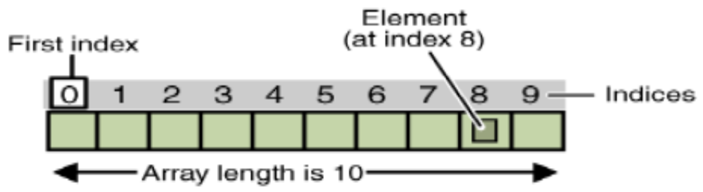
PBO 4 : Array & Arraylist

Royana Afwani

2020

Konsep Array (Larik)

- **Array** dalam java diperlakukan sebagai **objek**.
- Array adalah objek yang dapat digunakan untuk menyimpan sejumlah data dalam tipe sama dengan jumlah elemen tetap



- Elemen yang disimpan pada array dapat berupa tipe primitif (int, float, etc) atau objek (instan dari class)
- Langkah menciptakan array:
  1. Mendeklarasikan variabel array
  2. Menciptakan objek array

Deklarasi Variabel Array

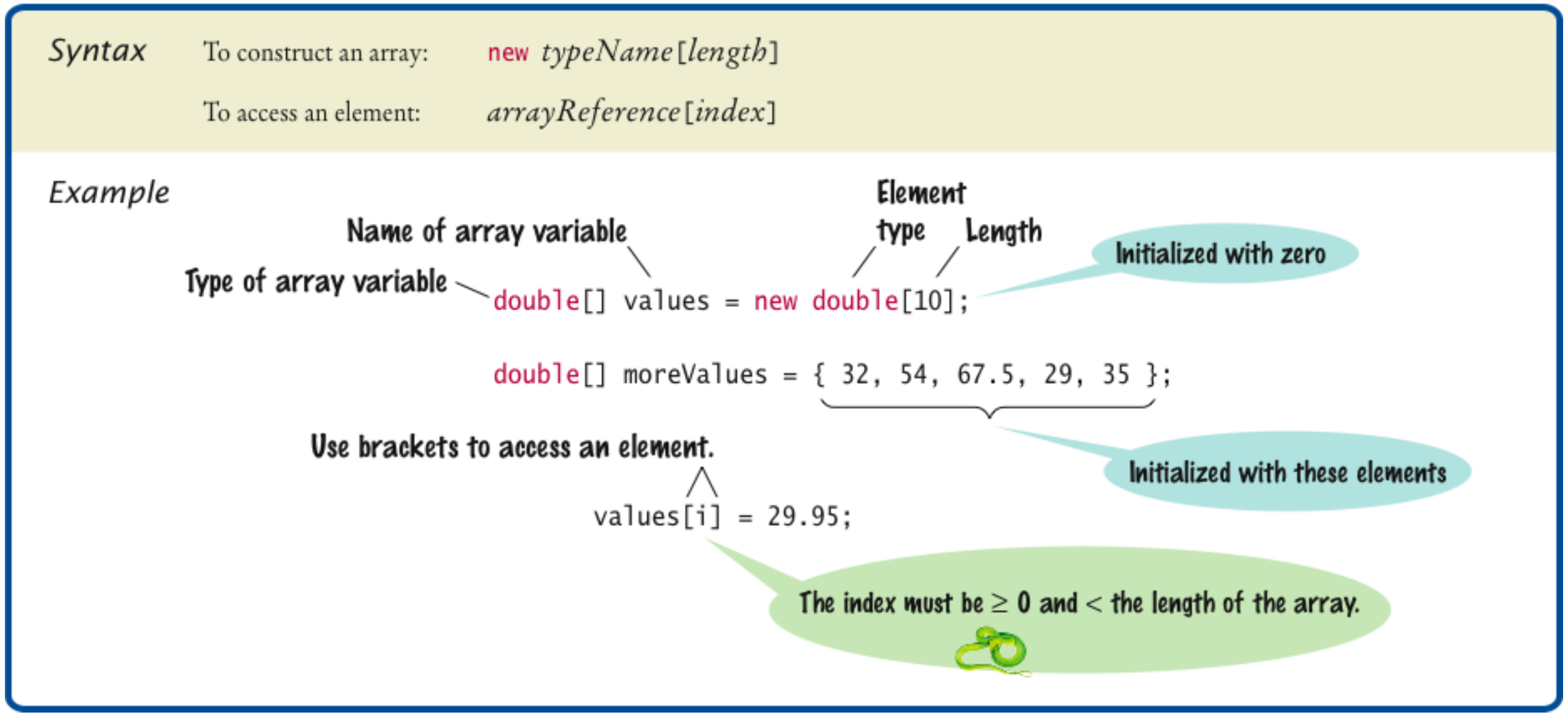
- Bentuk Deklarasi:
  ```
  tipePrimitif[] namaVariabel;
  namaKelas[] namaVariabel;
  ```

- Contoh:
  ```
  String[] kota;
  int[] nomor;
  ```

Menciptakan Objek Array

- Bentuk Deklarasi:
```
namaVariabel = new tipePrimitif[jumlahElemen];
namaVariabel = new namaKelas[jumlahElemen];
```
- Contoh:
```
nomor = new int[7];
kota = new String[8];
```
- Bentuk singkat deklarasi variable dan objek array:
```
String[] kota = new String[8];
int[] nomor = new int[7];
```

