プログラミングA 第10回・演習　回答

1　演習 1

参考資料：KamokuSummary.java

```
import java.util.*;
public class KamokuSummary {
   public static void main(String[] args) {
      int total = 0;
      ArrayList<Integer> list = new ArrayList<Integer>();
      list.add(80);
      list.add(100);
      list.add(90);
      list.add(75);
      list.add(90);
      for (int tokuten : list) {
         total += tokuten;
         System.out.println(tokuten);
      }
      System.out.println("合計: " + total);
      System.out.println("平均: " + total / list.size());
   }
}
```

2　演習 2

参考資料：KamokuSummary2.java

```
import java.util.*;
public class KamokuSummary2 {
   public static void main(String[] args) {
      int total = 0;
      int tokuten = 0;
      HashMap<String, Integer> map =
         new HashMap<String, Integer>();
      map.put("田中", 80);
      map.put("佐藤", 100);
      map.put("山本", 90);
      map.put("渡辺", 75);
      map.put("高橋", 90);
      for (String name : map.keySet()) {
         tokuten = map.get(name);
         total += tokuten;
         System.out.println(name + " -> " + tokuten);
```

```
    }
    System.out.println("合計: " + total);
    System.out.println("平均: " + total / map.size());
  }
}
```

3 演習 3

参考資料：StringMain.java

```
public class StringMain {
  static String text = "helloworld";
  static void sort() {
    char[] chars = text.toCharArray();
    for(int i = 0; i < chars.length - 1; i++) {
      for(int j = i + 1; j < chars.length; j++) {
        if(chars[i] > chars[j]) {
          char temp = chars[i];
          chars[i] = chars[j];
          chars[j] = temp;
        }
      }
    }
    System.out.println(chars);
  }
  public static void main(String[] args) {
    sort();
  }
}
```

4 演習 4

参考資料：StringMain2.java

```
public class StringMain2 {
  static String text = "helloworld";
  void sort() {
    char[] chars = text.toCharArray();
    for(int i = 0; i < chars.length - 1; i++) {
      for(int j = i + 1; j < chars.length; j++) {
        if(chars[i] > chars[j]) {
          char temp = chars[i];
          chars[i] = chars[j];
          chars[j] = temp;
        }
      }
```

```
    }
    System.out.println(chars);
  }
  public static void main(String[] args) {
    new StringMain2().sort();
  }
}
```

5 演習 5

省略。

6 演習 6

参考資料：Namable.java

```
public interface Namable {
  String name();
  int size();
}
```

参考資料：Person.java

```
public class Person
implements Namable {
  private String name = null;
  public Person(String s) {
    name = s; }
  public String name() {
    return name; }
  public int size() {
    return 4; }
}
```

参考資料：Pet.java

```
public class Pet
implements Namable {
  private String name = null;
  public Pet(String s) {
    name = s; }
  public String name() {
    return name; }
  public int size() {
    return 1; }
}
```

参考資料：HouseMain.java

```
public class HouseMain {
   public static void main(String[] args) {
      House h = new House();
      Namable person1 = new Person("Taro");
      Namable person2 = new Person("Hanako");
      Namable pet = new Pet("Pochi");
      h.add(person1);
      h.add(person2);
      h.add(pet);
      System.out.println(h.listUp());
      h.remove(person1);
      System.out.println(h.listUp());
      for(int i = 0; ; i++) {
         h.add(new Person("Person" + i));
         h.add(new Pet("Pet" + i));
         System.out.println(h.listUp());
      }
   }
}
```

参考資料：House.java

```
import java.util.*;
public class House {
   private List<Namable> list = new ArrayList<Namable>();
   private static final int MAX_SIZE = 20;
   private int totalSize = 0;
   public void add(Namable n) {
      if(totalSize + n.size() > MAX_SIZE) throw new OverSizeException();
      list.add(n);
      totalSize += n.size();
   }
   public void remove(Namable n) {
      list.remove(n);
      totalSize -= n.size();
   }
   public String listUp() {
      String result = "";
      for(Namable n: list) result += n.name() + ",";
      return result;
   }
}
```

参考資料：OverSizeException.java

```
public class OverSizeException extends RuntimeException {
}
```

7　演習 7

参考資料：GarbageMain.java

```
public class GarbageMain {
   public static void main(String[] args) {
      AbstractClass c = new ConcreteClass();
      c = null;
      System.gc();
   }
}
```

