

情報学群実験第1 Study02

2025 年 5 月 13 日

1 提出方法

作成したプログラム本体と、それぞれのプログラムの工夫した点をまとめたレポートの PDF を以下のコマンドで全てまとめて圧縮してひとつのファイルとして KUTLMS から提出すること。

```
$ tar zcf Study02_XXXXXXX_YYYY.tar.gz 圧縮したいファイルを並べる
```

このコマンドにより、Study02_XXXXXXX_YYYY.tar.gz というファイルが出力される。なお、XXXXXXX は自分の学籍番号（7桁）、YYYY は自分の氏名をローマ字表記にしたものとする。

例：学籍番号「1234567」、氏名「コウチタロウ」の場合

```
$ tar zcf Study02_1234567_Kouchi.tar.gz Study02.java Drink.java  
VendingMachine.java Study02.pdf
```

圧縮後は以下のコマンドで各プログラムとそれらの機能の説明および工夫点をまとめた PDF が含まれているか確認すること。

```
$ tar tzf Study02_XXXXXXX_YYYY.tar.gz
```

2 Study02

本課題では自動販売機を作成する。

表 1, 2 に示す仕様に従って, Drink クラスと VendingMachine クラスを作成しなさい。また, この 2 つを使用するプログラム Study02.java を作成しなさい。Study02.java は以下に示す処理を行うものとする。

1. 自動販売機に表の通りに飲み物の名前, 値段, 個数を登録する
2. 自動販売機の状態を出力する
3. 標準入力から行いたい操作の番号を入力する
 - 1 が入力された場合, 標準入力から金額を入力し, その分のお金を自動販売機に投入する
 - 2 が入力された場合, 標準入力から購入したい飲み物の番号を入力し, その飲み物を購入する。その後, 自動販売機の状態を出力する
 - 3 が入力された場合, 自動販売機からお釣りを返却する
 - 4 が入力された場合, 処理を終了する
4. 3. に戻る

備考

- Study02.java では, 表 1, 2 にないメソッドを使用してはならない
 - ただし, Study02.java 以外の別解答として別メソッドを利用した解答を提出できる。その際, レポート内で別のファイルを作成した旨を記述すること
 - なお, 適切な Study02.java が提出されていない場合, 別解答は採点されない
- 機能を実現するために必要なフィールドやコンストラクタは適宜定義して良いが, フィールドは全て `private` にすること
- 自動販売機に登録できる飲み物は 5 種類までとすること

表 1: Drink クラスに実装するメソッドの仕様

メソッド名	説明
public Drink(String name, int price)	Drink クラスのコンストラクタ. name で飲み物の名前を指定し, price で飲み物の値段を指定する
public String getName()	飲み物の名前を返す
public int getPrice()	飲み物の値段を返す

表 2: VendingMachine クラスに実装するメソッドの仕様

メソッド名	説明
public void registDrink (int index, Drink drink, int stock)	自動販売機の index 番に drink を stock 個登録する. index 番に飲み物が未登録の場合, drink を登録して登録内容を出力する. index 番に登録済みの場合, 既に飲み物が登録されていることを出力する.
public void insertMoney(int money)	お金を money 分投入し, 現在投入されている金額を出力する
public void buyDrink(int index)	自動販売機の index 番に登録されている飲み物を購入する. 以下の場合には購入に失敗する. <ul style="list-style-type: none"> • index 番に飲み物が登録されていない場合 • お金が足りない場合 • 在庫がない場合 購入に失敗した場合は, 購入に失敗したことと, 購入できない理由を出力する. 購入に成功した場合はお金をその飲み物の値段分だけ減らし, 在庫を 1 減らす. 飲み物の購入に成功したことを出力する.
public void returnMoney()	お釣りを全て返却し, 投入されている金額を 0 にする. お釣りの金額を出力する.
public void printInfo()	自動販売機の各番号に登録されている飲み物の名前, 値段を出力する. 在庫がない飲み物は右隣に「売切」の文字を出力する. 現在入っているお金を出力する.