Deklarasi dan Pemberian Nilai Array



Syntax To construct an array: `new typeName[length]`

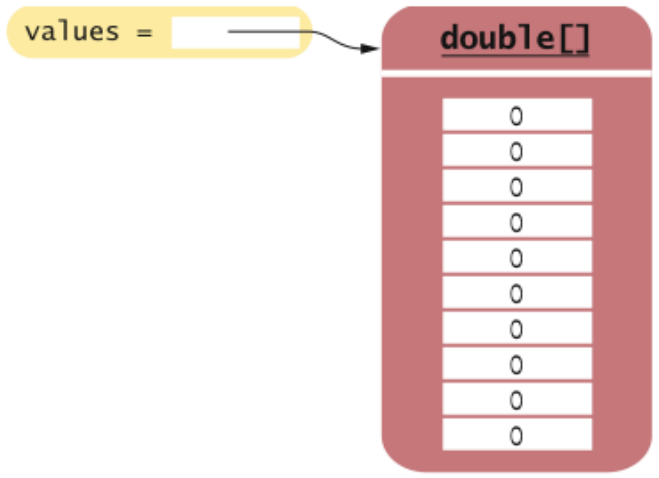
To access an element: `arrayReference[index]`

Example

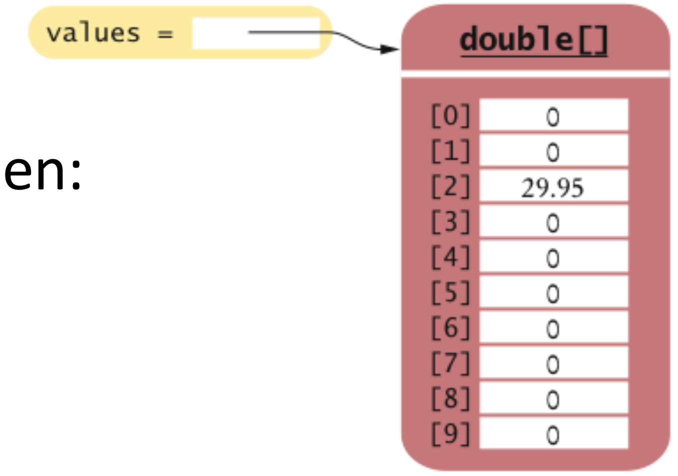
```
double[] values = new double[10];

double[] moreValues = { 32, 54, 67.5, 29, 35 };

values[i] = 29.95;
```

Name of array variable
Element type
Length
Type of array variable
Initialized with zero
Initialized with these elements
Use brackets to access an element.
The index must be ≥ 0 and < the length of the array.