参考資料：AbstractClass.java

```
abstract class AbstractClass {
}
```

参考資料：ConcreteClass.java

```
class ConcreteClass extends AbstractClass {
   ConcreteClass() {
      System.out.println("Created.");
   }
   protected void finalize() {
      System.out.println("To be garbage-collected.");
   }
}
```

8　演習 8

参考資料：Account.java

```
class Account {
   int balance = 0;
   public synchronized void deposit(int n) throws InterruptedException {
      while (balance >= 10000) {
         wait();
      }
      balance += n;
      notifyAll();
      System.out.println("deposit: " + n + "," + balance);
   }
   public synchronized int withdraw() throws InterruptedException {
      while(balance == 0) {
         wait();
      }
```

```
      int n = (int) (Math.random() * balance) + 1;
      notifyAll();
      balance -= n;
      System.out.println("withdraw: " + n + "," + balance);
      return n;
   }
}
```

参考資料：Parent.java

```
class Parent extends Thread {
   Account account = null;
   Parent(Account a) {
      account = a;
   }
   public void run() {
      try {
         while(true) {
            account.deposit(
                  (int)(Math.random()*50000)+1);
            sleepRandomly();
         }
      } catch (InterruptedException e) { }
   }
   void sleepRandomly() {
      try {
         int n = (int)(Math.random()*1000);
         Thread.sleep(n);
      } catch (InterruptedException e) {}
   }
}
```

参考資料：Child.java

```
class Child extends Thread {
   Account account = null;
   Child(Account a) {
      account = a;
   }
   public void run() {
      try {
         while (true) {
            int x = account.withdraw();
            sleepRandomly();
         }
      } catch (InterruptedException e) { }
   }
   void sleepRandomly() {
      try {
```

```
        int n = (int)(Math.random() * 1000);
        Thread.sleep(n);
      } catch (InterruptedException e) { }
   }
}
```

参考資料：AccountMain.java

```
public class AccountMain {
   public static void main(String[] args) {
      Account account = new Account();
      Parent parent = new Parent(account); Child child = new Child(account);
      parent.start(); child.start();
   }
}
```

9 演習 9

参考資料：jp/waseda/ess/access/Asset.java

```
package jp.waseda.ess.access;
public class Asset {
   private int privateField = 0;
   String unspecifiedField = "xxx";
   protected void protectedMethod() { }
   public void publicMethod() { }
   public Asset() {
      privateField = 1;
      unspecifiedField = "yyy";
      protectedMethod();
      publicMethod();
   }
}
```

参考資料：jp/waseda/ess/another/SubAsset.java

```
package jp.waseda.ess.another;
import jp.waseda.ess.access.Asset;
public class SubAsset extends Asset {
   public SubAsset() {
      //int n = privateField;
      //String temp = unspecifiedField;
      protectedMethod();
      publicMethod();
      Asset asset = new Asset();
      //int n = asset.privateField;
      //String temp = asset.unspecifiedField; //asset.protectedMethod();
```

```
    asset.publicMethod();
  }
}
```

参考資料：jp/waseda/ess/access/Client.java

```
package jp.waseda.ess.access;
public class Client {
  Client() {
    Asset asset = new Asset();
    //int n = asset.privateField;
    String temp = asset.unspecifiedField;
    asset.protectedMethod();
    asset.publicMethod();
  }
}
```

参考資料：jp/waseda/ess/anothe/Client.java

```
package jp.waseda.ess.another;
import jp.waseda.ess.access.Asset;
public class Client {
  Client() {
    Asset asset = new Asset();
    //int n = asset.privateField;
    //String temp = asset.unspecifiedField;
    //asset.protectedMethod();
    asset.publicMethod();
  }
}
```

10 演習 10

参考資料：CountMain.java

```
import java.util.*;
import java.io.*;
public class CountMain {
  public static void main(String[] args) throws FileNotFoundException {
    Scanner scanner = new Scanner(new FileReader(args[0])).useDelimiter("[¥¥.,;]*¥¥s+");
    Map<String, Integer> map = new TreeMap<String, Integer>();
    while(scanner.hasNext()) {
      String term = scanner.next().toLowerCase();
      if(map.containsKey(term)) map.put(term, map.get(term) + 1);
      else map.put(term, 1);
    }
    for(String s: map.keySet())
```

```
        System.out.println(s + "," + map.get(s));
    }
}
```