Study02 レポート

細川 夏風

2025 年 5 月 26 日

1 Study02.java

1.1 ソースコード

Listing 1 Study02.java

```
1   import java.util.Scanner;
2
3   class Study02 {
4     public static void main(String[] args) {
5
6       /* このファイルで使う変数宣言 */
7       Scanner scan = new Scanner(System.in);
8       int money;
9
10      /* 飲み物のインスタンス化 */
11      Drink drink1 = new Drink("ソーダ", 130);
12      Drink drink2 = new Drink("コーラ", 130);
13      Drink drink3 = new Drink("天然水", 100);
14      Drink drink5 = new Drink("玄米茶", 120);
15
16      /* 自動販売機のクラスのインスタンス化 */
17      VendingMachine venmach = new VendingMachine();
18
19      /* 登録の処理 */
20      venmach.registDrink(1, drink1, 1);
21      venmach.registDrink(1, drink2, 4);
```

```
22          venmach.registDrink(2, drink2, 2);
23          venmach.registDrink(3, drink3, 5);
24          venmach.registDrink(5, drink5, 2);
25
26          /* ターミナルの見やすくするための改行 */
27          System.out.println();
28
29          venmach.printInfo();
30
31          /* ターミナルの見やすくするための改行 */
32          System.out.println();
33          venmach.insertMoney(0);
34
35          System.out.println();
36
37          while(true) {
38            System.out.println("行いたい操作を指定してください");
39            System.out.print("（1: お金の投入, 2: 飲み物の購入, 3: お釣りの
                返却, 4: 終了 > ");
40            switch (scan.nextInt()) {
41              case 1 -> { /* お金の投入 */
42                System.out.print("投入する金額を指定してください > ");
43                money = scan.nextInt();
44                System.out.println(money + "円を投入しました");
45                venmach.insertMoney(money);
46              }
47
48              case 2 -> { /* 飲み物の購入 */
49                System.out.print("購入する飲み物の番号を指定してください >
                    ");
50                venmach.buyDrink(scan.nextInt());
51                System.out.println();
52                venmach.printInfo();
53                venmach.insertMoney(0);
54              }
55
```

```
56              case 3 -> { /* お釣りの返却 */
57                venmach.returnMoney();
58              }
59
60              case 4 -> { /* 終了 */
61                return;
62              }
63
64              default -> {
65                System.out.println("1~4の数字の中から選択してください");
66              }
67
68            }
69
70            System.out.println();
71          }
72        }
73      }
```

1.2 工夫した点

具体的に工夫した点は 2 つある．それを以下に記す．

(1). Scanner を最初に定義しておき，次から使う際に長ったらしく Scanner について記述しなくて良くなった点だ．これにより，文字列や数字を取得するプログラムを書く際に，1 行が極端に長くなるのを避けている．

(2). インスタンス化を行ったときはそれぞれコメントで明記し，今扱っているものが変数がインスタンス化した他のクラスなのかをわかるようにした．

2 Drink.java

2.1 ソースコード

Listing 2 Drink.java

```
1   class Drink {
2
3     private String name;
4     private int price;
5
6     /* (コンストラクタ)nameで名前を指定し，priceで値段を指定する */
7     public Drink(String name, int price) {
8       this.name = name;
9       this.price = price;
10    }
11
12    /* 飲み物の名前を返す */
13    public String getName() {
14      return this.name;
15    }
16
17    /* 飲み物の値段を返す */
18    public int getPrice() {
19      return this.price;
20    }
21  }
```

2.2 工夫した点

工夫した点は 1 つある．それは以下である．

(1). コンストラクタ，getter について，その内容を深く記した．これにより，Drink を簡単に扱えるようになった．

3 VendingMachine.java

3.1 ソースコード

Listing 3 VendingMachine.java

```
1   class VendingMachine {
2     int money = 0;
3     /* 売り切れ状態と未登録状態を区別するために配列の要素全てに-1のフラグを建てる */
4     int[] stock_array = {-1, -1, -1, -1, -1};
5     Drink[] drink_array = new Drink[5];
6
7     /* 自動販売機に登録する処理 */
8     public void registDrink(int index, Drink drink, int stock) {
9       if (this.stock_array[index - 1] == -1) {
10        this.stock_array[index - 1] = stock;
11        this.drink_array[index - 1] = drink;
12        System.out.println(index + "番に「" + drink.getName() + "」を" + this.stock_array[index - 1] + "個登録しました");
13      } else {
14        System.out.println(index + "番にはすでに飲みものが登録されています");
15      }
16    }
17
18    /* 現在投入されている金額の表示 */
19    public void insertMoney(int money) {
20      System.out.println("現在" + (this.money += money) + "円入っています");
21    }
22
23    /* 仕様書閲覧(購入の処理) */
24    public void buyDrink(int index) {
25    /* 購入失敗の場合 */
26      /* 未登録の場合 */
```

```
27          if (this.stock_array[index - 1] == -1) {
28            System.out.println("購入できませんでした(" + index + "番には飲
                  み物が登録されていません)");
29          /* 売り切れの場合 */
30          } else if (this.stock_array[index - 1] == 0) {
31            System.out.println("購入できませんでした(売り切れています)");
32          /* 金額が足りない場合 */
33          } else if (this.drink_array[index - 1].getPrice() > this.money) {
34            System.out.println("購入できませんでした(お金が足りません)");
35          }
36        /* 購入可能な場合 */
37          else {
38            this.money -= this.drink_array[index - 1].getPrice();
39            this.stock_array[index - 1]--;
40            System.out.println(index + "番の「" + drink_array[index - 1].
                  getName() +"」を購入しました");
41          }
42        }
43
44        /* 釣りの返却, 投入金額を0に, お釣りの出力 */
45        public void returnMoney() {
46          System.out.println(this.money + "円のお釣りを出力しました");
47          this.money = 0;
48        }
49
50        /* 登録されている飲みのもの情報と売り切れの表示 */
51        public void printInfo() {
52          System.out.println("==================================");
53          for (int i = 0; i < 5; i++) {
54            if (stock_array[i] > 0) {
55              /* 個数が1以上のとき */
56              System.out.println((i + 1) + "番 " + this.drink_array[i].
                    getName() + " " + this.drink_array[i].getPrice() + "円 "
                    + this.stock_array[i] + "個");
57            } else if (stock_array[i] == 0) {
58              /* 個数がゼロのとき(売切の場合) */
```

```
59              System.out.println((i + 1) + "番 " + this.drink_array[i].
                    getName() + " " + this.drink_array[i].getPrice() + "円 "
                    + "売切");
60            } else {
61              /* 個数が-1のとき(未登録のとき) */
62              System.out.println("-----未登録-----");
63            }
64          }
65          System.out.println("====================================");
66        }
67      }
```

3.2 工夫した点

工夫した点は 2 つある．それを以下に記す．

(1). 初期，stock_array を定義する際，その配列全てに −1 のフラグを立てることにより，未登録と在庫切れを判別できるようにした．このプログラムならば −1 の状態は未登録であり，0 の状態は在庫切れを表せるようになっている．

(2). 購入失敗の if 分について，しっかり何が原因で購入できないのかをコメントをつけて場合分けしていることによって，視覚的に判断できるようになっている．

参考文献

[1] 中山清喬，国元大悟，「スッキリわかる Java 入門 第 4 版」，株式会社インプレス，2024 年 11 月 1 日

[2] 柴田望洋，「新・明解 Java 入門」，SB クリエイティブ株式会社，2019 年 9 月 6 日