Deklarasi dan Pemberian Nilai Array



1. Deklarasikan array:
   double[] value = new double[10];



2. Gunakan [ ] untuk mengakses elemen:
   value[2] = 29.95;

Deklarasi Array

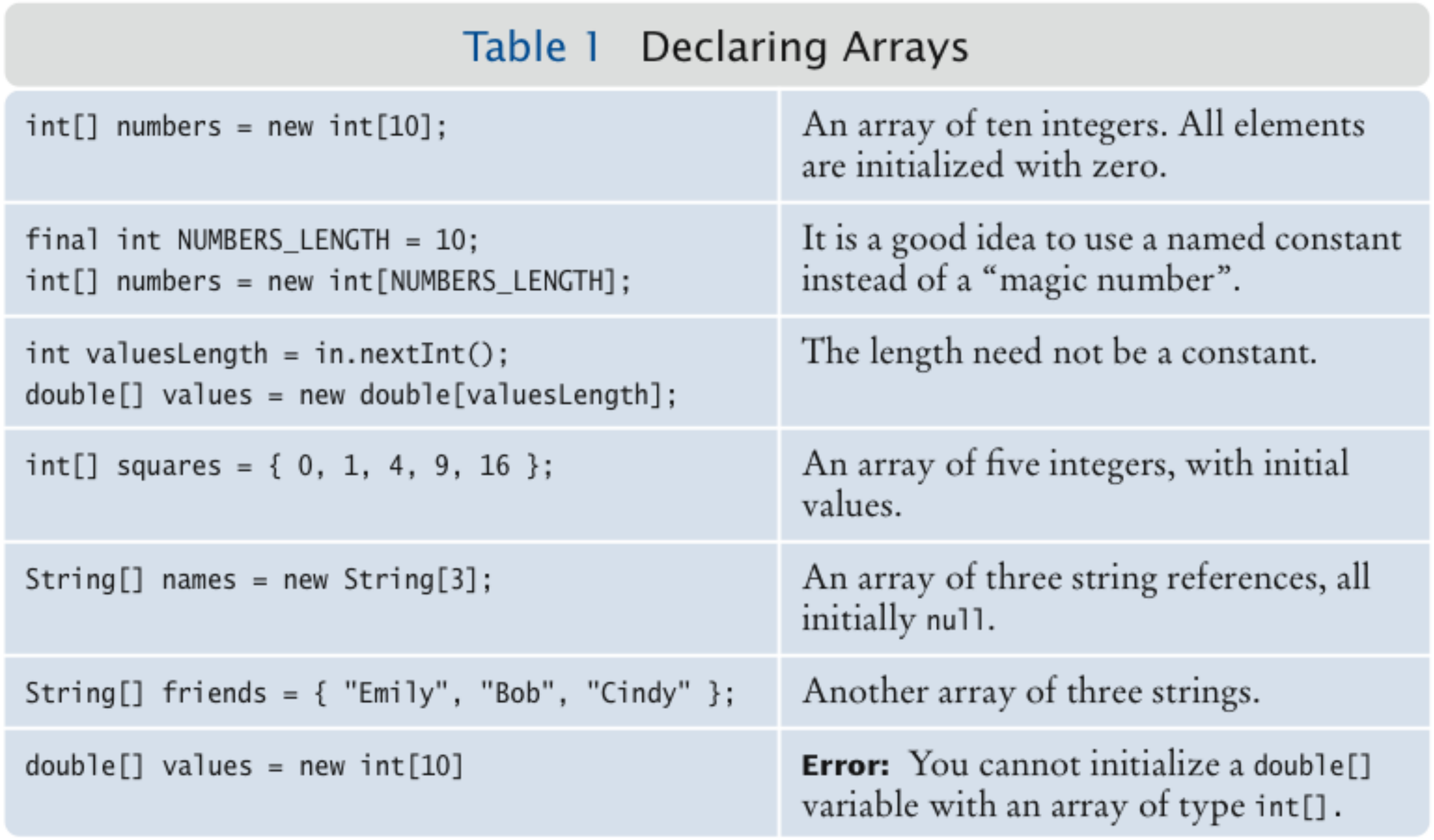


Table 1 Declaring Arrays

`int[] numbers = new int[10];`	An array of ten integers. All elements are initialized with zero.

`final int NUMBERS_LENGTH = 10;`
`int[] numbers = new int[NUMBERS_LENGTH];`
	It is a good idea to use a named constant instead of a “magic number”.

`int valuesLength = in.nextInt();`
`double[] values = new double[valuesLength];`
	The length need not be a constant.

`int[] squares = { 0, 1, 4, 9, 16 };`	An array of five integers, with initial values.
`String[] names = new String[3];`	An array of three string references, all initially `null`.
`String[] friends = { "Emily", "Bob", "Cindy" };`	Another array of three strings.
`double[] values = new int[10]`	**Error:** You cannot initialize a `double[]` variable with an array of type `int[]`.

Mengakses Elemen Array

- Bentuk Deklarasi namaVariabelArray[nomorElemen];

- Contoh:

  kota[0] = “Surabaya”;

ArrayKota.java

```
public class ArrayKota{
   public static void main(String[] args){
            String[] kota;                                    //deklarasi variabel array
            kota = new String[3];                             // membuat objek array

            // mengisi elemen array
            kota[0] = "Jakarta";
            kota[1] = "Surabaya";
            kota[2] = "Semarang";
            // menampilkan elemen array
            System.out.println(kota[0]);
            System.out.println(kota[1]);
            System.out.println(kota[2]);
   }
}
```

Pemberian Nilai Array Langsung

```
public class ArrayKota2{
	public static void main(String[] args){
		String[] kota = {“Jakarta”, “Surabaya”, “Semarang”};

		// menampilkan elemen array
		System out println(kota[0]);
		System.out.println(kota[1]);
		System.out.println(kota[2]);
	}
}
```

Mengetahui Jumlah Elemen Array

```
public class ArrayKota3{
        public static void main(String[] args){
                String[] kota = {“Jakarta”, “Surabaya”, “Semarang”};

                // menampilkan elemen array
                for(int i=0; i<kota.length; i++)
                        System.out.println(kota[i]);
        }
}
```

Array Multidimensi

```
class ArrayMultidimensi {
        public static void main(String[] args) {
                String[][] nama = {
                                {"Pak ", "Bu “, “Mbak”},
                                {"Joko", "Susi"}
                        };
                System.out.println(nama[0][0] + nama[1][0]);
                System.out.println(nama[0][1] + nama[1][1]);
                System.out.println(nama[0][2] + nama[1][0]);
        }
}
```

Array multidimensi adalah array dari array , dengan konsep pengaksesan [noBaris][noKolom]

Latihan: Buat Array Multidimensi

1. Buat class NegaraKota
2. Buat array multidimensi untuk nama negara dan ibukotanya
3. Masukkan dalam list array:
   nama negara = Amerika, Inggris, Jepang, Perancis, Indonesia, Iran, Irak
   ibukota = Teheran, Bekasi, Jakarta, Bantar Gebang, Tokyo
4. Akses array dan tampilkan di layar sebagai berikut:

   Ibukota Indonesia adalah Jakarta

   Ibukota Jepang adalah Tokyo

   Ibukota Iran adalah Teheran

Array Resizing

- Setelah array dibuat, array tidak dapat di-resize. Tapi,bisa dideklarasi ulang dengan menggunakan referensi yang sama.
- Contoh :

```
int[] myArray = new int[6];
myArray = new int[10];
```

