

# できるプログラマーを本気で育てる Java超入門 Webプログラマーへの 第一歩

## 第3回 コレクションと例外処理

テクノロジックアート

長瀬 嘉秀

- コレクションとは
- 例外処理
- Java言語とオブジェクト指向
- 属性と振る舞い
- クラスとメソッド
- オブジェクト指向の特徴
- 演習問題

# 勉強会の参考書



Java  
(アジャイルソフトウェア開発技術シリーズ・基礎編)  
【発売日】2012年5月10日  
【著作】株式会社テクノロジックアート  
【監修】長瀬 嘉秀  
【編者】濱川 剛、山下 智也  
【出版】東京電機大学出版局  
【ISBN】978-4501550400

# コレクション

- 要素「オブジェクト」を格納する
- 格納されているすべての要素をコレクションから取り除く
- コレクションが空かどうかを調べる
- 特定の要素をコレクションから取り除く

# コード 1

```
ArrayList<String> animals = new  
    ArrayList<String>();  
String dog = "dog";  
String cat = "cat";  
String horse = "horse";  
  
animals.add(dog);  
animals.add(cat);  
animals.add(horse);
```

## コード2

---

```
System.out.println("3件追加後の要素数は、" + animals.size() + "です");

animals.clear();
System.out.println("clear直後のisEmptyの結果は、" + animals.isEmpty()
    + "です");

animals.add(dog);
animals.add(cat);
System.out.println("0番目の要素は、" + animals.get(0) + "です");

animals.remove(0);
System.out.println("0番目の要素は、" + animals.get(0) + "です");
```

# LISTの主なメソッド

---

- add
- size
- clear
- isEmpty
- remove



- List
  - Listは、要素を順序付けて管理する。コレクションの要素に対して、インデックスを指定してアクセスすることができる。配列に近い構造である。
- Set
  - 重複要素のないオブジェクトの集合。Listのような順序管理は行わない。複数のオブジェクトを集合として表現したい場合に適している。
- Map
  - 要素とキーを1対1で関連付けて管理する。同一のキーを複数格納することはできない。

# Setのコード 1

---

```
HashSet <String> animals = new  
    HashSet<String>();  
String dog = "dog";  
String cat = "cat";  
String horse = "horse";  
  
animals.add(dog);  
animals.add(cat);  
animals.add(horse);
```

## Setのコード 2

```
System.out.println("3件追加後の要素数は、" + animals.size() +  
    "です");
```

```
animals.clear();
```

```
System.out.println("clear直後のisEmptyの結果は、" +  
    animals.isEmpty() + "です");
```

```
animals.add(dog);
```

```
animals.add(cat);
```

```
boolean result = animals.add(cat);
```

```
System.out.println("同じ要素を2度格納しようとする、" + result  
    + "です");
```

# Setのコード3

```
for ( Iterator<String> it = animals.iterator();it.hasNext();){  
    String animal = it.next();  
    System.out.println(animal);  
}
```

```
animals.remove(cat);  
System.out.println(" catを取り除いた後のsetの中身は、" );  
for ( Iterator<String> it = animals.iterator();it.hasNext();){  
    String animal = it.next();  
    System.out.println(animal);  
}
```

# Mapのコード 1

---

```
String dog = "dog";
```

```
String cat = "cat";
```

```
String horse = "horse";
```

```
animals.put(3,horse);
```

```
animals.put(2,cat);
```

```
animals.put(1,dog);
```

# Mapのコード 2

```
System.out.println("3件追加後の要素数は、" +  
    animals.size() + "です");
```

```
for ( Iterator<Integer> it =  
    animals.keySet().iterator();it.hasNext();){  
    String animal = animals.get(it.next());  
    System.out.println(animal);  
}
```

```
animals.clear();
```

```
System.out.println("clear直後のisEmptyの結果は、" +  
    animals.isEmpty() + "です");
```

# Mapのコード 3

```
animals.put(1,dog);
animals.put(2,cat);
animals.put(3,horse);
System.out.println("key=2の要素は、" + animals.get(2) + "です");

for ( Iterator<Integer> it = animals.keySet().iterator();it.hasNext();){
    String animal = animals.get(it.next());
    System.out.println(animal);
}

animals.remove(2);
System.out.println("key=2を取り除いた後のmapの中身は、");
for ( Iterator<Integer> it = animals.keySet().iterator();it.hasNext();){
    String animal = animals.get(it.next());
    System.out.println(animal);
}
```

# 演習問題 1

次の数値をTreeMapに入れて、昇順で表示してください。

また、合計値、平均値も表示してください。

1 10 12 3 8 90 34 82



# 例外处理

# 例外処理とは

---

- 例外とは、プログラムの実行中に発生した異常事態を検知する仕組み
- 開こうとしたファイルが存在しない、など

```
public static void main(String[] args)

throws FileNotFoundException {
    BufferedReader bufferedReader = new
    BufferedReader(new
    FileReader("test.txt"));
}
```

## try-catchの基本構文

```
try {  
    例外が発生する可能性のあるコード  
} catch (例外クラス 変数) {  
    例外発生時のエラー処理  
}
```

# 例外に処理を付ける

```
throws IOException {  
    BufferedReader bufferedReader;  
    try {  
        bufferedReader = new BufferedReader(new  
            FileReader("test.txt"));  
    } catch (FileNotFoundException e) {  
        System.out.println("ファイルが見つかりませ  
            せん");  
    }  
}
```

# 入出力

# 標準入出力

---

- 標準入力 System.in
- 標準出力 System.out
- 標準エラー System.err

- 標準入力からデータを受け取る

```
throws IOException {  
    int s = System.in.read();  
    System.out.println(s);  
}
```



- 標準入力から文字列を受け取る

```
throws IOException {  
    BufferedReader bufferedReader;  
    bufferedReader =  
        new BufferedReader(new InputStreamReader(System.in));  
    String s = bufferedReader.readLine();  
    System.out.println(s);  
}
```

## 演習問題 2

---

キーボードからの入力を受け付けて、入力内容を表示し、また入力待ち状態に戻るプロンプトのプログラムを作成してください。  
また、「exit」が入力されたら待受を終了する。

- ファイルにデータを書き込む

```
throws IOException {  
    BufferedWriter bufferedWriter =  
        new BufferedWriter(new FileWriter("study43.txt"));  
    bufferedWriter.write("1111122224444");  
    bufferedWriter.close();  
}
```

## 演習問題 3

ファイル(exe42.txt)を開き、月の英語名(January、February、...)を書き込むプログラムを作成してください。

January

February

March

April

May

June

July

August

September

October

November

December

## 演習問題 4

演習問題3で作成されたファイル(exe42.txt)を読み込み、各行を行番号付きで表示するプログラムを作成してください。

1 January

2 February

⋮

# できるプログラマーを本気で育てる Java超入門 Webプログラマーへの 第一歩

## 第3回 コレクションと例外処理

テクノロジックアート

長瀬 嘉秀