッド内で、CheckNumberクラスのcheckOver10()メソッドを呼び出したい。

その場合、ソースコード中の①～②に記述する内容の組み合わせとして、正しいものを選びなさい。

【ソースコード】

```
public class Test {  
  
    public static void main(String[] args) ① {  
  
        CheckNumber.checkOver10(10);  
  
    }  
}  
  
public class CheckNumber {  
  
    public static void checkOver10(int number) ① {  
  
        if (number <= 10) {  
  
            Exception e = new Exception("数値が10以下です");  
  
            ② e;  
  
        }  
  
    }  
}
```

1. ①:throw Exception ②:throws
2. ①:throws Exception ②:throw
3. ①:なにも書かない ②:throw
4. ①:なにも書かない ②:throws

正解:2

解説

『東京ITスクールJavaテキスト 下巻』第25章/例外を投げる/throws文

『東京ITスクールJavaテキスト 下巻』第25章/例外を投げる/throw文

Java総合試験（問題数:30問、想定所要時間:60分）

checkOver10()メソッド内の処理で意図的に例外を送出する場合は、例外クラスのオブジェクトを生成し、それを参照する変数を送出する処理を記述します。

例外の送出には「throw」と使用します。

そのため、②には「throw」を記述するため、選択肢1と4は不正解です。

また、例外を送出するメソッドでは、「どの種類の例外クラスを送出する可能性があるか」をthrowsで指定するか、try～catchブロックによる例外処理を記述しておく必要があります。

checkOver10()メソッドでは、tryブロックは記述されていないため、throwsを使用することになります。

そのため、①には「throws Exception」を記述する必要があり、選択肢3は不正解です。

また、checkOver10()メソッドを呼び出したメソッド内でも、送出された例外に対する処理を記述しておく必要があります。

main()メソッド内にもtryブロックは記述されていないため、throwsを使用しなければなりません。

したがって、選択肢2が正解です。

コレクションフレームワーク

問29

次のソースコードをコンパイル、実行したい。想定される結果として正しいものを選びなさい。

【ソースコード】

```
import java.util.ArrayList;

import java.util.List;

public class Test {

    public static void main(String[] args) {
```

```
List<Integer> list = new ArrayList<>();

list.add(10);

list.add(20);

list.add(30);


list.remove(1);


System.out.println(list.get(0));

System.out.println(list.get(1));

}

}
```

1. 10
20
2. 10
0
3. 10
30
4. 実行時に例外が送出される

正解:3

解説

『東京ITスクールJavaテキスト 下巻』第26章/ArrayListクラス/要素の削除

remove()メソッドは、「()」内で指定されたインデックスの要素そのものを削除します。

削除後は、削除された要素に後続する要素は、1つ手前に詰められます。

今回のソースコードでは、インデックス1番の要素を削除しています。

すると、インデックス2番の要素が1つ手前に詰められるため、結果的に要素2個からなるリストとなります。

したがって、選択肢3が正解です。

問30

次の実行結果となるように、①に入れるコードとして正しいものを選びなさい。

【実行結果】

2

【ソースコード】

```
import java.util.HashMap;
```

```
import java.util.Map;
```

```
public class Test {
```

```
    public static void main(String[] args) {
```

```
        ①
```

```
        map.put("a", 1);
```

```
        map.put("b", 2);
```

```
        map.put("c", 3);
```

```
        System.out.println(map.get("b"));  
    }  
}
```

1. Map<String,int> map = new HashMap<>();
2. Map<String,Integer> map = new HashMap<>();
3. Map<String> map = new HashMap<>();
4. Map<String,String> map = new HashMap<>();

正解:2

解説

『東京ITスクールJavaテキスト 下巻』第26章/HashMapクラス/マップの生成

Mapは1つの要素に「キー」と「値」の2種類の情報を保存します。

Mapのジェネリクスでは、キーと値それぞれの型を参照型で指定する必要があります。

選択肢1は、「値」の型intが参照型ではなく基本型となっています。

選択肢3は、型が1つしか指定されていません。

選択肢4は、「値」の型が文字列となっているため一致しません。

これら3つの選択肢の結果は、どれもコンパイルエラーとなります。

したがって、選択肢2が正解です。