Copy Array

`System.arraycopy()` method

- Java Programming language menyediakan spesial method pada `System` class, `arraycopy()`, untuk menyalin arrays.
- Contoh :

```
//source array
int src[] = {1,2,3,4,5,6};

//destination array
int dest[]= new int[src.length];
System.arraycopy(src,0,dest,0,src.length);

for (int i=0;i<dest.length;i++ )    {
       System.out.println(dest[i]);
}
```

ArrayList

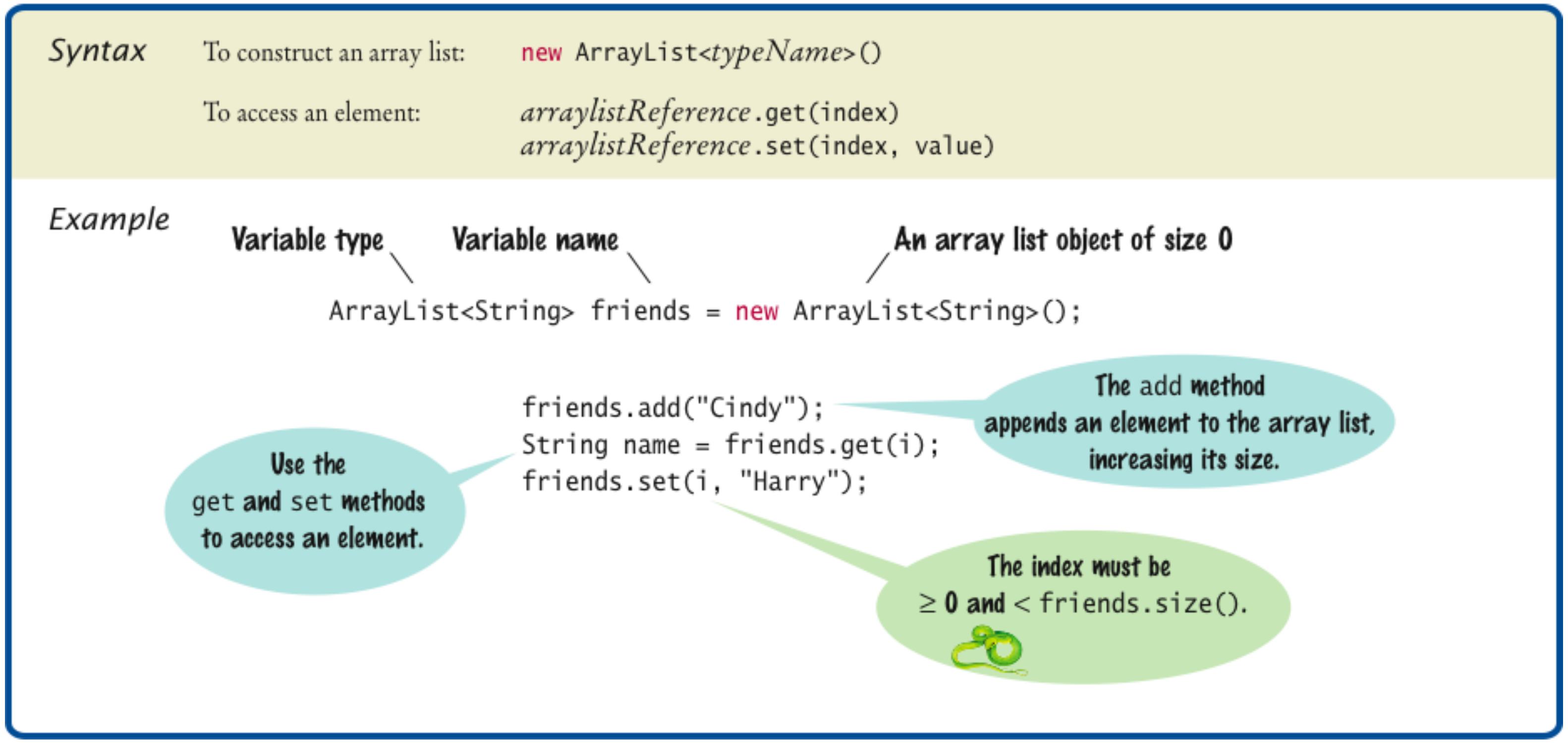
ArrayList

- `ArrayList` class mengelola urutan object, yang dapat bertambah dan berkurang sesuai dengan keperluan
- `ArrayList` class menyediakan banyak method untuk berbagi keperluan, misalnya menambah dan menghapus elemen
- `ArrayList` adalah suatu **generic class**:
- `ArrayList<T>` mengumpulkan object yang bertipe `T`:

```
ArrayList<String> names = new ArrayList<String>();
names.add("Emily");
names.add("Bob");
names.add("Cindy");
```

- `size` method untuk menghitung jumlah elemen

ArrayList



Syntax To construct an array list: `new ArrayList<typeName>()`

To access an element:

```
arraylistReference.get(index)
arraylistReference.set(index, value)
```

