

- ・暗記事項を参考にして予習ノートの作成
- ・予習範囲の熟読
- ・暗記事項の暗記
- ・教科書の演習問題を理解

上記を済ませてから練習問題に取り組んでください。

最終ページに解答を載せています。

## 予習範囲

P.18 の 2.3 節のメソッドから 2 章の終わりまで (章末問題を含む) です。

## 暗記事項

【暗記事項 1】 から【暗記事項 40】 までは、すべて、「A: 正しい」が正答となります。予習確認テストでは表現を変えて「B: 間違っている」が答えになることもあります。

ただ覚えるだけでなく、教科書中の当該事項を説明した文章を読んで理解につとめてください。

- 【暗記事項 1】 メソッド定義の先頭行は、「アクセス修飾子 戻り値の型 メソッド名(引数リスト)」となる。
- 【暗記事項 2】 メソッド名は通常、小文字から始める。
- 【暗記事項 3】 アクセス修飾子は「public, private, protected, 指定なし」の 4 種類がある。
- 【暗記事項 4】 メソッドの戻り値の型が void の時、そのメソッドは値を返さない。
- 【暗記事項 5】 メソッドの戻り値の型として、クラス名を書くことができる。
- 【暗記事項 6】 メソッドの戻り値の型が、クラス名であるとき、オブジェクト (厳密にはその位置) を返す。
- 【暗記事項 7】 メソッド名は小文字から始める。
- 【暗記事項 8】 メソッド先頭行の「アクセス修飾子 戻り値の型 メソッド名 (引数リスト)」の後ろの{}の間にコード (処理) を書く
- 【暗記事項 9】 クラス内でアクセス修飾子が付くのは、フィールド・メソッド・コンストラクタである。
- 【暗記事項 10】 戻り値を返すときには return 文を使う
- 【暗記事項 11】 戻り値の型と return 文で戻した変数の型が整合しないとエラーになる (整合=型が一致または拡大変換)。
- 【暗記事項 12】 戻り値の型が void の時に、return 文があってもよい。
- 【暗記事項 13】 戻り値の型が void ではないときは、return 文が必要である。
- 【暗記事項 14】 メソッド・コンストラクタの引数リストで引数がないときは括弧()だけを書く。
- 【暗記事項 15】 引数がある時は(型名 引数名)と書く
- 【暗記事項 16】 引数が二個以上の場合、カンマ「,」を引数の間に入れる。
- 【暗記事項 17】 コンストラクタでは、メソッド名に相当するところにクラス名が入る
- 【暗記事項 18】 コンストラクタの定義では「戻り値の型名」を書かない。
- 【暗記事項 19】 コンストラクタは new 演算子によりオブジェクトを生成した直後に呼び出される。
- 【暗記事項 20】 コンストラクタの定義ではアクセス修飾子を伴うことがある。
- 【暗記事項 21】 フィールドは、クラス内でメソッドとコンストラクタ以外のところで宣言された変数である。
- 【暗記事項 22】 フィールドはオブジェクトが生成されたときに作られる。
- 【暗記事項 23】 フィールドにはアクセス修飾子が付くことがある。

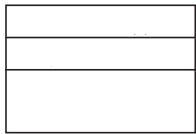
- 【暗記事項 24】 局所変数のことをローカル変数と呼ぶことがある。
- 【暗記事項 25】 局所変数はメソッド・コンストラクタの中で宣言された変数である。
- 【暗記事項 26】 局所変数にはアクセス修飾子は付かない
- 【暗記事項 27】 メソッド・コンストラクタ内にあるフィールドと同名の「引数，局所変数」はフィールドではなく，引数・局所変数として解釈される。
- 【暗記事項 28】 メソッド・コンストラクタ内にフィールドと同名の引数・局所変数が無ければ，それはフィールドとして解釈される。
- 【暗記事項 29】 キーワード `this` を使って `this.a` と書けば変数 `a` はフィールドである。
- 【暗記事項 30】 キーワード `this` は現在のオブジェクトのフィールド・メソッドを参照するために用いる
- 【暗記事項 31】 クラス図は UML の図の一つである。
- 【暗記事項 32】 クラス図はフィールド・メソッド・コンストラクタを図示するために用いる。
- 【暗記事項 33】 クラス図では，詳細なコードや局所変数を無視する。
- 【暗記事項 34】 クラス図は，クラス名を書く欄，フィールドを書く欄，メソッド・コンストラクタを書く欄の3つの欄がある。
- 【暗記事項 35】 クラス図では，フィールドと引数について，変数の型と変数名を書く順番が異なる。
- 【暗記事項 36】 クラス図では，メソッドの戻り値の型を書く位置が異なる
- 【暗記事項 37】 クラス図ではアクセス修飾子 `public` を+で表す。
- 【暗記事項 38】 クラス図ではアクセス修飾子 `private` を-で表す。
- 【暗記事項 39】 クラス図ではアクセス修飾子 `protected` を#で表す。
- 【暗記事項 40】 クラス図ではアクセス修飾子を省略した状態を~で表す。





