|  |
| --- |
| 0001_tis_logo.png |
| Java 受講前試験 |
| **（模範解答あり)** |
|  |
| **制限時間：60分** |
|  |

|  |
| --- |
| ※問題数は30問、制限時間は60分です。 |

以下のカテゴリに関する問題を、3問ずつ出題します。

受講前試験、受講後試験の試験結果は下記カテゴリに沿ってレーダーチャートで紹介します。

1. 変数と型
2. 式と演算子
3. 配列
4. 条件分岐
5. 反復処理
6. クラス基本（オブジェクト指向、クラス、フィールド、メソッド）
7. メソッド基本（アクセス制御、オーバーロード）
8. クラス応用（コンストラクタ、staticメンバ、クラスライブラリ）
9. 継承
10. インターフェイス

# 変数と型

## 問1

数値を扱う変数の初期化方法として、正常に動作するものを選びなさい。

1. int num = 4.0;
2. boolean num = 128;
3. float num = 2.5;
4. double num = 3;

正解：4

解説

『東京ITスクールJavaテキスト 上巻』第4章/変数を使う/変数の初期化

Javaは浮動小数点数リテラルをdouble型として扱います。

したがって、float型の変数に浮動小数点数を代入する場合は、末尾にfをつける必要があります。選択肢3は誤りです。

double型は浮動小数点数を扱うデータ型ですが、整数値を代入すると、値が暗黙的な型変換により浮動小数点数に変換されます。

したがって、選択肢4が正解です。

その他の選択肢は、型の不一致によりコンパイルエラーとなります。

## 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】

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

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3

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。

# 式と演算子

## 問4

以下の式はどのような処理を行うか。正しいものを選べ。

"答え：" + 2 + 10

1. 数値2と数値10の足し算が行われる。次に、その数値と文字列「答え：」が文字列連結する。

2. 文字列「答え：」と数値2が文字列連結する。次に、数値2と数値10の足し算が行われる。

3. 文字列「答え：」と数値2が文字列連結する。次に、その文字列と数値10が文字列連結する。

4. 型が異なる値で計算しようとするため、エラーになる。

正解：3

解説

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

「+」の演算子は2つのオペランドを持つことで、2種類の演算子のいずれかの機能を持ちます。

2つのオペランドが両方とも数値の場合は加算のための演算子になります。もしくは、2つのオペランドの内、片方でも文字列だった場合は文字列連結のための演算子となります。

また、「+」は左結合の特性を持っており、「+」が1つの式で複数回記述されている場合、左側に書かれた演算子から処理されます。

そのため、問題の文では、まず「答え：」という文字列と数値2の間で文字列連結が行われ、次に「答え：2」という文字列と数値10の文字列連結が行われます。

そして、この式の結果は「答え：210」となります。

そのため、選択肢3が正解です。

## 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) {

int number1 = 9;

int number2 = 10;

int number3 = ++number1;

int number4 = number2++;

System.out.println(number3 - number4);

}

}

1. 0
2. 1
3. 2
4. 3

正解：1

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/インクリメントとデクリメントの前置・後置

「int number3 = ++number1;」の処理では、最初にnumber1に1が加算され、その後にnumber1の値がnumber3に代入されます。

このときの各変数の値は下記になります。

number1：10

number2：10

number3：10

次に、「int number4 = number2++;」の処理では、最初にnumber2の値がnumber4に代入され、その後にnumber2の値に1が加算されます。

このときの各変数の値は下記になります。

number1：10

number2：11

number3：10

number4：10

結果、「number3 – number4」の値は「0」となります。

したがって、選択肢1が正解です。

## 

## 問6

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と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7

以下の文はどのような処理を行うか。正しいものを選べ。

ただし、変数numには配列の情報が代入されているものとする。

System.out.println(num.length);

1. 配列の要素数を出力する。
2. 配列に保管された文字列の総文字数を出力する。
3. 配列の添字の最大値を出力する。
4. 配列の型のサイズを出力する。

正解：1

解説

『東京ITスクールJavaテキスト』第7章/配列の長さ

配列変数に続けて「.length」と記述することで、参照している配列の要素数を求めることができます。そのため、選択肢1が正解です。

## 

## 問8

5個の要素を持った配列の初期化方法として、正しいものを選びなさい。

1. int[] array = int[5];
2. int[] array = {10, 5, 3, 8, 7};
3. int[] array = new int[5];
4. int[] array = int{1, 5, 6, 7, 3};