Example

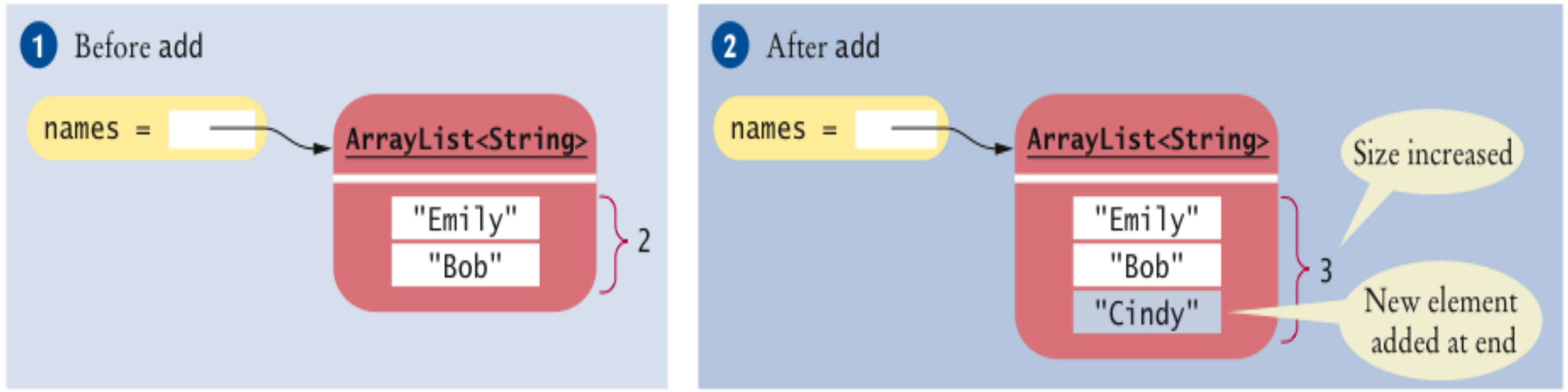
```
ArrayList<String> friends = new ArrayList<String>();

friends.add("Cindy");
String name = friends.get(i);
friends.set(i, "Harry");
```

Variable type
Variable name
An array list object of size 0
The add method appends an element to the array list, increasing its size.
Use the get and set methods to access an element.
The index must be ≥ 0 and < friends.size().

Menambahkan Elemen

- Untuk menambahkan sebuah elemen pada bagian akhir dari `ArrayList`, gunakan method `add` di bawah:

```
names.add("Emily");
names.add("Bob");
names.add("Cindy");
```



Menghapus Elemen

- Untuk menghapus elemen pada suatu indeks, menggunakan method `remove`:

```
names.remove(1);
```

Mendapatkan Nilai Elemen

- Untuk mendapatkan nilai elemen pada indeks, menggunakan metode `get`, dimana indeks dimulai dari 0

```
String name = names.get(2);
 //dapatkan elemen ketiga dari ArrayList
```

- Bila indeks keluar dari jangkauan, error akan keluar:

```
int i = names.size();
name = names.get(i); // Error
// legal index values are 0 ... i-1
```

