دستور کار ۴

Object composition

- what is object composition?
  - Object composition in Java refers to the practice of creating complex objects by combining or "composing" simpler objects. It involves creating new classes that contain instances of other classes as member variables, allowing them to collaborate and work together to achieve a desired behavior or functionality.

Object composition

- benefits of object composition:
  - Adds code reusability
  - Compensating for non-inheritance from multiple Java classes
  - Makes debugging the code easier
  - No restriction in naming methods compared to inheritance

Strings

- Strings are **immutable** objects in java
- when modified, a new String object will be created and returned
- String is a **built-in** object, so no need to import it in order to use it

popular String methods

- `str.length()`
- `str.concat(String str2)`
- `str.equals(String str2)`
- `str.indexOf(int chUnicode)`
  - `str.indexOf(char c)`
  - `str.indexOf(String str2)`
- `str.toUpperCase()`
- `str.toLowerCase()`
- `str.toCharArray()`
- `str.split(String s2,int limit)`
- `str.replace(char oldChar, char newChar)`

popular String methods

- `str1.compareTo(String str2)`
  - returns 0 if str1 and str2 are equal
  - returns + number if str1 is bigger than str2
  - returns - number if str1 is smaller than str2
  - bonus: find out what exactly this method returns in different situations
- `str.charAt(int index)`
- `str.subString(int beginIndex, int lastIndex)`

Javadoc

- why do we use javadoc?
  - Javadoc is a documentation generation tool in Java that allows developers to create documentation for their code. It is widely used to generate API documentation for Java libraries, frameworks, and applications.
- the compiler does not process these javadocs
- `/** place javadoc here`
- `@author`
- `@param`
- `@throws`
- `@version`
- `@returns`
- `*/`

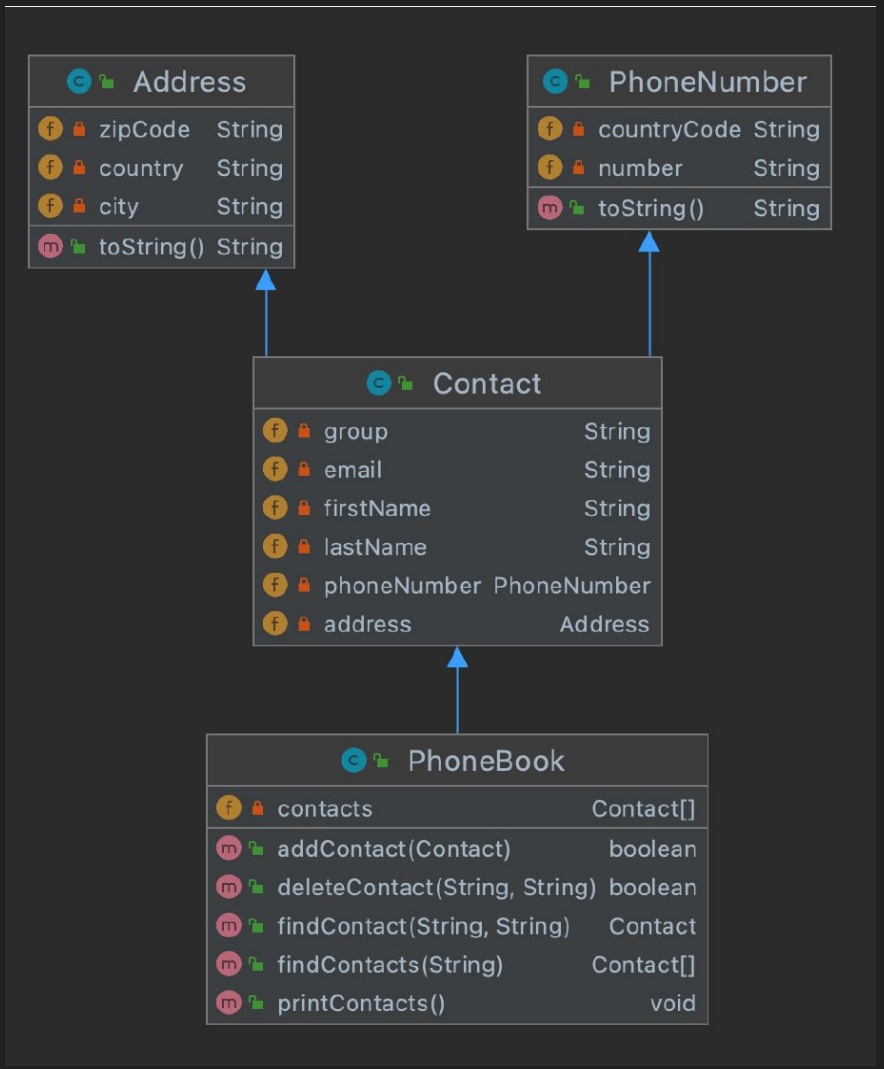
```java
/**
* <h1>Add Two Numbers!</h1>
* The AddNum program implements an application that
* simply adds two given integer numbers and Prints
* the output on the screen.
* <p>
* <b>Note:</b> Giving proper comments in your program makes it more
* user friendly and it is assumed as a high quality code.
*
* @author  ap-ce-aut
* @version 1.0
* @since   2021-08-24
*/

public class AddNum {
   /**
    * This method is used to add two integers. This is
    * a the simplest form of a class method, just to
    * show the usage of various javadoc Tags.
    * @param numA This is the first parameter to addNum method
    * @param numB  This is the second parameter to addNum method
    * @return int This returns sum of numA and numB.
    */
   public int addNum(int numA, int numB) {
       return numA + numB;
   }

   /**
    * This is the main method which makes use of the addNum method.
    * @param args Unused.
    * @return Nothing.
    */

   public static void main(String[] args) {
       AddNum obj = new AddNum();
       int sum = obj.addNum(10, 20);

       System.out.println("Sum of 10 and 20 is :" + sum);
   }
}
```

toString

- The `toString` method in Java is a built-in method defined in the Object class, which is the root class for all Java classes. It is used to obtain a string representation of an object.
- However, it is a common practice to override the toString method in custom classes to provide a more meaningful and human-readable representation of the object's state.
- When you print an object, the toString method of that object is called

```java
public class Person {
    2 usages
    private String name;
    2 usages
    private int age;
    no usages   new *
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    // Overriding the toString method
    1 usage   new *
    @Override
    public String toString() {
        return "Person{name='" + name + "', age=" + age + "}";
    }
    // Rest of the class implementation
    // ...
}
```

Today’s task


Address
zipCode String
country String
city String
toString() String
PhoneNumber
countryCode String
number String
toString() String
Contact
group String
email String
firstName String
lastName String
phoneNumber PhoneNumber
address Address
PhoneBook
contacts Contact[]
addContact(Contact) boolean
deleteContact(String, String) boolean
findContact(String, String) Contact
findContacts(String) Contact[]
printContacts() void