

- ・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

3.1 節です。

暗記事項

【暗記事項 1】 から【暗記事項 25】 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】 new 演算子によりオブジェクトを生成する書式は、「new クラス名(引数リスト)」である。

【暗記事項 2】 new 演算子によりオブジェクトを生成した直後にコンストラクタが呼び出される。

【暗記事項 3】 コンストラクタ呼出し時の引数リストには、引数の型名はいらぬ。

【暗記事項 4】 new 演算子は生成したオブジェクトの位置(アドレスのハッシュ値)を返す。

【暗記事項 5】 コンストラクタが未定義の場合、デフォルトコンストラクタ (引数リストが空で処理が書かれていないコンストラクタ) があるものとされ、このデフォルトコンストラクタが呼び出される。

【暗記事項 6】 メモリ容量の許容範囲内で一つのクラスから何個でもオブジェクトを生成することができる。

【暗記事項 7】 同一クラスから複数のオブジェクトを生成した場合、個々のオブジェクトのフィールドは別々の値を取りうる。

【暗記事項 8】 一つの参照型変数からは一つのオブジェクトを参照することができる。

【暗記事項 9】 一つのオブジェクトは複数の参照型変数から参照されることがある。

【暗記事項 10】 あるオブジェクトがどの参照型変数からも参照されなくなった場合、そのオブジェクトは消滅する。

【暗記事項 11】 メソッド呼出しの書式は「参照型変数.メソッド名(引数リスト)」である。

【暗記事項 12】 メソッド呼出し時の引数リストには、引数の型名はいらぬ。

【暗記事項 13】 呼び出すメソッドの定義を探す場合、情報実習 II の間は、

- 1) 参照型変数の型(クラス)を調べる。
- 2) (クラス名).java というファイルを調べる。
- 3) シグネチャが合致するメソッドを探し出す。

【暗記事項 14】 参照型変数の型を調べるためには、その参照型変数が宣言された箇所を探さないといけない。

【暗記事項 15】 参照型変数が局所変数の場合、メソッド呼び出しを行っているメソッド内でその変数が宣言されている。

【暗記事項 16】 参照型変数がフィールドの場合、クラス定義の中でメソッド・コンストラクタ定義の外でその変数が宣言されている。

- 【暗記事項 17】 メソッド名と引数リスト中の引数の型名をあわせてシグネチャという。
- 【暗記事項 18】 同一メソッド名のメソッドを複数つくることができる。
- 【暗記事項 19】 同一シグネチャのメソッドを複数つくることができない。
- 【暗記事項 20】 同一メソッド名でシグネチャが異なるメソッドを複数定義することをメソッドのオーバーロードという。
- 【暗記事項 21】 メソッドを呼び出すときは、参照型変数がオブジェクトを参照していなければならない。
- 【暗記事項 22】 参照型変数がオブジェクトを参照していないのにメソッドが呼び出された場合、`NullPointerException` というエラーがでる。
- 【暗記事項 23】 `rectangle.move(x,y)` のようなメソッド呼出しの時（引数が基本データ型変数）、呼出し側の変数 `x, y` が格納している「値」が引数にコピーされる。
- 【暗記事項 24】 `rectangle.copy(rectangle2)` のようにメソッド呼出しの引数が参照型変数の場合、メソッド呼出し時には、呼出し側の参照型変数 `rectangle2` の「値」、つまり、呼出し側の参照型変数が参照しているオブジェクトの位置（アドレス）が引数の参照型変数にコピーされる。つまり、呼出し側の参照型変数と引数の参照型変数は同じオブジェクトを参照することになる。
- 【暗記事項 25】 メソッド呼出しのことをメッセージパッシングという。