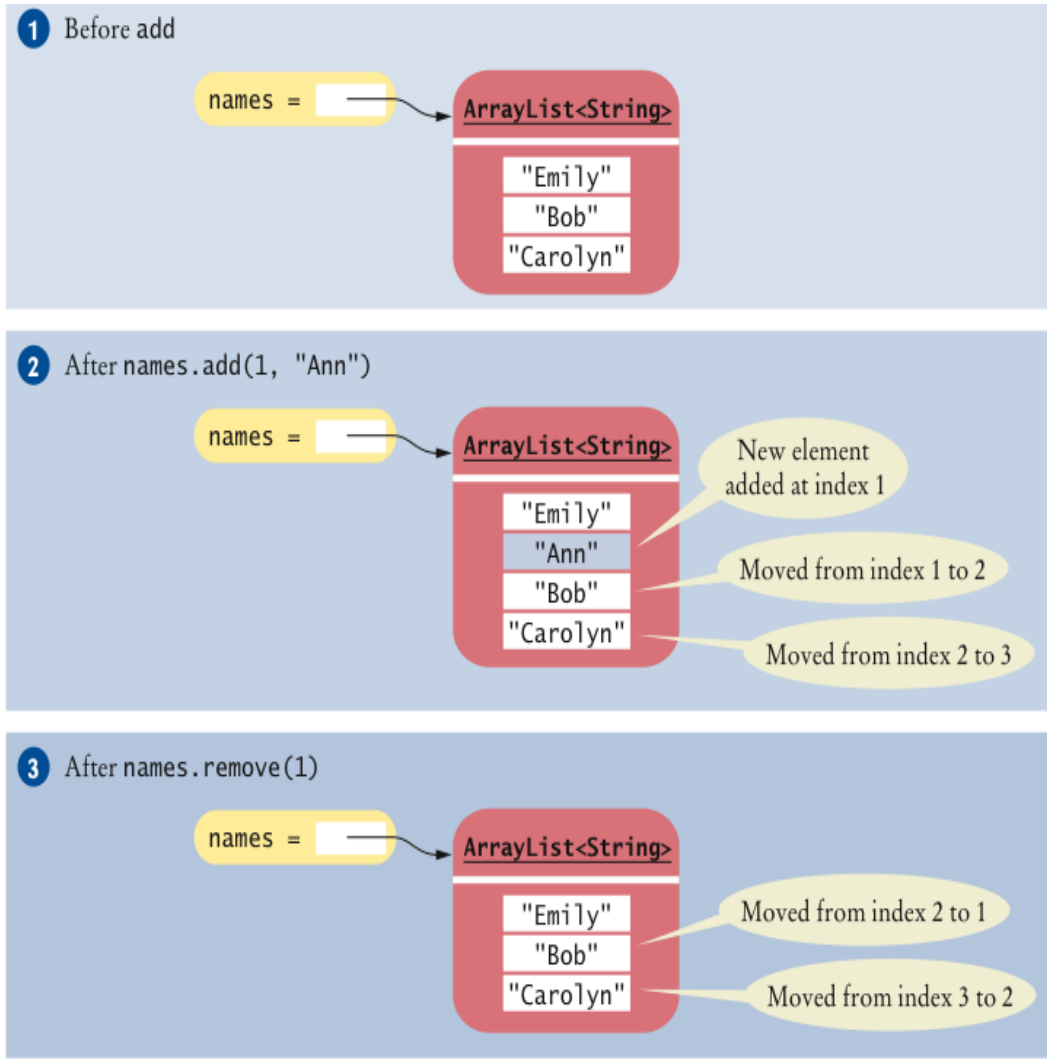
Menambah Nilai Baru ke Elemen

- Untuk menambahkan nilai baru ke elemen, digunakan method `set`:

```
names.set (2, "Carolyn");
```

Menambah dan Menghapus Elemen

```
names.add("Emily");
names.add("Bob");
names.add("Cindy");
names.set(2,"Carolyn"); 1
names.add(1,"Ann"); 2
names.remove(1); 3
```



ArrayList<String> names = new ArrayList<String>();	Constructs an empty array list that can hold strings

names.add("Ann");
names.add("Cindy");
	Adds elements to the end

System.out.println(names);	Prints [Ann, Cindy]
names.add(1, "Bob");	Inserts an element at index 1. names is now [Ann, Bob, Cindy]
names.remove(0);	Removes the element at index 0. names is now [Bob, Cindy]
names.set(0, "Bill");	Replaces an element with a different value. names is now [Bill, Cindy]
String name = names.get(i);	Gets an element
String last = **names.get(names.size() - 1);**	Gets the last element

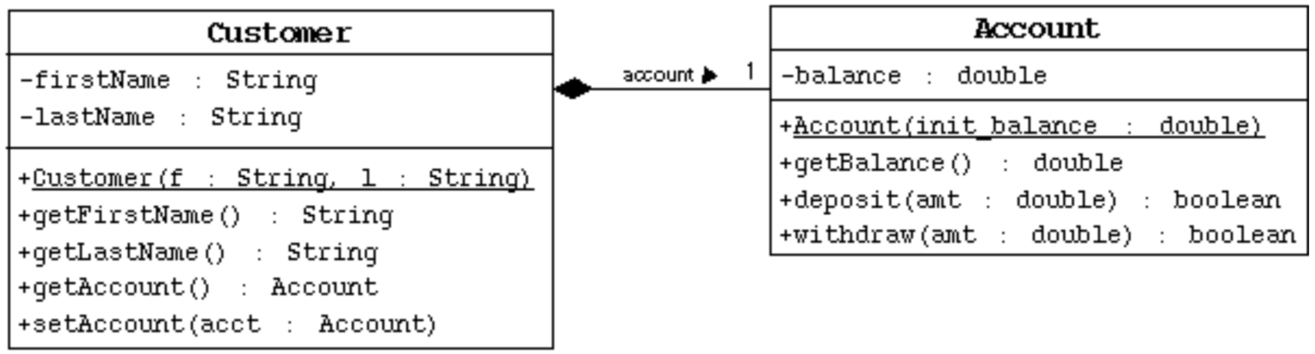
BankAccount.java

```java
public class BankAccount {
      private double balance;
      private int accountNumber;

      public BankAccount(int accountNumber){
         balance = 0;
         this.accountNumber = accountNumber;
      }

      public void deposit(double amount){
         balance = balance + amount;
      }

      public void withdraw(double amount){
         balance = balance - amount;
      }
   public int getAccountNumber(){
       return accountNumber;
    }

   public double getBalance(){
     return balance;
   }
}
```

BankAccountArrayBeraksi.java

```
public class BankAccountArrayBeraksi{
   public static void main(String[] args) {
            ArrayList<BankAccount> accounts = new ArrayList<BankAccount>();
            accounts.add(new BankAccount(1001));
            accounts.add(new BankAccount(1015));
            accounts.add(new BankAccount(1729));
            accounts.add(1, new BankAccount(1008));
            accounts.remove(0);

            System.out.println("Size: " + accounts.size());
            System.out.println("Expected: 3");
            BankAccount first = accounts.get(0);
            System.out.println("First account number: " + first.getAccountNumber());
            System.out.println("Expected: 1008");
            BankAccount last = accounts.get(accounts.size() - 1);
            System.out.println("Last account number: " + last.getAccountNumber());
            System.out.println("Expected: 1729");
   }
}
```

