Problem 1

Write a class **Casino** objects of which are *iterable*, i.e., the class implements interface **Iterable**. Objects of this type (as, for example, objects representing collections) may be used in "for-each" loops. Therefore, object of our class **Casino** should also make it possible. The implementation should be such that iteration over an object of class **Casino** returns random numbers from the set $\{0, 1\}$ until three last returned numbers are identical (three zeros or three ones in a row). Hence, a program like:

download *Casino.java*

```
public class Main {
    public static void main(String[] args) {
        for (int turn = 0; turn < 10; ++turn) {
            for (Integer i : new Casino())
                System.out.print(i + " ");
            System.out.println();
        }
    }
}
```

should print something like:

```
0 1 0 1 0 0 0
0 1 1 0 0 0
0 0 1 0 1 0 1 0 1 1 1
0 1 1 1
1 0 1 0 1 0 0 1 0 0 0
1 1 1
1 0 1 1 0 1 1 1
1 1 0 1 1 0 1 1 1
0 0 1 1 1
0 0 1 0 0 1 0 1 0 0 1 0 1 1 1
```

[ATTENTION: there must be no loops, strings, arrays or other collections in your classes!]

Problem 2

Write a program which reads, from a text file, data describing purchases made by a few customers. Each line of the file contains, space separated, name of a customer, name of a purchased product, and its price; for example

```
Mary shoes 1000
John beer 5
Mary stockings 70
John beer 6
```

```
Mary lipstick 40
John hammer 25
John beer 4
Mary shoes 1200
James butter 9
John beer 7
Mary shoes 2250
John beer 7
```

The program puts the data read into a map of type

```
Map<String,List<Purchase>>
```

where the name of a customer is the keys, and a list of his/her purchases plays the rôle of the associated value; each purchase is represented by an object of type **Purchase** with (private) fields name (name of a product) and price (its price).

After all data has been read and put into the map, the program writes another text file with a summary of purchases: each line of the file contains, space separated,

1. name of a customer;
2. number of all his/her purchases;
3. number of *different* (with different names) products bought by this customer;
4. total expenses for all his/her purchases.

For the data as in the example above, the result file should thus contain:

```
James 1 1 9
John 6 2 54
Mary 5 3 4560
```

Problem 3

Write a program creating a map of type

```
Map<String,List<Car>>
```

from the data contained in an array of **String**s. Each three consecutive elements of the array specify the name of the salon and the make and price of a car (the price is given as a **String**). Names of salons are keys of the map and lists of cars (of type **Car**) offered in a given salon are values.

The array arr should not be used after the data has been placed in the map. The program prints

- in any form, the contents of the map;
- the make, price and salon of the least expensive car.

For the following array (defined at the beginning of the **main** function):

```
String[] arr = {
    "salon A", "Mercedes", "130000",
    "salon B", "Mercedes", "120000",
    "salon C", "Ford", "110000",
    "salon B", "Opel", "90000",
    "salon C", "Honda", "95000",
    "salon A", "Ford", "105000",
    "salon A", "Renault", "75000"
};
```

the result should be something like

```
{salon A=[Mercedes 130000, Ford 105000, Renault 75000],
 salon B=[Mercedes 120000, Opel 90000],
 salon C=[Ford 110000, Honda 95000]}

Renault in salon A for 75000
```
