# DD1339 Introduktion till datalogi 2013/2014

# Uppgift nummer: Hemuppgift 2

# Namn: Marcus Larsson

# Grupp nummer: 5

# Övningsledare: Marcus Dicander

# Betyg: ..... Datum: .............. Rättad av: .......................................

# Exercise 2.44 och 2.45

Implementerade konstuktorerna enligt övning 2.44 i källkoden som bifogas nedan.

Frågorna från övning 2.45 besvaras som följande:

Metoden behöver inte ta några parametrar och det är en "mutator" eftersom den ändrar ett värde.

## Källkod från klassen TicketMachine:

/\*\*

\* TicketMachine models a naive ticket machine that issues

\* flat-fare tickets.

\* The price of a ticket is specified via the constructor.

\* It is a naive machine in the sense that it trusts its users

\* to insert enough money before trying to print a ticket.

\* It also assumes that users enter sensible amounts.

\*

\* @author David J. Barnes and Michael Kölling (Latest modified by Marcus Larsson)

\* @version 2013.09.13

\*/

public class TicketMachine

{

// The price of a ticket from this machine.

private int price;

// The amount of money entered by a customer so far.

private int balance;

// The total amount of money collected by this machine.

private int total;

/\*\*

\* Create a machine that issues tickets of the given price.

\* Note that the price must be greater than zero, and there

\* are no checks to ensure this.

\* @param cost Enter a price for the tickets

\*/

public TicketMachine(int cost)

{

price = cost;

balance = 0;

total = 0;

}

/\*\*

\* Create a machine that issues tickets of a default price of 1000 cents

\*/

public TicketMachine()

{

price = 1000;

balance = 0;

total = 0;

}

/\*\*

\* This method will empty the machine of all the money it has collected.

\*/

public void empty(){

total = 0;

}

/\*\*

\* Return the price of a ticket.

\*/

public int getPrice()

{

return price;

}

/\*\*

\* This method will change the prive of the tickets.

\* Not possible to enter negative value. Then price will automatically be set to 0.

\* @param price Enter new price of the tickets.

\*/

public void setPrice(int price){

if(price>0){

this.price = price;

} else {

this.price=0;

}

}

/\*\*

\* Return the amount of money already inserted for the

\* next ticket.

\*/

public int getBalance()

{

return balance;

}

/\*\*

\* Receive an amount of money from a customer.

\*/

public void insertMoney(int amount)

{

balance = balance + amount;

}

/\*\*

\* Print a ticket.

\* Update the total collected and

\* reduce the balance to zero.

\*/

public void printTicket()

{

// Simulate the printing of a ticket.

System.out.println("##################");

System.out.println("# The BlueJ Line");

System.out.println("# Ticket");

System.out.println("# " + price + " cents.");

System.out.println("##################");

System.out.println();

// Update the total collected with the balance.

total = total + balance;

// Clear the balance.

balance = 0;

}

}