## 2.6 節 クラス図

- ・クラス図は \_\_\_\_\_ の図の一つである [31](#)
- ・ \_\_\_\_\_ を図示するために用いる [32](#)
- ・詳細なコードや \_\_\_\_\_ を無視する [33](#)



[34](#)

### クラスのソースコードとクラス図の違い

	(ソースコード)		(クラス図)
フィールド :	型名 変数名	⇨	_____ : _____ <a href="#">35</a>
メソッド :	戻り値の型名 メソッド名 (引数リスト)	⇨	_____ : _____ <a href="#">36</a>
アクセス修飾子 :	public	⇨	_____ <a href="#">37</a>
	private	⇨	_____ <a href="#">38</a>
	protected	⇨	_____ <a href="#">39</a>
	アクセス修飾子の省略	⇨	_____ <a href="#">40</a>

## 練習問題

確認テストも、この練習問題と同じ方法で、解答してください。最終ページに答えを載せています。

解答が複数ある場合は、ハイフン で繋いで答えること。

例: A と B と C を解答したい場合、A-B-C と解答欄に記入する。答えがない場合は-1 と解答すること。

行番号を解答するとき、左詰めの 0 は取ること

例: 001 行目を解答するときは 1 を解答すること

以下のソースコードについて【練習 01】から【練習 18】を解答しなさい

```
01 public class BasicProb {
02     BasicProb() {
03         System.out.println("constructor1");
04     }
05     protected BasicProb(int a) {
06         System.out.println("constructor2:"+a);
07     }
08     private void print() {
09         System.out.println("method1");
10     }
11     public double calc() {
12         System.out.println("method2");
13         return 0.0;
14     }
15 }
```

【練習 01】 コンストラクタの個数を答えなさい

【練習 02】 メソッドの個数を答えなさい

【練習 03】 コンストラクタのすべてについて、アクセス修飾子を答えよ

【練習 04】 メソッドのすべてについて、メソッド名を答えよ

【練習 05】 メソッドのすべてについて、アクセス修飾子を答えよ

【練習 06】 メソッドのすべてについて、「戻り値の型」を答えよ

【練習 07】 ~ 【練習 18】 下のクラス図を埋めなさい

【練習 07】
【練習 08】
【練習 09】
【練習 10】
【練習 11】
【練習 12】
【練習 13】
【練習 14】
【練習 15】
【練習 16】
【練習 17】
【練習 18】

以下のソースコードについて【練習 19】から【練習 43】を解答しなさい

```
01 public class ManyConstructorsAndMethods {
02     private BasicProb basicProb;
03     protected int data;
04     public ManyConstructorsAndMethods() {
05         basicProb = new BasicProb();
06         data = 3;
07     }
08     public ManyConstructorsAndMethods(BasicProb basicProb, int data) {
09         this.basicProb = basicProb;
10         this.data = data;
11     }
12     public ManyConstructorsAndMethods(BasicProb basicProb) {
13         this.basicProb = basicProb;
14         this.data = 2;
15     }
16     public ManyConstructorsAndMethods(int data) {
17         basicProb = new BasicProb();
18         this.data = data;
19     }
20     public void printData() {
21         System.out.println(data);
22     }
23     public BasicProb getBasicProb() {
24         return basicProb;
25     }
26     public void setBasicProb(BasicProb basicProb) {
27         this.basicProb = basicProb;
28     }
29 }
```

