

# プログラミング A 第10回・演習

演習の提出は Moodle で木曜日までに行ってください。各演習ごとに提出ファイルを zip 等で一つのファイルにまとめて該当する Moodle の課題に提出しなさい。この資料や関係するコードをインターネットなどに公開することは著作権上、禁止されています。

## 1 演習 1

メソッド main の内部においてクラス `java.util.ArrayList` を用いて、得点 80, 100, 90, 75, 90 の一覧、合計、平均を下記のように標準出力に出力するクラス `KamokuSummary` を作成し、その全体を提出せよ。

```
80
100
90
75
90
合計: 435
平均: 87
```

## 2 演習 2

メソッド main の内部においてクラス `java.util.HashMap` を用いて、氏名と得点の組の の一覧、合計、平均を下記のように標準出力に出力するクラス `KamokuSummary2` を作成し、その全体を提出せよ。具体的な氏名と得点の組としては次を用いること: 田中 80、佐藤 100、山本 90、渡辺 75、高橋 90。

`HashMap` の利用にあたり、氏名をキー、得点を値とすること。なお `HashMap` の特性上、各要素が追加した順に出現するとは限らないため、一覧における出現順の違いは問わない。

```
高橋 -> 90
山本 -> 90
佐藤 -> 100
田中 -> 80
渡辺 -> 75
合計: 435
平均: 87
```

## 3 演習 3

添付の `StringMain` において `String` 型のクラスフィールド `text` にアルファベット小文字のみからなる文字列 (例では `"helloworld"`) が与えられている。このとき、各文字を辞書順で並べ替えて標準出力に出力するクラスメソッド `sort` メソッドを書き (例では `"dehllloorw"`)、`StringMain` 全体を提出せよ。ただしコレクションフレームワークは用いないこと。

参考資料：StringMain.java

---

```
public class StringMain {
    static String text = "helloworld";

    static void sort() {

    }

    public static void main(String[] args) {
        sort();
    }
}
```

---

## 4 演習 4

StringMain を StringMain2 としてコピーしたうえで、StringMain2 インスタンスを生成したうえでインスタンスメソッドとしての sort を呼び出すように修正し、その修正後の StringMain2 全体を提出せよ。ただしコレクションフレームワークは用いないこと。

## 5 演習 5

あるクラスを拡張してサブクラスを複数定義し、インスタンスメソッドのオーバーライドにより、同インスタンスメソッドの呼び出しにおいて対象インスタンスの型に応じて振る舞いが異なることを標準出力に示さない。

## 6 演習 6

以下のすべてを満足する形で家に人とペットがいることを表すプログラムを作成し、すべてを提出せよ。

- 人とペットが名前を返す機能を持つことを、インタフェースを用意して実現すること。ここで、互いにサブクラスとせず、また共通のスーパークラスとして java.lang.Object 以外を持たないこと。
- 家に、人やペットを登録したり登録解除できること。ここで 1 人をペット 4 体分とカウントするとき、一軒の家にはペット換算で合計 20 体まで登録できるものとし、これを超える場合は RuntimeException を拡張した独自の例外クラスのインスタンスを投げること。
- 家に登録された全ての人およびペットの名前をリストアップできること。
- 新規に作成するプログラムのあらゆる箇所において、instanceof を用いないこと。
- 実行して上記機能の実現を標準出力において確認可能な HouseMain を用意すること。その形態は自由。

## 7 演習 7

以下を満足するプログラムを提出せよ。

- 何らかの抽象クラスおよびそれを拡張した具体的なクラスを定義している。
- その具体的なクラスのインスタンスを生成して何らかの形で参照したうえで、インスタンスへの参照がなくなり、強制的なガーベッジコレクションの実行によりインスタンスをガーベッジとして回収している。
- プログラムの実行により、その具体的なクラスのインスタンスの生成および回収の様子を、標準出力において確認できること。

## 8 演習 8

以下を満足するプログラムを提出せよ。

- 親は、共有口座に残高がなければランダムな額だけ金を振り込み、残高が 1 万円以上あれば（誰かによる）消費を待つ。
- 子は、共有口座に残高があれば 1 円から残高までの範囲内でランダムな額だけ消費し、残高がなければ（誰かによる）振り込みを待つ。
- 親と子はそれぞれ異なるスレッドクラスとして表現され、互いに直接には関わらない。つまり、親は子への直接の参照を持たず、子は親への直接の参照を持たない。

ヒント：Guarded Suspension パターン、Producer-Consumer パターン

## 9 演習 9

以下を満足するプログラムを提出せよ。

- 自身のメールアドレスを用いて他人とできるだけ衝突しないパッケージを作成している。
- そのパッケージ中で public なクラスを作成している。
- そのクラスに private, 無指定, protected, public の各アクセス修飾子の指定によるフィールドやメソッドを定義し、それらにアクセス可能な全ての場所からアクセスしている。そのような場所として、必要に応じて別のパッケージやクラスを作成すること。

そのうえで、上記におけるアクセスされる側とアクセスする側の関係を説明せよ（例: private なメソッドは・・・からのみアクセスできる。従って〇〇メソッドは、××クラスから～）。

## 10 演習 10

プログラム実行時の引数で指定された英文テキストのファイル中の語の出現回数を、語ごとにカウントして、語の辞書順に、語と出現回数の組を一行ごとに出力するプログラム CountMain を提出せよ。ただし、大文字・小文字は区別せず、出力は小文字で統一すること。例えば添付の sample.txt を用いて java CountMain sample.txt とすると、以下を標準出力に得ること。

```
a,1
accept,1
```

an,1  
and,1  
are,1  
areas,1  
as,2  
be,2  
bugs,1  
crucial,1  
difficult,1  
contain,1  
contained,1  
entire,1  
errors,2  
getting,1  
grip,1  
here,1  
in,1  
inevitable,1  
is,4  
isolate,1  
it,1  
much,1  
on,1  
overall,1  
possible,1  
program,1  
programs,1  
reality,1  
should,1  
so,1  
some,2  
that,2  
the,1  
there,1  
they,1  
to,2  
unacceptable,1  
what,1  
where,1  
while,1

will,2

ヒント: java.util.HashMap の代わりに java.util.TreeMap を用いると良い。TreeMap はデフォルトでは、キーの自然順序付け（ソート）を実現する。

参考資料: Sample.txt

---

Getting a grip on an entire program is, in reality, difficult, so it is inevitable that programs will contain some bugs. What is crucial here is to isolate the areas where errors are unacceptable, and to accept that while there will be some errors overall they should be contained as much as possible.

---