

- ・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

3.2 節から 3.4 節です。

暗記事項

【暗記事項 1】 から【暗記事項 17】 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】 アクセス修飾子 `public` はどこからでもアクセス可能ということを意味している。

【暗記事項 2】 アクセス修飾子 `private` は自クラスからのみアクセス可能ということを意味している。

【暗記事項 3】 アクセス可能なフィールドは「参照型変数.フィールドの変数名」で
フィールドへの代入・フィールドの値の参照ができる

【暗記事項 4】 フィールドのアクセス修飾子を `private` にしてクラスの外部からアクセスできないようにすることをデータのカプセル化という。

【暗記事項 5】 `private` なフィールドに値を代入するメソッドのことをセッターという

【暗記事項 6】 `private` なフィールドの値を参照するメソッドのことをゲッターという

【暗記事項 7】 `private` なフィールド `width` のセッターのメソッド定義は、慣習的に、

```
public void setWidth(int width) {  
    this.width = width;  
}
```

となる。

【暗記事項 8】 `private` なフィールド `width` のゲッターのメソッド定義は、慣習的に、

```
public int getWidth() {  
    return width;  
}
```

となる。

【暗記事項 9】 オブジェクト図はオブジェクトのフィールドを図示する

【暗記事項 10】 オブジェクト図のフィールドの値はプログラムの実行箇所によって異なる。

【暗記事項 11】 UML では、オブジェクトは「参照型変数名:クラス名」という書き方をし、
さらにアンダーラインを引く

【暗記事項 12】 オブジェクト図では、フィールドの定義の隣に=3 のように値を書くことがある

【暗記事項 13】 オブジェクト図では、クラス図にあったメソッド・コンストラクタの欄がない。

【暗記事項 14】 局所変数は変数が宣言された直上の括弧開き『{』に対応する括弧閉じ『}』までが
スコープである。

【暗記事項 15】 フィールドは自身を内包しているオブジェクトが存在している限り利用可能である。

【暗記事項 16】 フィールドは自クラス内では、どこでも使用可能である。

【暗記事項 17】 フィールドは自クラス外では、アクセス修飾子によって利用可能な範囲が変わる

3.2 節 アクセス修飾子

メソッドにおけるアクセス修飾子

- **public** : でもメソッド呼出し可能 ¹
- **private** : で定義されたメソッド・コンストラクタの定義内でメソッド呼出し可能 ²

フィールドへのアクセス

アクセス可能なフィールドに参照型変数を介して

フィールドへの `参照型変数.フィールド名` ³を行うことができる

参照型変数を介したフィールドへのアクセス

`参照型変数.フィールド名` ³

(注1) `this.width` は自身を参照する特別な参照型変数 `this` を介してフィールド `width` にアクセスしている。

(注2) それ以外の用法で、参照型変数を介してフィールドにアクセスすることはあまりない。アクセスが必要な場合は後述するセッターやゲッターを定義するべきである。

データのカプセル化

フィールドのアクセス修飾子を `private` にして、クラスの外部から `フィールド名` できないようにすること ⁴

セッター・ゲッター

`private` なフィールドへのアクセスをするためのもの：

- **private** なフィールドに値を代入するメソッド ⁵
(値をセットする)
- **private** なフィールドに値を参照するメソッド ⁶
(値を取り出す)

セッターの例

`int` 型のフィールド `width` に引数の値を代入する

```
public void setWidth(int width) {  
    this.width = width;  
}
```

• アクセス修飾子は `public` • 戻り値の型は `void`

- メソッド名は `set` で始めてフィールドの変数名を続ける (先頭文字 `Width` の `W` は大文字)
- 引数の変数名はフィールドと同じにする
- 引数の型はフィールドと同じにする
- `this` が必要

ゲッターの例

フィールド `width` の値を戻り値として取り出す

```
public int getWidth() {  
    return width;  
}
```

• アクセス修飾子は `public` • 戻り値の型はフィールドと同じにする • `this` は不要

- メソッド名は `get` で始めてフィールドの変数名を続ける (先頭文字 `Width` の `W` は大文字)
- 引数リストは空

3.3 節 オブジェクト図

メモリ上に生成されたオブジェクトの個数や、オブジェクト間の関係を図示

⇨ オブジェクトの _____ を図示 [9](#)

オブジェクト図の書式

を引くこと [11](#)

参照型変数名:クラス名
フィールド

- ・フィールドの値が判っていたら、変数の型名の隣に「 _____ 」を書く [12](#)
- ・フィールドの値は、プログラムの _____ によって異なる [10](#)
- ・ _____ の欄がない [13](#)