【練習 19】 コンストラクタの個数を答えなさい

【練習 20】 メソッドの個数を答えなさい

【練習 21】 メソッドのすべてについて、メソッド名を答えよ

【練習 22】 メソッドのすべてについて、「戻り値の型」を答えよ

【練習 23】 3つ目のコンストラクタの引数のすべてについて、変数の型を答えよ

【練習 24】 2つ目のコンストラクタの引数のすべてについて、変数名を答えよ

【練習 25】 2つ目のメソッドの引数のすべてについて、変数の型を答えよ

【練習 26】 3つ目のメソッドの引数のすべてについて、変数名を答えよ

【練習 27】 すべてのフィールドについて、アクセス修飾子を答えよ

【練習 28】 すべてのフィールドについて、変数の型を答えよ

【練習 29】 すべてのフィールドについて、変数名を答えよ

【練習 30】 から【練習 33】の選択肢：

- |                 |              |
|-----------------|--------------|
| A: 基本データ型・フィールド | B: 参照型・フィールド |
| C: 基本データ型・引数    | D: 参照型・引数    |

【練習 30】 21 行目の **data** の変数の種別を答えよ

【練習 31】 24 行目の **basicProb** の変数の種別を答えよ

【練習 32】 27 行目の左辺の **basicProb** の変数の種別を答えよ

【練習 33】 27 行目の右辺の **basicProb** の変数の種別を答えよ

【練習 34】～【練習 43】 前ページのソースコードに対して、下のクラス図を埋めなさい  
該当する行がない場合は -1 と答えること

【練習 34】
【練習 35】
【練習 36】
【練習 37】
【練習 38】
【練習 39】
【練習 40】
【練習 41】
【練習 42】
【練習 43】



以下のソースコードについて【練習 44】から【練習 66】を解答しなさい

```
01 public class VariousVariables {
02     private ManyConstructorsAndMethods mcam;
03     private int data;
04     public VariousVariables() {
05         data = 2;
06         mcam = new ManyConstructorsAndMethods();
07     }
08     public VariousVariables(int data) {
09         data = data;
10         ManyConstructorsAndMethods mcam = new ManyConstructorsAndMethods();
11     }
12     public VariousVariables(ManyConstructorsAndMethods mcam, int data) {
13         this.data = data;
14         this.mcam = mcam;
15     }
16     public void method1(int data) {
17         System.out.println(data);
18     }
19     public void method2() {
20         System.out.println(data);
21     }
22     public void method3() {
23         System.out.println(this.data);
24     }
25     public void method4() {
26         int data = 0;
27         System.out.println(data);
28     }
29     public void method5(int data) {
30         System.out.println(this.data);
31     }
32     public void method6(ManyConstructorsAndMethods mcam) {
33         mcam = null;
34     }
35     public void method7() {
36         mcam = null;
37     }
38     public void method8() {
39         mcam = null;
40         ManyConstructorsAndMethods mcam = new ManyConstructorsAndMethods();
41         mcam = null;
42         this.mcam = new ManyConstructorsAndMethods();
43     }
44 }
```

【練習 44】から【練習 66】の選択肢：

- |                 |              |
|-----------------|--------------|
| A: 基本データ型・フィールド | B: 参照型・フィールド |
| C: 基本データ型・局所変数  | D: 参照型・局所変数  |
| E: 基本データ型・引数    | F: 参照型・引数    |

【練習 44】 02 行目の `mcam` の種別を答えよ

【練習 45】 03 行目の `data` の種別を答えよ

【練習 46】 05 行目の `data` の種別を答えよ

【練習 47】 06 行目の `mcam` の種別を答えよ

【練習 48】 08 行目の `data` の種別を答えよ

- 【練習 49】 09 行目の左辺の **data** の種別を答えよ
- 【練習 50】 09 行目の右辺の **data** の種別を答えよ
- 【練習 51】 10 行目の左辺の **mcam** の種別を答えよ
- 【練習 52】 12 行目の **mcam** の種別を答えよ
- 【練習 53】 12 行目の **data** の種別を答えよ
- 【練習 54】 13 行目の左辺の **data** の種別を答えよ
- 【練習 55】 13 行目の右辺の **data** の種別を答えよ
- 【練習 56】 17 行目の **data** の種別を答えよ
- 【練習 57】 20 行目の **data** の種別を答えよ
- 【練習 58】 23 行目の **data** の種別を答えよ
- 【練習 59】 27 行目の **data** の種別を答えよ
- 【練習 60】 30 行目の **data** の種別を答えよ
- 【練習 61】 33 行目の **mcam** の種別を答えよ
- 【練習 62】 36 行目の **mcam** の種別を答えよ
- 【練習 63】 39 行目の **mcam** の種別を答えよ
- 【練習 64】 40 行目の **mcam** の種別を答えよ
- 【練習 65】 41 行目の **mcam** の種別を答えよ
- 【練習 66】 42 行目の **mcam** の種別を答えよ

