

1、 B

問題 次のプログラムの3行目にインスタンスを生成するコードを記述したい。
正しいものを選びなさい。(1つ選択)

```
1 | public class Main {  
2 |     public static void main(String[] args) {  
3 |         // インスタンス生成のコード  
4 |     }  
5 | }  
6 | class ClassA {  
7 |     // クラス内の定義  
8 | }
```

- ☐ A.ClassA classA = new ClassA;
- ☒ B.ClassA classA = new ClassA();
- ☐ C.ClassA classA = create ClassA();
- ☐ D.ClassA classA = create ClassA;

クラスのインスタンス化に関する問題です。

クラスは「型」の定義であり、そのままでは利用することはできません。

定義されたクラスを基にして、インスタンスを生成して初めて利用することができます。

インスタンスを生成することを「インスタンス化」と呼びます。

インスタンスを生成するには、次のように記述します。

書式

クラス名 変数名 = new クラス名();

クラス名 変数名 = new クラス名(引数);

「クラス名」は、インスタンス化するクラスの名前を記述します。

インスタンス生成のコードの例を示します。

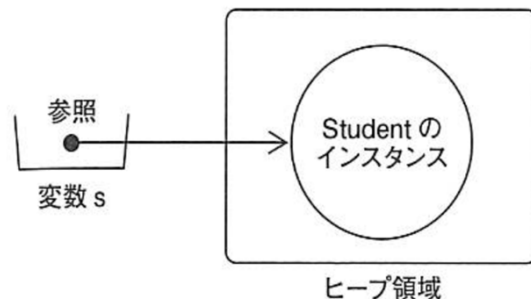
例：インスタンスの生成

```
1. public class Sample {  
2.     public static void main(String[] args) {  
3.         Student s = new Student();  
4.     }  
5. }  
6.  
7. class Student {  
8. }
```

コードの3行目では、Studentクラスのインスタンスを生成しています。

このとき、変数sにインスタンスそのものが入るわけではないことに注意しましょう。
インスタンスは別の領域(ヒープ)に作られるため、変数にはその領域にアクセスするためのリンク情報が
入ります。
このリンク情報のことを「参照」と呼びます。
参照型変数の中には、インスタスではなく参照が入っていることを忘れないようにしましょう。

【参照型変数は参照を保持】



以上のとおり、選択肢Bが正解です。

その他の選択肢は以下の理由により誤りです。

- A. カッコが付いていません。
- C.D. インスタンスを生成するキーワードはcreateではなくnewです。

2、 A

問題 次のプログラムの4行目にメソッド呼び出しのコードを記述したい。
正しいものを選びなさい。(1つ選択)

```
1 | public class Main {  
2 |     public static void main(String[] args) {  
3 |         ClassA classA = new ClassA();  
4 |         // メソッド呼び出しのコード  
5 |     }  
6 | }  
7 | class ClassA {  
8 |     void doMethod() {  
9 |         // メソッド内の処理  
10 |    }  
11 | }
```

- ☒ A.classA.doMethod();
- ☐ B.classA+doMethod();
- ☐ C.classA doMethod();
- ☐ D.classA:doMethod();

インスタンスのメソッド呼び出しに関する問題です。メソッド呼び出しは、次のように記述します。

書式

変数名.メソッド名(引数);

「変数名」は、インスタンスへの参照を保持した変数を指定します。その後ろにドット「.」を続け、呼び出したい「メソッド名」を指定します。「引数」が必要な場合は、カッコ「()」内に引数を記述します。

以上のことから、選択肢Aが正解です。

選択肢Bのプラス記号「+」、選択肢Cの空白、選択肢Dのコロン「:」は、メソッド呼び出しには仕様できません。

次のコードは、メソッド呼び出しの例です。3行目で、Roomクラスのインスタンスを生成し、その参照を変数rに代入しています。4行目で、変数rが保持している参照先のインスタンスのprintメソッドを呼び出しています。

例：メソッド呼び出し

```
1. public class Sample {  
2.     public static void main(String[] args) {  
3.         Room r = new Room(203);  
4.         r.print();  
5.     }  
6. }  
7.  
8. class Item {  
9.     private int number;  
10.  
11.     Room(int number) {  
12.         this.number = number;  
13.     }  
14.     void print() {  
15.         System.out.println(number + "号室");  
16.     }  
17. }
```

例：実行結果

203号室

問題 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。(1つ選択)

```
1 | public class Main {  
2 |     public static void main(String[] args) {  
3 |         Document d = new Document("James");  
4 |         d.printOwner();  
5 |     }  
6 | }  
7 | class Document {  
8 |     private String owner;  
9 |     public Document() {  
10 |         this.owner = "none";  
11 |     }  
12 |     public Document(String owner) {  
13 |         this.owner = owner;  
14 |     }  
15 |     public void printOwner() {  
16 |         System.out.println(owner);  
17 |     }  
18 | }
```