3.4 節 スコープとオブジェクトの消滅

変数は、宣言された時に作られる。本節では、変数・オブジェクトが消滅する契機を解説する

局所変数のスコープ

局所変数は変数が宣言された直上の括弧開き『{』に
対応する括弧閉じ『}』までが _____ である [14](#)

フィールドの存在期間

- ・フィールドは _____ しているオブジェクトが存在している限り利用可能である [15](#)
- ・フィールドは _____ では、どこでも使用可能である [16](#)
- ・フィールドは _____ では、 _____ によって利用可能な範囲が変わる [17](#)

フィールドの存在期間

- ・オブジェクトを参照している参照型変数がなくなった時、
そのオブジェクトは消滅する。

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

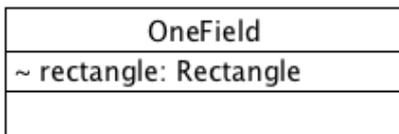
`System.out.println()` (改行あり) は `main` メソッドにしかありません。

他の場所では、`System.out.print()` (改行なし) を使っています

行番号を振っていない空行に続いて、行番号を振りなおしているソースコードが続く場合は、別の `Java` ファイル (クラス) であることを示しています。

以下のクラス図を参照して下のソースコードの【練習 01】と【練習 02】を埋めなさい

```
01 【練習 01】 {  
02     【練習 02】  
03 }
```



上のソースコードにセッターとゲッターを追加する。【練習 03】から【練習 06】を埋めなさい

```
01 【練習 01】 {  
02     【練習 02】  
03  
04     【練習 03】 { // セッターを解答してください  
05         【練習 04】  
06     }  
07  
08     【練習 05】 { // ゲッターを解答してください  
09         【練習 06】  
10     }  
11 }
```

以下のクラス図を参照して下のソースコードの【練習 07】と【練習 17】を埋めなさい。ただし、クラス図では

Graphics
- noFigures: int # triangle: Triangle

セッターとゲッターを省略している

```
01 【練習 07】 {
02     【練習 08】
03     【練習 09】
04
05     【練習 10】 { // 【練習 08】 のセッターを解答してください
06         【練習 11】
07     }
08
09     【練習 12】 { // 【練習 08】 のゲッターを解答してください
10         【練習 13】
11     }
12
13     【練習 14】 { // 【練習 09】 のセッターを解答してください
14         【練習 15】
15     }
16
17     【練習 16】 { // 【練習 09】 のゲッターを解答してください
18         【練習 17】
19     }
20 }
```

以下のクラス図を参照して下のソースコードの【練習 18】から【練習 28】を埋めなさい。ただし、クラス図ではセッターとゲッターを省略している

Circle
- coordinate: Coordinate - radius: float
+ getArea(): float

```
01 【練習 18】 {
02     【練習 19】
03     【練習 20】
04
05     【練習 21】 { // 【練習 19】 のセッターを解答してください
06         【練習 22】
07     }
08
09     【練習 23】 { // 【練習 19】 のゲッターを解答してください
10         【練習 24】
11     }
12
13     【練習 25】 { // 【練習 20】 のセッターを解答してください
14         【練習 26】
15     }
16
17     【練習 27】 { // 【練習 20】 のゲッターを解答してください
18         【練習 28】
19     }
20     public float getArea() {
21         return radius * radius * Math.PI;
22     }
23 }
```

以下のソースコードについて【練習 29】から【練習 32】を解答しなさい

```
001 public class PrivateField {
002     private int fdata;
003 }

011 public class PublicField {
012     public int fdata;
013 }

021 public class withReferencePublicField {
022     public static void main(String[] args) {
023         PublicField data = new PublicField();
024         PublicField data2 = new PublicField();
025         data2.fdata = 3;
026         data.fdata = data2.fdata;
027     }
028 }

031 public class withReferencePrivateField {
032     public static void main(String[] args) {
033         PrivateField data = new PrivateField();
034         PrivateField data2 = new PrivateField();
035         data2.fdata = 3;
036         data.fdata = data2.fdata;
037     }
038 }

041 public class NoReferencePublicField {
042     public static void main(String[] args) {
043         PublicField data;
044         PublicField data2 = new PublicField();
045         data2.fdata = 3;
046         data.fdata = data2.fdata;
047     }
048 }

051 public class NoReferencePrivateField {
052     public static void main(String[] args) {
053         PrivateField data;
054         PrivateField data2 = new PrivateField();
055         data2.fdata = 3;
056         data.fdata = data2.fdata;
057     }
058 }
```