期待される出力 1/3

```
$ javac Study02.java
```

```
$ java Study02
```

```
1 番に「ソーダ」を 1 個登録しました
1 番には既に飲み物が登録されています
2 番に「コーラ」を 2 個登録しました
3 番に「天然水」を 5 個登録しました
5 番に「玄米茶」を 2 個登録しました
```

```
=====
```

```
1 番 ソーダ 130 円 1 個
2 番 コーラ 130 円 2 個
3 番 天然水 100 円 5 個
4 番 ----- 未登録 -----
5 番 玄米茶 120 円 2 個
```

```
=====
```

現在 0 円入っています

行いたい操作を指定してください

(1: お金の投入, 2: 飲み物の購入, 3: お釣りの返却, 4: 終了) > 1

投入する金額を指定してください > 200

200 円を投入しました

現在 200 円入っています

行いたい操作を指定してください

(1: お金の投入, 2: 飲み物の購入, 3: お釣りの返却, 4: 終了) > 2

購入する飲み物の番号を指定してください > 1

1 番の「ソーダ」を購入しました

```
=====
```

```
1 番 ソーダ 130 円 売切
2 番 コーラ 130 円 2 個
3 番 天然水 100 円 5 個
4 番 ----- 未登録 -----
5 番 玄米茶 120 円 2 個
```

```
=====
```

現在 70 円入っています

行いたい操作を指定してください

(1: お金の投入, 2: 飲み物の購入, 3: お釣りの返却, 4: 終了) > 2

購入する飲み物の番号を指定してください > 2

購入できませんでした (お金が足りません)

期待される出力 2/3

=====

1 番 ソーダ 130 円 売切
2 番 コーラ 130 円 2 個
3 番 天然水 100 円 5 個
4 番 ----- 未登録 -----
5 番 玄米茶 120 円 2 個

=====

現在 70 円入っています

行いたい操作を指定してください

(1: お金の投入, 2: 飲み物の購入, 3: お釣りの返却, 4: 終了) > 1

投入する金額を指定してください > 100

100 円を投入しました

現在 170 円入っています

行いたい操作を指定してください

(1: お金の投入, 2: 飲み物の購入, 3: お釣りの返却, 4: 終了) > 2

購入する飲み物の番号を指定してください > 1

購入できませんでした (売り切れています)

=====

1 番 ソーダ 130 円 売切
2 番 コーラ 130 円 2 個
3 番 天然水 100 円 5 個
4 番 ----- 未登録 -----
5 番 玄米茶 120 円 2 個

=====

現在 170 円入っています

行いたい操作を指定してください

(1: お金の投入, 2: 飲み物の購入, 3: お釣りの返却, 4: 終了) > 2

購入する飲み物の番号を指定してください > 4

購入できませんでした (4 番には飲み物が登録されていません)

=====

1 番 ソーダ 130 円 売切
2 番 コーラ 130 円 2 個
3 番 天然水 100 円 5 個
4 番 ----- 未登録 -----
5 番 玄米茶 120 円 2 個

=====

現在 170 円入っています

期待される出力 3/3

行いたい操作を指定してください

(1: お金の投入, 2: 飲み物の購入, 3: お釣りの返却, 4: 終了) > 2

購入する飲み物の番号を指定してください > 5

5 番の「玄米茶」を購入しました

=====

1 番 ソーダ 130 円 売切

2 番 コーラ 130 円 2 個

3 番 天然水 100 円 5 個

4 番 ----- 未登録 -----

5 番 玄米茶 120 円 1 個

=====

現在 50 円入っています

行いたい操作を指定してください

(1: お金の投入, 2: 飲み物の購入, 3: お釣りの返却, 4: 終了) > 3

50 円のお釣りを出力しました

行いたい操作を指定してください

(1: お金の投入, 2: 飲み物の購入, 3: お釣りの返却, 4: 終了) > 4