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。

以下のソースコードについて【練習 01】から【練習 04】を解答しなさい

```
01 public class ConstructorOnly {
02     public ConstructorOnly() {
03         System.out.print("1");
04     }
05 }

11 public class TestDrive04_1 {
12     public static void main(String[] args) {
13         System.out.println("a");
14         ConstructorOnly constructorOnly;
15         System.out.println("b");
16         constructorOnly = new ConstructorOnly();
17         System.out.println("c");
18         ConstructorOnly co = new ConstructorOnly();
19         System.out.println("d");
20     }
21 }
```

【練習 01】 出力の一行目を答えなさい

【練習 02】 出力の二行目を答えなさい

【練習 03】 出力の三行目を答えなさい

【練習 04】 出力の四行目を答えなさい

以下のソースコードについて【練習 05】から【練習 07】を解答しなさい

```
01 public class MethodOnly {
02     public void method() {
03         System.out.print("1");
04     }
05 }

11 public class TestDrive04_2 {
12     public static void main(String[] args) {
13         System.out.println("a");
14         MethodOnly methodOnly = new MethodOnly();
15         System.out.println("b");
16         methodOnly.method();
17         System.out.println("c");
18     }
19 }
```

【練習 05】 出力の一行目を答えなさい

【練習 06】 出力の二行目を答えなさい

【練習 07】 出力の三行目を答えなさい

以下のソースコードについて【練習 08】と【練習 09】を解答しなさい

```
01 public class BasicStructure {
02     private int data;
03     public BasicStructure(int data) {
04         this.data = data;
05     }
06     public void printData() {
07         System.out.print(data);
08     }
09 }

11 public class TestDrive04_3 {
12     public static void main(String[] args) {
13         BasicStructure basicStructure = new BasicStructure(2);
14         basicStructure.printData();
15         basicStructure = new BasicStructure(3);
16         System.out.println("a");
17         basicStructure.printData();
18         System.out.println("b");
19     }
20 }
```

【練習 08】 出力の一行目を答えなさい

【練習 09】 出力の二行目を答えなさい

以下のソースコードについて【練習 10】から【練習 13】を解答しなさい

```
01 public class MethodWithReturn {
02     public int method() {
03         System.out.print("1");
04         return 2;
05     }
06 }

11 public class TestDrive04_4 {
12     public static void main(String[] args) {
13         System.out.println("a");
14         MethodWithReturn methodWithReturn = new MethodWithReturn();
15         System.out.println("b");
16         int a = methodWithReturn.method();
17         System.out.println("c");
18         System.out.println(a);
19     }
20 }
```

【練習 10】 出力の一行目を答えなさい

【練習 11】 出力の二行目を答えなさい

【練習 12】 出力の三行目を答えなさい

【練習 13】 出力の四行目を答えなさい

以下のソースコードについて【練習 14】と【練習 15】を解答しなさい

```
01 public class ManyOverloadedMethods {
02     public void method() {
03         System.out.print("1");
04     }
05     public void method(int a) {
06         System.out.print("2");
07     }
08     public void method(int a, double b) {
09         System.out.print("3");
10     }
11     public void method(MethodOnly mo) {
12         System.out.print("4");
13     }
14 }

21 public class TestDrive04_5 {
22     public static void main(String[] args) {
23         ManyOverloadedMethods mom = new ManyOverloadedMethods();
24         MethodOnly mo = new MethodOnly();
25         int a = 1;
26         int b = 2;
27         double c = 3.0;
28         mom.method();
29         mom.method(mo);
30         mom.method(c);
31         mom.method(a, c);
32         mom.method(a, b);
33         System.out.println("a");
34     }
35 }
```