Latihan


Customer
-firstName : String
-lastName : String
+Customer(f : String, l : String)
+getFirstName() : String
+getLastName() : String
+getAccount() : Account
+setAccount(acct : Account)
account
1
Account
-balance : double
+Account(init_balance : double)
+getBalance() : double
+deposit(amt : double) : boolean
+withdraw(amt : double) : boolean

Exercise

Create the `Account` class in the file `Account.java`

- declare one private object attribute: balance; this attribute will hold the current (or "running") balance of the bank account
- declare a public constructor that takes one parameter (init_balance); that populates the balance attribute
- declare a public method getBalance that retrieves the current balance
- declare a public method deposit that adds the amount parameter to the current balance
- declare a public method withdraw that removes the amount parameter from the current balance

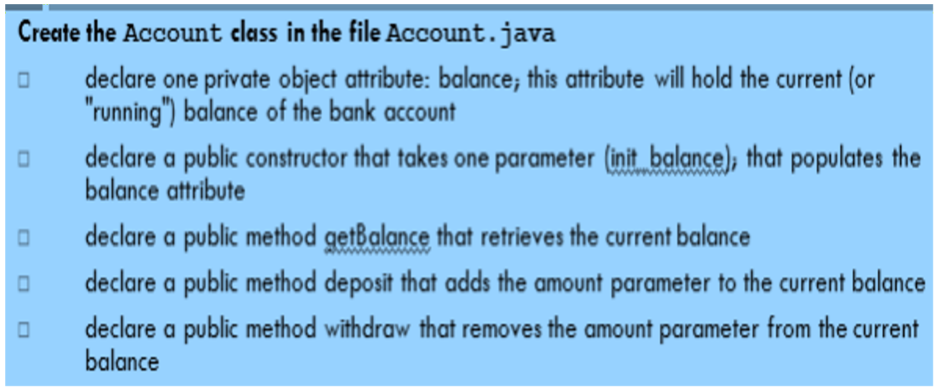
Create the `Customer` class in the file `Customer.java`

- declare three private object attributes: firstName, lastName, and **account**
- declare a public constructor that takes two parameters (f and l) that populate the object attributes
- declare two public accessors for the object attributes; these methods getFirstName and getLastName simply return the appropriate attribute
- declare the setAccount method to assign the account attribute
- declare the account method to retrieve the account attribute

Solution

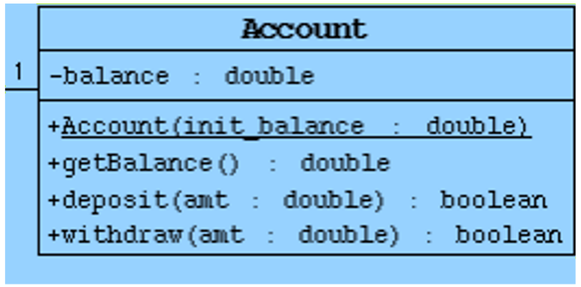
Account Class :

```
public class Account {
  protected double balance;
  public Account(double bal) {
    balance = bal;
  }
  public double getBalance() {
    return balance;
  }
  public boolean deposit(double amount) {
    if (amount>0) {
         balance = balance + amount;
        return true;
    } else
        return false;
}
   public boolean withdraw(double amount) {
    if ( balance >= amount ) {
      balance = balance - amount;
      return true;
    } else
      return false;
   }
}
```



Create the Account class in the file Account.java

- declare one private object attribute: balance; this attribute will hold the current (or "running") balance of the bank account
- declare a public constructor that takes one parameter (init_balance); that populates the balance attribute
- declare a public method getBalance that retrieves the current balance
- declare a public method deposit that adds the amount parameter to the current balance
- declare a public method withdraw that removes the amount parameter from the current balance

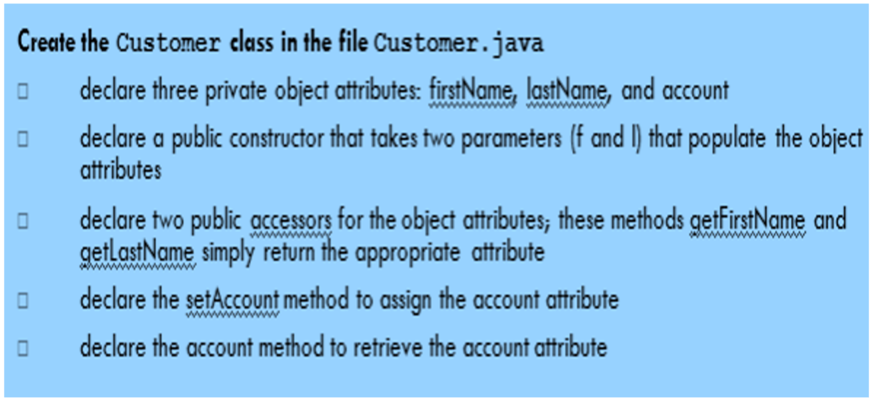


1
Account
-balance : double
+Account(init_balance : double)
+getBalance() : double
+deposit(amt : double) : boolean
+withdraw(amt : double) : boolean

