SEMINAR

Ushtrim 1

In Java, it is also possible to nest classes (a class within a class). The purpose of nested classes is to group classes that belong together, which makes your code more readable and maintainable. To access the inner class, create an object of the outer class, and then create an object of the inner class:

```
class OuterClass {
  int x = 10;
  class InnerClass {
    int y = 5;
  }
}
public class Main {
  public static void main(String[] args) {
    OuterClass myOuter = new OuterClass();
    OuterClass.InnerClass myInner = myOuter.new InnerClass();
    System.out.println(myInner.y + myOuter.x);
  }
}
```

Ushtrim 2

Private Inner Class - Unlike a "regular" class, an inner class can be private or protected. If you don't want outside objects to access the inner class, declare the class as private:

```
class OuterClass {
  int x = 10;
  private class InnerClass {
    int y = 5;
  }
}
public class Main {
```

```
  public static void main(String[] args) {
    OuterClass myOuter = new OuterClass();
    OuterClass.InnerClass myInner = myOuter.new InnerClass();
    System.out.println(myInner.y + myOuter.x);
  }
}
```

If you try to access a private inner class from an outside class (MyMainClass), an error occurs:

```
Main.java:13: error: OuterClass.InnerClass has private access in OuterClass
    OuterClass.InnerClass myInner = myOuter.new InnerClass();
              ^
```

Ushtrim 3

static Inner Class

An inner class can also be static, which means that you can access it without creating an object of the outer class:

```
class OuterClass {
  int x = 10;
  static class InnerClass {
    int y = 5;
  }
}
public class Main {
  public static void main(String[] args) {
    OuterClass.InnerClass myInner = new OuterClass.InnerClass();
    System.out.println(myInner.y);
  }
}
```

Ushtrim 4

Te ndertohet kodi qe paraqet permbajtjen e nje tabele objektesh te tipit stringe, qe shton 4 elemente fruta, dhe paraqet elementin me indeks 1, ndryshon emertimin e elem me indeks 1, dhe ne fund afishon elementet e tabeles.

```
import java.util.*;
public class ArrayListExample4{
 public static void main(String args[]){
  ArrayList<String> al=new ArrayList<String>();
  al.add("Mango");
  al.add("Apple");
  al.add("Banana");
  al.add("Grapes");
  //accessing the element
  System.out.println("Returning element: "+al.get(1));//it will return the 2nd element, because index s
tarts from 0
  //changing the element
  al.set(1,"Dates");
  //Traversing list
  for(String fruit:al)
    System.out.println(fruit);
 }
}
```

Ushtrim 5

te ndertohet kodi qe afishon nese nje tabele objektesh eshte bosh ose jo, me pas shton 4 elem Brenda kesaj tabele dhe kthen true ose false ne varesi te permbajtjes se tabeles.

Java ArrayList example of isEmpty() method

```
import java.util.*;
 class ArrayList10{
       public static void main(String [] args) {
         ArrayList<String> al=new ArrayList<String>();
         System.out.println("Is ArrayList Empty: "+al.isEmpty());
         al.add("Ravi");
         al.add("Vijay");
```

```
        al.add("Ajay");
        System.out.println("After Insertion");
        System.out.println("Is ArrayList Empty: "+al.isEmpty());
      }
    }
```

ushtrim 6

Java ArrayList Example: Book - **Let's see an ArrayList example where we are adding books to list and printing all the books.**

```
import java.util.*;
class Book {
int id;
String name,author,publisher;
int quantity;
public Book(int id, String name, String author, String publisher, int quantity) {
    this.id = id;
    this.name = name;
    this.author = author;
    this.publisher = publisher;
    this.quantity = quantity;
}
}
public class ArrayListExample20 {
public static void main(String[] args) {
    //Creating list of Books
    List<Book> list=new ArrayList<Book>();
    //Creating Books
    Book b1=new Book(101,"Let us C","Yashwant Kanetkar","BPB",8);
    Book b2=new Book(102,"Data Communications and Networking","Forouzan","Mc Graw Hill",4)
;
    Book b3=new Book(103,"Operating System","Galvin","Wiley",6);
    //Adding Books to list
    list.add(b1);
```

```
    list.add(b2);
    list.add(b3);
    //Traversing list
    for(Book b:list){
        System.out.println(b.id+" "+b.name+" "+b.author+" "+b.publisher+" "+b.quantity);
    }
}
}
```