以下のクラス図を参照して、以下の問いに答えなさい

GenerateSRCFromCD
- basicProb: BasicProb
~ str: String
# data: int
+ GenerateSRCFromCD()
+ judge(arg: int): void

【練習 67】 から 【練習 72】 以下の空欄を埋めなさい

```
01 public 

|         |
|---------|
| 【練習 67】 |
|---------|

 {  
02 

|         |
|---------|
| 【練習 68】 |
|---------|

 ;  
03 

|         |
|---------|
| 【練習 69】 |
|---------|

 ;  
04 

|         |
|---------|
| 【練習 70】 |
|---------|

 ;  
05 

|         |
|---------|
| 【練習 71】 |
|---------|

 { }  
06 

|         |
|---------|
| 【練習 72】 |
|---------|

 { }  
07 }
```

## 練習問題の解答

【練習 01】 2

【練習 02】 2

【練習 03】 アクセス修飾子の省略-protected もしくは、package-protected (試験では後者で解答して下さい)

【練習 04】 print-calc

【練習 05】 private-public

【練習 06】 void-double

【練習 07】 BasicProb

【練習 08】 -1

【練習 09】 -1

【練習 10】 -1

【練習 11】 -1

【練習 12】 -1

【練習 13】 -1

【練習 14】 -1

【練習 15】 ~ BasicProb()

【練習 16】 # BasicProb(a: int)

【練習 17】 - print(): void

【練習 18】 + calc(): double

【練習 19】 4

【練習 20】 3

【練習 21】 printData-getBasicProb-setBasicProb

【練習 22】 void-BasicProb-void

【練習 23】 BasicProb

【練習 24】 basicProb-data

【練習 25】 -1

【練習 26】 basicProb

【練習 27】 private-protected

【練習 28】 BasicProb-int

【練習 29】 basicProb-data

【練習 30】 A

【練習 31】 B

【練習 32】 B

【練習 33】 D

【練習 34】 ManyConstructorsAndMethods

【練習 35】 - basicProb: BasicProb

【練習 36】 # data: int

【練習 37】 + ManyConstructorsAndMethods()

【練習 38】 + ManyConstructorsAndMethods(basicProb: BasicProb, data: int)

【練習 39】 + ManyConstructorsAndMethods(basicProb: BasicProb)

【練習 40】 + ManyConstructorsAndMethods(data: int)

【練習 41】 + printData(): void

【練習 42】 + getBasicProb(): BasicProb

【練習 43】 + setBasicProb(basicProb: BasicProb): void

【練習 44】 B

【練習 45】 A

【練習 46】 A

【練習 47】 B

【練習 48】 E

【練習 49】 E

【練習 50】 E

09 行目は左辺と右辺が同じ変数なので、意味のない文である。正しくは 13 行目のように書きたかったように思われる

【練習 51】 D

何週間か後に行うが、10 行目の右辺で生成されたクラス **ManyConstructorsAndMethods** のオブジェクトは、メソッド終了時 (11 行目) にオブジェクトを参照する変数がなくなるため、使われることなく消滅してしまう。

正しくは 06 行目のように書きたかったように思われる

【練習 52】 F

【練習 53】 E

【練習 54】 A

【練習 55】 E

【練習 56】 E

【練習 57】 A

【練習 58】 A

22 行目から始まるメソッド **method3()** には、**data** という変数名の局所変数や引数が存在していない。よって、この場合、20 行目のように「自身のオブジェクトを参照する特別な参照型変数 **this**」を省略して書く。

【練習 59】 C

【練習 60】 A

【練習 61】 F

【練習 62】 B

【練習 63】 B

【練習 64】 D

【練習 65】 D

【練習 66】 B

プログラムはメソッド内では上から順番に実行される。39 行目を実行している時には、**mcam** という名前の局所変数や引数は存在しない。したがって、39 行目はフィールドである。一方、40 行目で局所変数が宣言されているので、41 行目の **mcam** は局所変数となる。もし、フィールドを参照したければ、42 行目のように **this** をつける

【練習 67】 class GenerateSRCFromCD

【練習 68】 private BasicProb basicProb

【練習 69】 String str

【練習 70】 protected int data

【練習 71】 public GenerateSRCFromCD()

【練習 72】 public void judge(int arg)