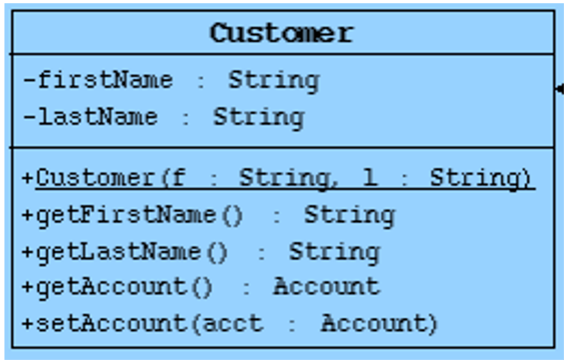
Solution

Customer Class :

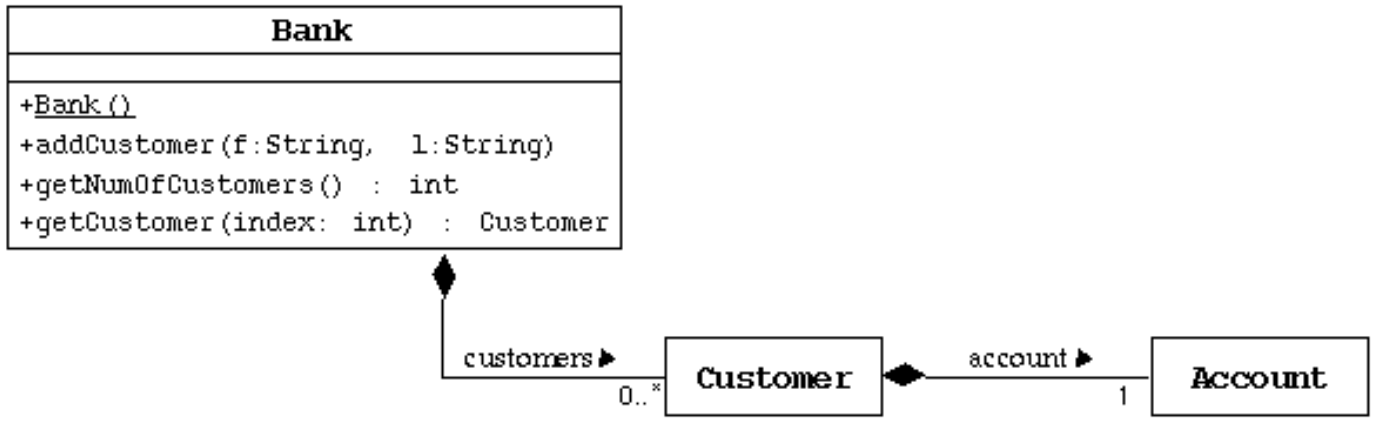
```
public class Customer {
  private String   firstName;
  private String   lastName;
  private Account[] accounts = new Account[5];
  private int numberOfAccounts = 0;

  public Customer(String f, String l) {
    firstName = f;
    lastName = l;
  }
  public String getFirstName() {
    return firstName;
  }
  public String getLastName() {
    return lastName;
  }
  public void setAccount(Account acct) {
    if (numberOfAccounts<5)
           accounts[numberOfAccounts++] = acct;
  }
  public Account getAccount(int account_index) {
    return accounts[account_index];
  }
  public int getNumOfAccounts() {
    return numberOfAccounts;
  }
}
```


Create the Customer class in the file Customer.java

- declare three private object attributes: firstName, lastName, and account
- declare a public constructor that takes two parameters (f and l) that populate the object attributes
- declare two public accessors for the object attributes; these methods getFirstName and getLastName simply return the appropriate attribute
- declare the setAccount method to assign the account attribute
- declare the account method to retrieve the account attribute



Exercise


Bank
+Bank()
+addCustomer(f:String, l:String)
+getNumOfCustomers() : int
+getCustomer(index: int) : Customer
customers ▶
0..*
Customer
account ▶
1
Account

Exercise

Create the `Bank` class in the file `Bank.java`

- Add two attributes to the Bank class: customers (an array of Customer objects) and numberOfCustomers (an integer that keeps track of the next customers array index).
- Add a public constructor that initializes the customers array with some appropriate maximum size (at least bigger than 5).
- Add the addCustomer method. This method must construct a new Customer object from the parameters (first name, last name) and place it on the customers array. It must also increment the numberOfCustomers attribute.
- Add the getNumOfCustomers accessor method, which returns the numberOfCustomers attribute.
- Add the getCustomer method. This method returns the customer associated with the given index parameter.

Solution

Bank Class :

```
public class Bank {
  private Customer[] customers = new Customer[5];
  private int numberOfCustomers;

  public Bank() {
    numberOfCustomers = 0;
  }
  public void addCustomer(String f, String l) {
    if (numberOfCustomers<5)
      customers[numberOfCustomers++] = new Customer(f, l);
  }
  public int getNumOfCustomers() {
    return numberOfCustomers;
  }
  public Customer getCustomer(int customer_index) {
    return customers[customer_index];
  }
}
```

TERIMA KASIH