ushtrim 7

paraqit elementet e tabeles ne inversin e tyre.

```
public class ReverseArrayList {
    public static void main(String[] args) {
        List<String> l = new ArrayList<String>();
        l.add("Mango");
        l.add("Banana");
        l.add("Mango");
        l.add("Apple");
        System.out.println("Before Reversing");
        System.out.println(l.toString());

        Collections.reverse(l);
        System.out.println("After Reversing");
        System.out.println(l);
    }
```

ushtrim 8

te hiqen elementet e duplikuar ne tabele

```
public class RemoveDuplicateArrayList {
    public static void main(String[] args) {
        List<String> l = new ArrayList<String>();
        l.add("Mango");
        l.add("Banana");
        l.add("Mango");
```

```
        l.add("Apple");
        System.out.println(l.toString());
        Set<String> s = new LinkedHashSet<String>(l);
        System.out.println(s);
    }
}
```

ushtrim 9

klasa Student e cila ka tiparet/properties si:

emri Student, nr I regjistrimit dhe mosha student

```
public class Student  {
    private String studentname;
    private int rollno;
    private int studentage;

    public Student(int rollno, String studentname, int studentage) {
        this.rollno = rollno;
        this.studentname = studentname;
        this.studentage = studentage;
    }

    public String getStudentname() {
        return studentname;
    }
    public void setStudentname(String studentname) {
            this.studentname = studentname;
    }
    public int getRollno() {
            return rollno;
    }
    public void setRollno(int rollno) {
            this.rollno = rollno;
    }
    public int getStudentage() {
            return studentage;
    }
    public void setStudentage(int studentage) {
            this.studentage = studentage;
    }
}
```

deshirojme te kemi nje ArrayList te objekteve Student si me poshte:

```
import java.util.*;
public class ArrayListSorting  {

    public static void main(String args[]){
            ArrayList<Student> arraylist = new ArrayList<Student>();
            arraylist.add(new Student(223, "Chaitanya", 26));
            arraylist.add(new Student(245, "Rahul", 24));
            arraylist.add(new Student(209, "Ajeet", 32));

            Collections.sort(arraylist);

            for(Student str: arraylist){
                        System.out.println(str);
            }
    }
}
package beginnersbook.com;

public class Student implements Comparable {
   private String studentname;
   private int rollno;
   private int studentage;

   public Student(int rollno, String studentname, int studentage) {
      this.rollno = rollno;
      this.studentname = studentname;
      this.studentage = studentage;
   }
   ...
   //getter and setter methods same as the above example
   ...
   @Override
   public int compareTo(Student comparestu) {
      int compareage=((Student)comparestu).getStudentage();
      /* For Ascending order*/
      return this.studentage-compareage;

      /* For Descending order do like this */
      //return compareage-this.studentage;
   }

   @Override
   public String toString() {
      return "[ rollno=" + rollno + ", name=" + studentname + ", age=" + studentage + "]";
   }
```

```
}
```

Now we can very well call Collections.sort on ArrayList

```
import java.util.*;
public class ArrayListSorting  {

		public static void main(String args[]){
		  ArrayList<Student> arraylist = new ArrayList<Student>();
		  arraylist.add(new Student(223, "Chaitanya", 26));
		  arraylist.add(new Student(245, "Rahul", 24));
		  arraylist.add(new Student(209, "Ajeet", 32));

		  Collections.sort(arraylist);

		  for(Student str: arraylist){
					System.out.println(str);
		  }
		}
}
```

ushtrim 10

- How to create an ArrayList using the `ArrayList()` constructor.
- Add new elements to an ArrayList using the `add()` method.

```
import java.util.ArrayList;
import java.util.List;
public class CreateArrayListExample {
    public static void main(String[] args) {
        // Creating an ArrayList of String
        List<String> animals = new ArrayList<>();
        // Adding new elements to the ArrayList
        animals.add("Lion");
        animals.add("Tiger");
        animals.add("Cat");
```

```
        animals.add("Dog");
        System.out.println(animals);
        // Adding an element at a particular index in an ArrayList
        animals.add(2, "Elephant");
        System.out.println(animals);
    }
}
```

ushtrim 11

This example shows:

- How to create an ArrayList from another collection using the `ArrayList(Collection c)` constructor.
- How to add all the elements from an existing collection to the new ArrayList using the `addAll()` method

```
import java.util.ArrayList;
import java.util.List;
public class CreateArrayListFromCollectionExample {
    public static void main(String[] args) {
        List<Integer> firstFivePrimeNumbers = new ArrayList<>();
        firstFivePrimeNumbers.add(2);
        firstFivePrimeNumbers.add(3);
        firstFivePrimeNumbers.add(5);
        firstFivePrimeNumbers.add(7);
        firstFivePrimeNumbers.add(11);
```

```
        // Creating an ArrayList from another collection
        List<Integer> firstTenPrimeNumbers = new ArrayList<>(firstFivePrimeNumbers);
        List<Integer> nextFivePrimeNumbers = new ArrayList<>();
        nextFivePrimeNumbers.add(13);
        nextFivePrimeNumbers.add(17);
        nextFivePrimeNumbers.add(19);
        nextFivePrimeNumbers.add(23);
        nextFivePrimeNumbers.add(29);
        // Adding an entire collection to an ArrayList
        firstTenPrimeNumbers.addAll(nextFivePrimeNumbers);
        System.out.println(firstTenPrimeNumbers);
    }
}
```

Ushtrim 12

This example shows:

- How to check if an ArrayList is empty using the `isEmpty()` method.
- How to find the size of an ArrayList using the `size()` method.
- How to access the element at a particular index in an ArrayList using the `get()` method.
- How to modify the element at a particular index in an ArrayList using the `set()` method.

```
package projekt2;
import java.util.ArrayList;
import java.util.List;
public class AccessElementsFromArrayListExample {
    public static void main(String[] args) {
        List<String> topCompanies = new ArrayList<>();
```

```
        // Check if an ArrayList is empty
        System.out.println("A eshte lista e kompanive bosh? : " + topCompanies.isEmpty());
        topCompanies.add("Google");
        topCompanies.add("Apple");
        topCompanies.add("Microsoft");
        topCompanies.add("Amazon");
        topCompanies.add("Facebook");
        // Find the size of an ArrayList
        System.out.println("Ketu paraqiten " + topCompanies.size() + " top kompanite ne bote");
        System.out.println(topCompanies);
        // Retrieve the element at a given index
        String bestCompany = topCompanies.get(0);
        String secondBestCompany = topCompanies.get(1);
        String lastCompany = topCompanies.get(topCompanies.size() - 1);
        System.out.println("Best kompania: " + bestCompany);
        System.out.println("Best kompania e dyte: " + secondBestCompany);
        System.out.println("kompania e fundit ne liste: " + lastCompany);
        // Modify the element at a given index
        topCompanies.set(4, "Walmart");
        System.out.println("Lista e Modifikuar e top kompanive: " + topCompanies);
    }
}
```

Ushtrim

ArrayList of user defined objects- Since ArrayList supports generics, you can create an ArrayList of any type. It can be of simple types like Integer, String, Double or complex types like an ArrayList of ArrayLists, or an ArrayList of HashMaps or an ArrayList of any user defined objects. In the following example, you'll learn how to create an ArrayList of user defined objects

```
import java.util.ArrayList;
```

```java
import java.util.List;

class User {
    private String name;
    private int age;

    public User(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }
}
```

```
public class ArrayListUserDefinedObjectExample {

    public static void main(String[] args) {

        List<User> users = new ArrayList<>();

        users.add(new User("Rajeev", 25));

        users.add(new User("John", 34));

        users.add(new User("Steve", 29));


        users.forEach(user -> {

            System.out.println("Name : " + user.getName() + ", Age : " +
user.getAge());

        });

    }

}
```