正解：2

解説

『東京ITスクールJavaテキスト 上巻』第7章/配列の準備/配列の初期化

配列の初期化は「型[] 変数名 = {値1, 値2, ・・・};」という形式で記述します。

したがって、選択肢2が正解です。

なお、選択肢3は「配列変数の宣言」と「要素の確保」を1行にまとめた処理ですが、値の代入までは行っていないため、配列の初期化ではありません。

## 

## 問9

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;

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。

# 条件分岐

## 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 number1 = 10;

int number2 = 10;

if (number1 > number2) {

System.out.println("条件1");

} else if (number1 < number2) {

System.out.println("条件2");

}

}

}

1. 条件1
2. 条件2
3. 何も出力されない
4. コンパイルエラーになる

正解：3

解説

『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とは

1つ目の条件「number1 > number2」はnumber1とnumber2が同値なので、条件を満たしません（false）。

2つ目の条件「number1 < number2」もnumber1とnumber2が同値なので、条件を満たしません（false）。

構文には誤りがないためコンパイルは実行されます。

したがって、選択肢3が正解です。

## 問11

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の【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、該当する選択肢の値は7のみとなるため、選択肢3が正解です。

## 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[]) {

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13

次のソースコードの処理を意図的に無限ループさせるには、①にどのような内容を記述するべきか。

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文で無限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。

そのため、常に「i <= 0」の条件を満たすことになり、無限ループが発生します。

したがって、選択肢4が正解です。

## 問14

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。

## 問15

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 = 0;

do {

System.out.println(number);

number++;

} while (number < 0);

}

}

1. 実行時に例外が送出される
2. 何も表示されない
3. 無限ループが発生する
4. 0

正解：4

解説

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

ソースコードは構文上正しく、記述内容に起因して例外が発生する可能性もありません。

そのため、選択肢1は不正解です。

変数numberの初期値は0です。

それに対して、do～while文の条件は「number < 0」であるため、初期値がそもそも条件を満たしていないことが分かります。

しかし、do～while文は無条件で最低1回は繰り返し処理が実行されます。

そのため、今回は「0」が必ず表示されるため、選択肢2は不正解です。

繰り返し処理中では、出力処理後に変数numberの値を1ずつ加算しています。

そのため、1回目の繰り返し処理後の条件判定はfalseとなり、do～while文の処理が終了し、無限ループとならず「0」のみが出力されることになります。

したがって、選択肢4が正解です。

# クラス基本

# （オブジェクト指向、クラス、フィールド、メソッド）

## 問16

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;

1. car.num() = 5248;

car.gas() = 21.5;

1. num = car(5248);

gas = car(21.5);

1. 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17

次のTestクラス内で、Aクラスのオブジェクトからdisp()メソッドを実行し、次の実行結果を出力させたい。①に記述すべき内容として正しいものを選びなさい。

【実行結果】

Hello

【ソースコード】

public class Test {

public static void main(String[] args) {

A a = new A();

a.disp("Hello");

}

}

public class A {

① {

System.out.println(hoge);

}

}

1. void disp()
2. void disp(String hoge)
3. String disp()
4. String disp(String hoge)

正解：2

解説

『東京ITスクールJavaテキスト 下巻』第14章/メソッドの基本/メソッドを使用する

『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/戻り値を持たないメソッド

disp()メソッドを呼び出す処理において、引数が"Hello"となっていることから、disp()メソッドにはString型の引数を定義する必要があります。

また、disp()メソッド内にはreturn文がないことから、戻り値はvoidとなります。

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 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。

# メソッド基本（アクセス制御、オーバーロード）

## 問19

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をしていますが、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20

次のプログラムにおいてnumフィールドのsetterを定義する場合、①に記述すべき、適切な処理を選びなさい。

なお、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 {

this.num = num;

}

}

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

正解：2

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

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。

また、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、setterの意味を成しません。

したがって、選択肢2が正解です。

## 問21

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

オーバーロードは、引数の型・個数に応じたメソッドが自動的に呼び出されます。

今回の場合は"141"というString型の引数をメソッドに渡しているため、void calc(String num)が動作します。

したがって、選択肢2が正解です。

# クラス応用

# （コンストラクタ、staticメンバ、クラスライブラリ）

## 問22

次のプログラムをコンパイルしたところ、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。

【ソースコード】

public class A {

private A() { // ①

}

public A(String str) { // ②

System.out.print(str);

}

}

public class B {

public static void main(String[] args) {

A a = new A(); // ③

}

}

1. ①
2. ②
3. ③
4. ①と③

正解：3

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/コンストラクタのオーバーロード/コンストラクタのオーバーロード