【練習 29】から【練習 32】までの選択肢

- A: 正しい文である。
- B: data2 と data は型が違うのでコピーできない
- C: フィールド fdata のアクセス修飾子により、エラーが出る。
- D: ドット「.」はメソッド呼び出しにしか用いることが出来ないのでエラーがでる。
- E: 左辺の参照型変数はオブジェクトを参照していないので、NullPointerException が出る。

【練習 29】 26 行目について正しい説明を選びなさい

【練習 30】 36 行目について正しい説明を選びなさい

【練習 31】 46 行目について正しい説明を選びなさい

【練習 32】 56 行目について正しい説明を選びなさい

下のソースコードを読んで、設問に答えなさい：

```
001
002 public class Test {
003     private int x;
004     public int y;
005
006     public Test(){
007         x = 1;
008         y = x;
009     }
010 }

021
022 public class Main {
023     public static void main(String[] args) {
024         int y;
025         Test test;
026         y = 1;
027         test.y = 0;
028         test = new Test();
029         test.x = 1;
030
031         if(test != null){
032             int x = 0;
033             test.y = y;
034             y = 0;
035             test = null;
036         }
037         x = 1;
038         test.y = 2;
039     }
040 }
```

【練習 33】 から 【練習 37】 までの選択肢

- A: 正しい文である。
- B: 左辺の変数がスコープの外なので、コンパイルが通らない
- C: フィールドのアクセス修飾子により、エラーが出る。
- D: ドット「.」はメソッド呼び出しにしか用いることが出来ないのでエラーがでる。
- E: 左辺の参照型変数はオブジェクトを参照していないので、NullPointerException が出る。

【練習 33】 27 行目について、正しい説明を選びなさい

【練習 34】 29 行目について、正しい説明を選びなさい

【練習 35】 35 行目について、正しい説明を選びなさい

【練習 36】 37 行目について、正しい説明を選びなさい

【練習 37】 38 行目について、正しい説明を選びなさい

下のソースコードを読んで、設問に答えなさい：

```
001 public class Practice {
002     private void a() {
003         System.out.println("a");
004     }
005     public void b() {
006         System.out.println("b");
007     }
008 }

011 public class Main{
012     public static void main(String[] args) {
013         int x,y;
014         Practice some1 = null;
015         Practice some2;
016         some2 = new Practice();
017         some2.b();
018         some1.a();
019         if (some1 == null && some2 != null){
020             x=2;
021             some1.b();
022             some2.a();
023             some2=null;
024         }
025         some2.b();
026         some1.b();
027     }
028 }
029
```

【練習 38】 から 【練習 43】 までの選択肢

A: 正しい文である。

B: メソッドのアクセス修飾子により、エラーが出る。

C: 参照型変数がオブジェクトを参照していないので、NullPointerException が出る。

【練習 38】 17 行目について、正しい説明を選びなさい

【練習 39】 18 行目について、正しい説明を選びなさい

【練習 40】 21 行目について、正しい説明を選びなさい

【練習 41】 22 行目について、正しい説明を選びなさい

【練習 42】 25 行目について、正しい説明を選びなさい

【練習 43】 26 行目について、正しい説明を選びなさい

下のソースコードを読んで、設問に答えなさい：

```
001 public class PersonalInfomation {
002     private String persona;
003     public int level;
004
005     public PersonalInfomation(){
006         persona = "アルセーヌ";
007         level = 1;
008     }
009 }

021 public class Main {
022     public static void main(String args[]){
023         int battle = 0;
024         PersonalInfomation lvas;
025         lvas.level = 3;
026         lvas = new PersonalInfomation();
027
028         for(int i = 0; i < 5; i++){
029             battle += 1;
030         }
031
032         if(battle != 0){
033             battle = 0;
034             int up = 1;
035             i = 0;
036         }else{
037             lvas = null;
038         }
039
040         if(lvas != null){
041             lvas.level += 1;
042             up = 0;
043         }
044
045         lvas.persona = "リヤナンシー";
046     }
047 }
```

【練習 44】 から 【練習 49】 までの選択肢

- A: 正しい文である。
- B: 左辺の変数がスコープの外なので、コンパイルが通らない
- C: フィールドのアクセス修飾子により、エラーが出る。
- D: ドット「.」はメソッド呼び出しにしか用いることが出来ないのでエラーがでる。
- E: 左辺の参照型変数はオブジェクトを参照していないので、NullPointerException が出る。

