

Class TicketMachine – naive-ticket-machine

1/3

```
1  /**
2   * TicketMachine models a naive ticket machine that issues
3   * flat-fare tickets.
4   * The price of a ticket is specified via the constructor.
5   * It is a naive machine in the sense that it trusts its users
6   * to insert enough money before trying to print a ticket.
7   * It also assumes that users enter sensible amounts.
8   *
9   * @author David J. Barnes and Michael K€lling
10  * @version 2011.07.31
11  *
12  * Modified by Artem Los
13  * artem@artemlos.net
14  */
15  public class TicketMachine
16  {
17      // The price of a ticket from this machine.ticketMal
18      private int price;
19      // The amount of money entered by a customer so far.
20      private int balance;
21      // The total amount of money collected by this machine.
22      private int total;
23
24
25      /**
26       * creates a machine with a default price of 4711.
27       */
28      public TicketMachine()
29      {
30          price = 4711;
31      }
32
33
34      /**
35       * Create a machine that issues tickets of the given price.
36       * Note that the price must be greater than zero, and there
37       * are no checks to ensure this.
38       */
39      public TicketMachine(int cost)
40      {
41          price = cost;
42          balance = 0;
43          total = 0;
44      }
45
46      /**
47       * clearing total
48       * this method is an mutator since it
49       * alters a variable.
```

```
50     */
51     public void empty()
52     {
53         total = 0;
54     }
55
56
57     /**
58      * Sets a new price for the tickets.
59      */
60     public void setPrice(int newPrice)
61     {
62         price = newPrice;
63     }
64
65     /**
66      * Return the price of a ticket.
67      */
68     public int getPrice()
69     {
70         return price;
71     }
72
73     /**
74      * Return the amount of money already inserted for the
75      * next ticket.
76      */
77     public int getBalance()
78     {
79         return balance;
80     }
81
82     /**
83      * Receive an amount of money from a customer.
84      */
85     public void insertMoney(int amount)
86     {
87         balance = balance + amount;
88     }
89
90     /**
91      * Print a ticket.
92      * Update the total collected and
93      * reduce the balance to zero.
94      */
95     public void printTicket()
96     {
97         // Simulate the printing of a ticket.
98         System.out.println("#####");
```

```
99         System.out.println("# The BlueJ Line");
100        System.out.println("# Ticket");
101        System.out.println("# " + price + " cents.");
102        System.out.println("#####");
103        System.out.println();
104
105        // Update the total collected with the balance.
106        total = total + balance;
107        // Clear the balance.
108        balance = 0;
109    }
110
111    public int getTotal()
112    {
113        return total;
114    }
115 }
116
```