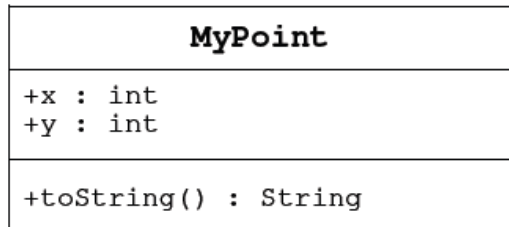


1. uzdevums

UML diagrammā ir redzama MyPoint klase. Ievērojiet, ka instances mainīgie x un y abi ir publiski. Tas nozīmē, ka tiem var piekļūt tiešā veidā. Šajā klasē tiek arī izmantota metode toString, lai varētu klases instances vērtības izdrukāt standarta izvadā (System.out)



MyPoint.java kods:

```
public class MyPoint {  
  
    public int x;  
  
    public int y;  
  
    public String toString() {  
        return "[" + x + "," + y + "];"  
    }  
}
```

Darbības:

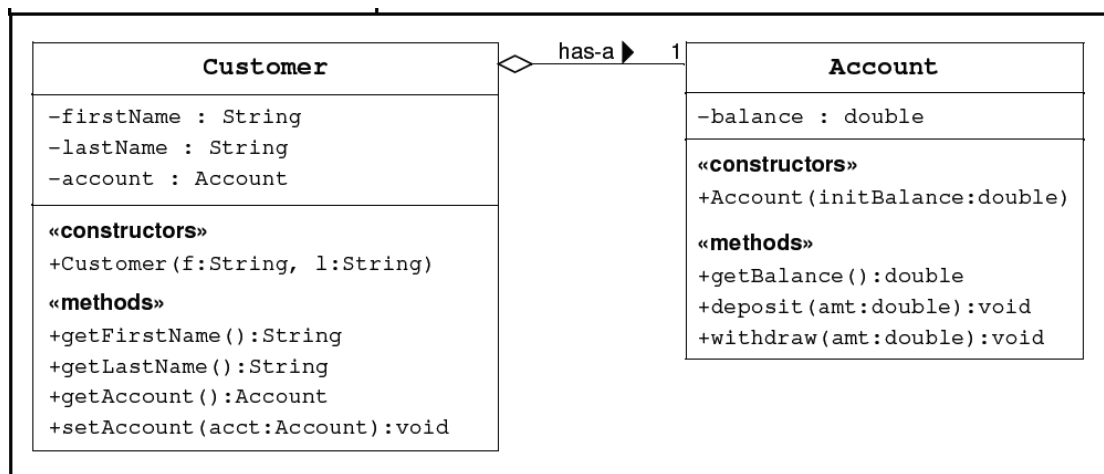
1. Izveidojiet jaunu projektu.
2. Projektā izveidojiet klasi MyPoint
3. Izveidojiet projektā TestMyPoint klasi ar main metodi. Metodē ir jāveic sekojošas darbības:
 - a. deklarēt start un end mainīgos ar tipu MyPoint.
 - b. Abiem mainīgajiem piesaistiet jaunus MyPoint objektus
 - c. uzstādīt start objektam x un y vērtības uz 10.
 - d. Uzstādīt end objektam x vērtību 20, y vērtību 30.
 - e. Izdrukāt abu (start, end) mainīgos vērtības standarta izvadā (System.out)

4. Nokompilējiet un pārbaudiet TestMyPoint darbību.
5. Papildiniet TestMyPoint main metodi:
 - a. deklarējiet jaunu mainīgo ar nosaukumu stray un tipu MyPoint
 - b. piešķiriet stray mainīgajam end mainīgā vērtību
 - c. Izdrukājiet stray un end objektus standarta izvadā.
 - d. Piešķiriet stray mainīgā atribūtam x vērtību 47, y atribūtam vērtību 40
 - e. izdrukājiet visu trīs mainīgo (start, end, stray) vērtības standarta izvadā.

2. uzdevums

Šajā uzdevumā būs jāizveido Bankas projekts, pievienojot Customer un Account klasi.

UML diagrammā ir redzama Customer klase un tās attiecības ar Account klasi. Šo attiecību lasa kā „Klientam ir viens konts”.



Darbības:

1. Izveidot Customer klasi saskaņā ar doto specifikāciju
2. Izveidot Account klasi saskaņā ar doto specifikāciju
3. Pārbaudīt Customer klasi izmantojot zemāk doto programmu TestBanking

TestBanking.java kods:

```
package test;
```

```

import domain.*;

public class TestBanking {

    public static void main(String[] args) {

        Customer customer;

        Account account;

        // Create an account that can has a 500.00 balance and type savings.
        System.out.println("Creating the customer Jane Smith.");
        customer = new Customer("Jane", "Smith");
        System.out.println("Creating her account with a 500.00 balance.");
        customer.setAccount(new Account(500.00));

        account = customer.getAccount();
        System.out.println("Withdraw 150.00");
        account.withdraw(150.00);

        System.out.println("Deposit 22.50");
        account.deposit(22.50);

        System.out.println("Withdraw 47.62");
        account.withdraw(47.62);

        // Print out the final account balance
        System.out.println("Customer [" + customer.getLastName()
                           + ", " + customer.getFirstName()
                           + "] has a balance of " + account.getBalance());
    }
}

```

3. uzdevums

Izveidojiet programmu, kura apstaigā veselus skaitļus no 1 līdz 50 (ieskaitot), izdrukā vērtību atsevišķā rindā. Rindā arī jāizdrukā vārds „foo”,

ja skaitlis dalās ar 3. Rindā jāizdrukā vārds „bar”, ja skaitlis dalās ar 5. Ja skaitlis dalās ar 7, tad jāizdrukā vārds „baz”.