【練習 44】 25 行目について、正しい説明を選びなさい

【練習 45】 29 行目について、正しい説明を選びなさい

【練習 46】 35 行目について、正しい説明を選びなさい

【練習 47】 41 行目について、正しい説明を選びなさい

【練習 48】 42 行目について、正しい説明を選びなさい

【練習 49】 45 行目について、正しい説明を選びなさい

下のソースコードを読んで、設問に答えなさい：

```
001 public class AccessorTest {
002     private int data;
003     public int data2;
004     public AccessorTest(int data, int data2) {
005         this.data = data;
006         this.data2 = data2;
007     }
008     private void printData() {
009         System.out.println(data);
010         System.out.println(data2);
011     }
012     public void setData(int data) {
013         this.data = data;
014     }
015     public void copyData(AccessorTest arg) {
016         this.data = arg.data;
017         this.data2 = arg.data2;
018     }
019 }

031 public class Main {
032     public static void main(String[] args) {
033         AccessorTest at1 = new AccessorTest(1, 2);
034         AccessorTest at2 = new AccessorTest(3, 4);
035         AccessorTest at3 = new AccessorTest(5, 6);
036
037         at1.printData();
038         at1.data2 = at2.data;
039         at3.copyData(at2);
040     }
041 }
```

【練習 50】 37 行目について、正しい説明を選びなさい

- A: 正しいメソッド呼び出しである。
- B: メソッド定義でのアクセス修飾子が **private** なので、このようなメソッド呼び出しはできない
- C: この参照型変数はオブジェクトを参照していないので、**NullPointerException** が出る。
- D: このメソッドは定義されていない。
- E: このメソッドと同名のメソッドは定義されているが、シグネチャが合致しないので呼び出せない。

【練習 51】 38 行目について、正しい説明を選びなさい

- A: 正しい文である。
- B: **data2** と **data** は型が違うのでコピーできない
- C: 右辺のフィールド **data** のアクセス修飾子により、エラーが出る。
- D: 左辺のフィールド **data2** の アクセス修飾子により、エラーが出る。
- E: ドット「.」はメソッド呼び出しにしか用いることが出来ないのでエラーがでる。
- F: この参照型変数はオブジェクトを参照していないので、**NullPointerException** が出る。

35 行目終了時点での参照型変数 **at3** に対するオブジェクト図を作成する。

【練習 52】 から 【練習 54】 を埋めなさい。

____ 【練習 52】 ____
【練習 53】
【練習 54】

下のソースコードを読んで、設問に答えなさい：

```
001 public class AccessorTest {
002     private int data;
003     public int data2;
004     public AccessorTest(int data, int data2) {
005         this.data = data;
006         this.data2 = data2;
007     }
008     private void printData() {
009         System.out.println(data);
010         System.out.println(data2);
011     }
012     public void show(){
013         printData();
014     }
015     public void copyData(AccessorTest arg) {
016         this.data = arg.data;
017         this.data2 = arg.data2;
018     }
019 }

031 public class Main {
032     public static void main(String[] args) {
033         AccessorTest at1 = new AccessorTest(1, 2);
034         AccessorTest at2 = new AccessorTest(3, 4);
035
036         at1.show();
037         at1.data = at2.data2;
038     }
039 }
```

【練習 55】 13 行目について、正しい説明を選びなさい

- A: 正しいメソッド呼び出しである。
- B: メソッド定義でのアクセス修飾子が **private** なので、このようなメソッド呼び出しはできない
- C: このメソッドは定義されていない。
- D: このメソッドと同名のメソッドは定義されているが、シグネチャが合致しないので呼び出せない。

【練習 56】 37 行目について、正しい説明を選びなさい

- A: 正しい文である。
- B: **data2** と **data** は型が違うのでコピーできない
- C: 左辺のフィールド **data** のアクセス修飾子により、エラーが出る。
- D: 右辺のフィールド **data2** の アクセス修飾子により、エラーが出る。
- E: ドット「.」はメソッド呼び出しにしか用いることが出来ないのでエラーがでる。
- F: この参照型変数はオブジェクトを参照していないので、**NullPointerException** が出る。

34 行目終了時点での参照型変数 **at2** に対するオブジェクト図を作成する。

【練習 57】 から 【練習 59】 を埋めなさい。

【練習 57】
【練習 58】
【練習 59】

下のソースコードを読んで、設問に答えなさい：

```
001 public class ConfirmTest {
002     private int data;
003     int data2;
004     private int data3;
005
006     public ConfirmTest(int data, int data2, int data3) {
007         this.data = data;
008         this.data2 = data2;
009         this.data3 = data3;
010     }
011
012     private void printData() {
013         System.out.println(data3);
014     }
015
016     public void setData(int data) {
017         this.data = data;
018     }
019
020     void copyData(ConfirmTest arg){
021         this.data = arg.data;
022         this.data2 = arg.data2;
023         this.data3 = arg.data3;
024     }
025 }

031
032 public class Main {
033     public static void main(String[] args) {
034         ConfirmTest ct1 = new ConfirmTest(1,2,3);
035         ConfirmTest ct2 = new ConfirmTest(4,5,6);
036         ConfirmTest ct3 = new ConfirmTest(7,8,9);
037
038         ct1.printData();
039         ct2.data2 = ct3.data3;
040         ct3.copyData(ct2);
041     }
042 }
```