Aクラス内ではコンストラクタがオーバーロードされています。

ただし、アクセス修飾子がprivateとなっているコンストラクタは他のクラスから呼び出すことが出来ません。

したがって、選択肢3が正解です。

## 問23

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。

## 問24

以下の文を実行すると、変数dog1と変数dog2はどのような状態となるか。正しいものを選べ。

Dog dog1 = new Dog();

Dog dog2;

dog2 = dog1;

1. 変数dog1から変数dog2にオブジェクトそのものが渡され、2つの変数が別々のオブジェクトを参照している状態となる。

2. 変数dog1から変数dog2にオブジェクトを参照するための情報が渡され、2つの変数が1つのオブジェクトを参照している状態となる。

3. クラス型の変数同士の間では値を渡すことができないため、変数dog1のみがオブジェクトを参照している状態となる。

4. 変数dog1から変数dog2にオブジェクトのフィールド情報のみが渡され、変数dog1はオブジェクトを参照し、変数dog2にはフィールドの値が保存されている。

正解：2

解説

『東京ITスクールJavaテキスト』 第19章 クラス型の変数/値渡しと参照渡し

クラス型の変数の中にはオブジェクトを参照するための情報が保存されています。

そのため、クラス型の変数の間で代入を行った場合、オブジェクトを参照するための情報がコピーされることになります。

その結果、2つの変数はどちらも同じ1つのオブジェクトを参照している状態となります。したがって、選択肢2が正解です。

なお、このように参照するための情報を変数同士で受け渡すことを「参照渡し」と呼びます。

また、int型の数値など、実際の値を直接受け渡すことを「値渡し」と呼びます。

# 継承

## 問25

スーパークラスとサブクラスに関する説明として正しいものはどれか。

1. スーパークラスからはオブジェクトを生成できない。また、サブクラスからはオブジェクトを生成できる。

2. スーパークラスとサブクラスは必ず同じクラス名で定義する。

3. サブクラスはスーパークラスのメンバを自分のメンバとして利用できる。

4. スーパークラスとサブクラスの記述方法に違いはない。

正解：3

解説

『東京ITスクールJavaテキスト』 第20章 継承/継承とは

2つのクラスが継承関係にある時、継承元のクラスを「スーパークラス」、継承先のクラスを「サブクラス」と呼びます。

サブクラスはスーパークラスが持つメンバを、あたかも自分のクラス内で定義されたメンバのように利用することができます。したがって、選択肢3が正解です。

スーパークラスとサブクラスはどちらも通常のクラスとして利用でき、オブジェクトも生成できます。そのため、選択肢1は不正解です。

また、スーパークラスとサブクラスのクラス名は異なる名前でも問題ありません。そのため、選択肢2は不正解です。

また、サブクラスは、「class サブクラス名 extends スーパークラス名」という形式で定義しなければいけません。

この点はスーパークラスとの記述方法の違いになります。そのため、選択肢4は不正解です。

## 問26

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。

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27

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

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28

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処理（文）は何行分あるか。正しい個数を選びなさい。

【ソースコード】

public interface Human {

int legNumber = 2;

int armNumber;

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 armNumber;」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 armNumber;」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という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。

## 問29

以下のクラスをコンパイルしようとすると、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。

## 問30

下記のDクラス内で定義しなければならないメンバはどれか。正しいものを選びなさい。

【ソースコード】

public interface A {

public void doA();

}

public interface B extends A {

public void doB();

}

public interface C {

public void doC();

}

public class D implements B, C {

}

1. doA()メソッド、doB()メソッド、doC()メソッド、doD()メソッド
2. doA()メソッド、doB()メソッド、doC()メソッド
3. doB()メソッド、doC()メソッド
4. doA()メソッド、doC()メソッド

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実装

『東京ITスクールJavaテキスト 下巻』第24章/インターフェイスの拡張

DクラスはBインターフェイス、Cインターフェイスを実装しています。

そのため、BインターフェイスとCインターフェイス内で定義された抽象メソッド全てに対して具象メソッドを定義する必要があります。

つまり、doB()メソッド、doC()メソッドは定義しなければなりません。

また、BインターフェイスはAインターフェイスと継承関係にあります。

Bインターフェイスがサブインターフェイスの位置付けにあたるため、Bインターフェイスを実装したクラスは、スーパーインターフェイスであるAインターフェイス中の抽象メソッドに対する具象メソッドも定義する必要があります。

また、doD()メソッドはどのインターフェイス内でも定義されていませんので、Dクラス内で定義する必要はありません。

したがって、選択肢2が正解です。