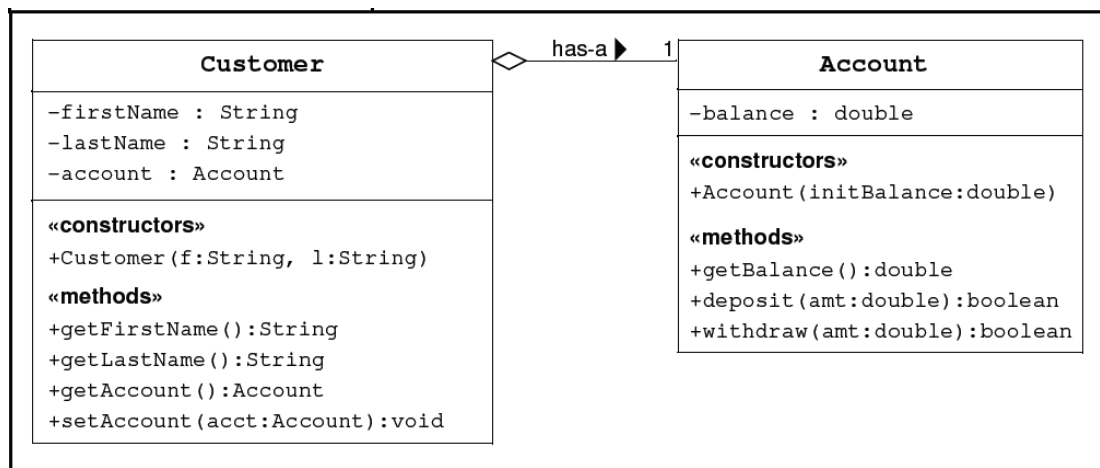
Lūk piemērs no programmas izvada:

```
1
2
3 foo
4
5 bar
6 foo
7 baz
8
9 foo
10 bar
11
12 foo
13
14 baz
15 foo bar
16
```

4. uzdevums

Šajā uzdevumā nepieciešams modificēt Account klases withdraw metodi tā, lai tā atgrieztu boolean vērtību. Vērtība nosaka, vai darbība bija veiksmīga vai nē.

UML diagramma



Darbības:

1. Izveidot kodu atbilstoši UML diagrammai
2. Pārbaudīt Account klases darbību izmantojot TestBanking programmu

TestBanking.java kods:

```
package test;
```

```
import domain.*;

/*
 * This class creates the program to test the banking classes.
 * It creates a new Customer and Account (with an initial balance),
 * and performs a series of transactions with the Account object.
 */

public class TestBanking {

    public static void main(String[] args) {

        Customer customer;

        Account account;

        boolean operationResult;

        // Create an account that can has a 500.00 balance and type savings.

        System.out.println("Creating the customer Jane Smith.");

        customer = new Customer("Jane", "Smith");

        System.out.println("Creating her account with a 500.00 balance.");

        customer.setAccount(new Account(500.00));

        account = customer.getAccount();

        operationResult = account.withdraw(150.00);

        System.out.println("Withdraw 150.00: " + operationResult);

        operationResult = account.deposit(22.50);

        System.out.println("Deposit 22.50: " + operationResult);

        operationResult = account.withdraw(47.62);

        System.out.println("Withdraw 47.62: " + operationResult);

        operationResult = account.withdraw(400.00);
```

```

        System.out.println("Withdraw 400.00: " + operationResult);

        // Print out the final account balance

        System.out.println("Customer [" + customer.getLastName()

            + ", " + customer.getFirstName()

            + "] has a balance of " + account.getBalance());

    }

}

```

5. uzdevums

TestIsSubString klasē implementēt metodi isSubString.

Ir divi veidi kā to izdarīt:

1. Vienkāršākais izmantojot String klases metodi indexOf(String). Metode atgriežīs -1, ja arguments nebūs sastopams dotajā String objektā.
2. Vēlamais (izmantojot saistītos ciklus) – pielietot String metodes charAt metodi. Stringu var pārveidot pa simbolu masīvu izmantojot String klases metodi toCharArray(). Masīvu indeksācija sākas ar 0.

TestIsSubString.java programma:

```

public class TestIsSubString {

    public static boolean isSubString(String x, String text){

        return false;

    }

    public static void main(String[] args) {

        String text = "The cat in the hat.";

        System.out.println("isSubString(\"cat\", \"The cat in the hat.\") "

            + isSubString("cat", text)); //true

        System.out.println("isSubString(\"bat\", \"The cat in the hat.\") "

            + isSubString("bat", text)); //false
    }
}

```

```
System.out.println("isSubString(\"The\", \"The cat in the hat.\") "  
    + isSubString("The", text)); //true
```

```
System.out.println("isSubString(\"hat.\", \"The cat in the hat.\") "  
    + isSubString("hat.", text)); //true
```