- ☐ A.何も表示されない。
- ☐ B.「none」と表示される。
- ☒ C.「James」と表示される。
- ☐ D.コンパイルエラーになる。

オーバーロードされたコンストラクタの呼び出しに関する問題です。

オーバーロードされたメソッドを呼び出す際、引数の型や数、順番によって呼び出されるメソッドが選択されます。これはメソッドの一種であるコンストラクタも同様です。

設問のコード8～20行目のDocumentクラスには、引数なしのコンストラクタ(11～13行目)とString型の引数1つを受け取るコンストラクタ(14～16行目)が定義されています。

3行目ではDocumentクラスのインスタンスを生成しています。

コンストラクタにString型のデータを渡してインスタンス化しているので、14～16行目で定義しているString型のデータを1つ受け取るコンストラクタが呼び出されます。

コンストラクタブロック内では、引数のデータ"James"をフィールドに代入しています。

4行目でprintOwnerメソッドを呼び出し、フィールドownerが保持しているデータをコンソールに出力します。

したがって、選択肢Cが正解です。

問題 コンストラクタの定義に関する説明として正しいものを選びなさい。(1つ選択)

- ☐ A. コンストラクタの戻り値型はvoidにする。
- ☐ B. コンストラクタには、引数を定義できない。
- ☒ C. コンストラクタ名は、クラス名と同じにする。
- ☐ D. コンストラクタには、戻り値型を指定できる。

コンストラクタ定義の規定に関する問題です。

コンストラクタは、生成したインスタンスがほかのインスタンスから使われる前に事前の準備をするためのメソッドです。

コンストラクタの定義は、次のように記述します(選択肢B)。

書式

```
アクセス修飾子 コンストラクタ名(引数) {  
    // インスタンスの初期化处理  
}
```

コンストラクタに関しては、次の3つのルールがあります。

1. コンストラクタ名は、クラス名と同じでなければいけない(選択肢C)
2. 戻り値型は記述できない(選択肢A、D)
3. インスタンス生成時にしか使えない

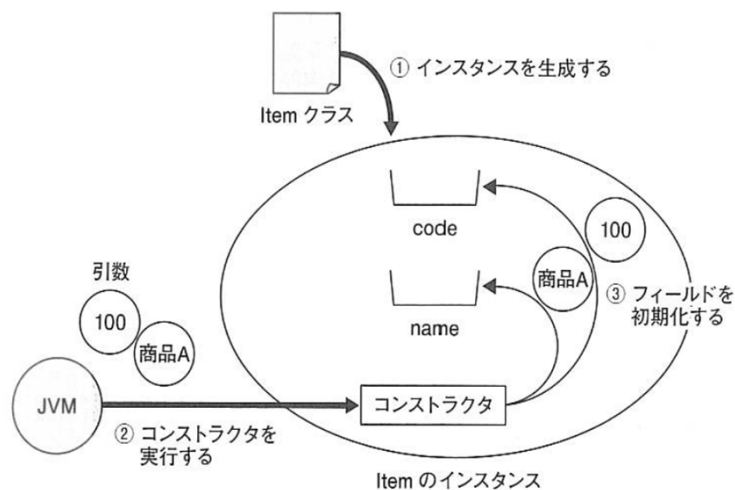
次のコード例では、インスタンスを生成して、13～16行目の定義に基づいてコンストラクタでフィールドを初期化するなど、インスタンスの準備をしています。

例：コンストラクタの定義とインスタンス化

```
1. public class Sample {  
2.     public static void main(String[] args) {  
3.         Item item = new Item(100, "商品A");  
4.         int code = item.getCode ();  
5.         String name = item.getName();  
6.         System.out.println(code + " " + name);  
7.     }  
8. }  
9. class Item {  
10.     private int code;  
11.     private String name;  
12.  
13.     public Item(int code, String name) {  
14.         this.code = code;  
15.         this.name = name;  
16.     }  
17.     // any code  
18. }
```

上記のmainメソッドの動作を図で表すと次のようになります。

【Itemクラスのコンストラクタを利用してインスタンス化したイメージ】



前述のとおり、コンストラクタはインスタンスの事前準備をするためのメソッドです。そのため、図のようにまずインスタンスが作られ、その後コンストラクタが実行されます。コード例でコンストラクタの引数に「100」と「商品A」というデータを渡しています。Itemのコンストラクタはこのデータを使ってフィールドの値を初期化しています。その後、このインスタンスへの参照が変数itemに代入されます。

問題 次のクラスに定義するコンストラクタとして、正しいものを選びなさい。(1つ選択)

```
1 | public class Account {  
2 |     private int balance;  
3 | }
```

- ☐ A. public void Account(int balance) {
 this.balance = balance;
}
- ☒ B. public Account() {
 this.balance = 0;
}
- ☐ C. public static Account() {
 this.balance = 0;
}
- ☐ D. public Constructor(int balance) {
 this.balance = balance;
}

コンストラクタの定義に関する問題です。

各選択肢については以下のとおりです。

- A. 戻り値型としてvoidを記述しているため、コンストラクタではなく、メソッド定義です。
- B. コンストラクタのルールに沿って、正しく記述されています。
- C. staticキーワードは、コンストラクタには記述できません。このコードはコンパイルエラーになります。
- D. コンストラクタはクラス名と同一でなければいけません。このコードはコンパイルエラーになります。

したがって、選択肢Bが正解です。