【練習 60】 38 行目について、正しい説明を選びなさい

- A: 正しいメソッド呼び出しである。
- B: メソッド定義でのアクセス修飾子が **private** なので、このようなメソッド呼び出しはできない
- C: このメソッドは定義されていない。
- D: このメソッドと同名のメソッドは定義されているが、シグネチャが合致しないので呼び出せない。

【練習 61】 39 行目について、正しい説明を選びなさい

- A: 正しい文である。
- B: data2 と data3 は型が違うのでコピーできない
- C: 左辺のフィールド data2 のアクセス修飾子により、エラーが出る。
- D: 右辺のフィールド data3 の アクセス修飾子により、エラーが出る。
- E: ドット「.」はメソッド呼び出しにしか用いることが出来ないのでエラーがでる。
- F: この参照型変数はオブジェクトを参照していないので、NullPointerException が出る。

エラーが出る文をコメントしたものとして、40行目終了時点での参照型変数 `ct3` に対するオブジェクト図を作成する。

【練習 62】 から 【練習 65】 を埋めなさい。

____ 【練習 62】 ____
【練習 63】
【練習 64】
【練習 65】

練習問題の解答

- 【練習 01】 `public class OneField`
- 【練習 02】 `Rectangle rectangle;`
- 【練習 03】 `public void setRectangle(Rectangle rectangle)`
- 【練習 04】 `this.rectangle = rectangle;`
- 【練習 05】 `public Rectangle getRectangle()`
- 【練習 06】 `return rectangle;`
- 【練習 07】 `public class Graphics`
- 【練習 08】 `private int noFigures;`
- 【練習 09】 `protected Triangle triangle;`
- 【練習 10】 `public void setNoFigures(int noFigures)`
- 【練習 11】 `this.noFigures = noFigures;`
- 【練習 12】 `public int getNoFigures()`
- 【練習 13】 `return noFigures;`
- 【練習 14】 `public void setTriangle(Triangle triangle)`
- 【練習 15】 `this.triangle = triangle;`
- 【練習 16】 `public Triangle getTriangle()`
- 【練習 17】 `return triangle;`
- 【練習 18】 `public class Circle`
- 【練習 19】 `private Coordinate coordinate;`
- 【練習 20】 `private float radius;`
- 【練習 21】 `public void setCoordinate(Coordinate coordinate)`
- 【練習 22】 `this.coordinate = coordinate;`
- 【練習 23】 `public Coordinate getCoordinate()`
- 【練習 24】 `return coordinate;`
- 【練習 25】 `public void setRadius(float radius)`
- 【練習 26】 `this.radius = radius;`
- 【練習 27】 `public float getRadius()`
- 【練習 28】 `return radius;`
- 【練習 29】 A
- 【練習 30】 C
- 【練習 31】 E
- 【練習 32】 C

これだけ単純なソースだとコンパイル時にエラーが検出されるので E でもいいような感じもするが、基本的に `NullPointerException` は実行時のエラーであるのに対し、アクセス修飾子の間違いはコンパイル時に検出されるものである。従って、【練習 10】の場合、E より C が優先されるべきである（実際にコンパイルすると C のエラーが出る）

- 【練習 33】 E
- 【練習 34】 C
- 【練習 35】 A
- 【練習 36】 B
- 【練習 37】 E

【練習 38】 A

【練習 39】 B

【練習 40】 C

【練習 41】 B

【練習 42】 C

【練習 43】 C

【練習 44】 E

【練習 45】 A

【練習 46】 B

【練習 47】 A

【練習 48】 B

【練習 49】 C

【練習 50】 B

【練習 51】 C

【練習 52】 at3: AccessorTest

【練習 53】 data: int = 5

【練習 54】 data2: int = 6

オブジェクト図について、本講義でのテストでは、アクセス修飾子を省略してください。

【練習 55】 A

【練習 56】 C

【練習 57】 at2: AccessorTest

【練習 58】 data: int = 3

【練習 59】 data2: int = 4

【練習 60】 B

【練習 61】 D

【練習 62】 ct3: ConfirmTest

【練習 63】 data: int = 4

【練習 64】 data2: int = 5

【練習 65】 data3: int = 6