【練習 14】 28 行目から 32 行目のメソッド呼出しでコンパイルが通らない行があれば、すべて答えよ

【練習 15】 コンパイルが通らない行をコメントしたものととして、出力を答えよ

以下のソースコードについて【練習 16】と【練習 19】を解答しなさい

```
01 public class Pokemon {
02     private String name;
03     private int hp;
04     private String action;
05     public Pokemon(String name, int hp, String action) {
06         this.name = name;
07         this.hp = hp;
08         this.action = action;
09     }
10     public String getName() {
11         return name;
12     }
13     public int getHp() {
14         return hp;
15     }
16     public String attack() {
17         return action;
18     }
19     public void decreaseHP(int damage) {
20         hp -= damage;
21     }
22 }

31 public class TestDrive04_6 {
32     public static void main(String[] args) {
33         Pokemon picachu = new Pokemon("ピカチュウ", 570, "10まんボルト");
34         Pokemon kairiky = new Pokemon("カイリキー", 660, "こわいかお");
35
36         Pokemon player = picachu;
37         Pokemon opponent = kairiky;
38         while(true) {
39             int damage = 0;
40             if (player.attack().equals("10まんボルト")) {
41                 damage = 100;
42             }
43             if (player.attack().equals("爆裂パンチ")) {
44                 damage = 50;
45             }
46             System.out.println(player.getName()+ "の攻撃");
47             System.out.println(opponent.getName()+"は"+damage+"のダメージを受けた");
48             opponent.decreaseHP(damage);
49             if (opponent.getHp() <= 0) {
50                 System.out.println(opponent.getName()+"はポケモンセンターに行った");
51                 break;
52             }
53             Pokemon temp = player;
54             player = opponent;
55             opponent = temp;
56         }
57     }
58 }
```

【練習 16】出力の一行目を答えなさい

【練習 17】出力の二行目を答えなさい

【練習 18】出力の三行目を答えなさい

【練習 19】出力の四行目を答えなさい

以下のソースコードについて【練習 20】を解答しなさい

```
001 public class ManyOverloadedMethods {
002     public void method() {
003         System.out.print("1");
004     }
005     public void method(float a) {
006         System.out.print("2");
007     }
008     public void method(int a, double b) {
009         System.out.print("3");
010     }
011     public void method(MethodOnly mo) {
012         System.out.print("4");
013     }
014 }

021
022 public class TestDrive04_7 {
023     public static void main(String[] args) {
024         ManyOverloadedMethods mom = null;
025         MethodOnly mo = new MethodOnly();
026         int    a = 1;
027         int    b = 2;
028         double c = 3.0;
029         mom.method();
030         mom = new ManyOverloadedMethods();
031         mom.method2();
032         mom.method(mom);
033         mom.method(c);
034         mom.method(a);
035     }
036 }
```

【練習 20】 29 行目から 34 行目のメソッド呼出しでコンパイルが通らない行があれば、すべて答えよ

練習問題の解答

【練習 01】 a

【練習 02】 b

ソースコードの 14 行目は参照型変数の宣言を行っただけで、オブジェクト生成並びにコンストラクタの呼び出しは行われていない

【練習 03】 1c

【練習 04】 1d

【練習 05】 a

【練習 06】 b

【練習 07】 1c

【練習 08】 2a

【練習 09】 3b

【練習 10】 a

【練習 11】 b

【練習 12】 1c

【練習 13】 2

18 行目は以下のようにになっているので、変数 **a** の「値」が出力される。

```
18      System.out.println(a);
```

`System.out.println("a");` だったら、"a" という文字列が引数になっているので、画面に **a** と出力される

【練習 14】 30

【練習 15】 1433a

【練習 16】 ピカチュウの攻撃

【練習 17】 カイリキーは 100 のダメージを受けた

【練習 18】 カイリキーの攻撃

【練習 19】 ピカチュウは 50 のダメージを受けた

【練習 20】 29-31-32-33

29 行目は参照型変数が `null` なのでぬるぼが出る。31 行目はメソッド `method2()` が未定義なため。

32 行目と 33 行目はシグネチャが合わない。32 行目は、引数に `ManyOverloadedMethods` クラスの参照型変数を取るようなメソッドがない。33 行目は、引数 1 つのメソッド `method()` の引数の型は `float` であり、より大きい `double` 型の基本データ型変数を引数として与えることはできない。一方、34 行目は `int` 型の引数を与えているが、「拡大変換」により、自動的に `float` 型へ変換しているのでエラーは起きない。