

CS112 - INTRODUCTION TO PROGRAMMING

Examples of various features that can occur in class definitions

The following listing is from the SavingsAccount class used in class demonstrations.

```
/**
 * Representation for an interest-bearing savings account
 *
 * @author Russell C. Bjork
 * @version February 14, 2008
 */
public class SavingsAccount
{
    private Customer owner;
    private int currentBalance;    // Represented in cents
    private final int accountNumber;

    private static int nextAccountNumber = 1;
    private static double annualInterestRate;

    /**
     * Constructor for objects of class SavingsAccount
     *
     * @param owner the owner of this account
     *
     * The balance will be set to zero
     * The account number will be set to the first available unused number
     */
    public SavingsAccount(Customer owner)
    {
        this.owner = owner;
        owner.addAccount(this);
        currentBalance = 0;
        accountNumber = nextAccountNumber ++;
    }

    /**
     * Deposit money
     *
     * @param amount the amount to deposit (in cents)
     */
    public void deposit(int amount)
    {
        currentBalance += amount;
    }

    /**
     * Withdraw money
     *
     * @param amount the amount to withdraw (in cents)
     * @exception IllegalArgumentException if insufficient balance on hand
     */
    public void withdraw(int amount)
    {
        if (currentBalance < amount)
            throw new IllegalArgumentException();
        currentBalance -= amount;
    }
}
```

← File/Class Comment

← Instance Variables

← Class Variables

← Constructor

← Instance Method (Mutator)

← Instance Method (Mutator)

```

/**
 * Modify the interest rate
 *
 * @param newRate the new annual interest rate
 */
public static void setAnnualInterestRate(double newRate)
{
    annualInterestRate = newRate;
}

/**
 * Calculate interest for one month.
 */
public void calculateInterest()
{
    if (currentBalance >= MINIMUM_AMOUNT_FOR_INTEREST)
        currentBalance += (int)
            (currentBalance * annualInterestRate / 12.0);
}

/**
 * Report current balance.
 * @return current balance, formatted neatly as dollars and
 *         cents, with a dollar sign and decimal point
 */
public String reportBalance()
{
    String result = "$" + currentBalance/100 + ".";
    int cents = currentBalance % 100;
    if (cents < 10)
        result += "0";
    result += cents;
    return result;
}

/** Accessor for account number
 *
 * @return account number for this account
 */
int getAccountNumber()
{
    return accountNumber;
}

/** The minimum balance an account can have and still receive interest
 */
public static final int
    MINIMUM_AMOUNT_FOR_INTEREST = 500;
}

```

← Class Method (Mutator)

← Instance Method (Mutator)

← Instance Method (Accessor)

← Instance Method (Accessor)

← Class Constant