Mücahit Kurtlar 19290259

1) Namespace, Java'da package'a tekabül etmektedir. Bünyesinde birden fazla sınıf barındırabilir. Proje ise solution denebilir.
Bu nedenle aynı sınıf için "private" kullanılabilir.
Aynı paket/namespace için "protected" kullanılabilir.
Farklı paket/namespace için "protected internal" kullanılabilir.
Farklı proje/solution için "public" kullanılabilir.

2)

```
using System;
namespace DArray{
    class Person{
        public Person(String n){
            this.name = n;
        }
        public String name;
    }

    class Program {

        Static void Main(string [] args] {
```

2.a)

```
            Person [] arr1 = new Person[4];
            for ( int i = 0; i<4; i++) {
                arr1[i] = new Person("person[" + i.ToString() + "]"); }
            foreach (var per in arr1) {
                Console.WriteLine(per.name);
            }
```

2.b)

```
            Person [,] arr2 = new Person[2,2];
            for (int i = 0; i<2; i++) {
                for(int j = 0; j<2; j++) {
                    arr2 [i, j] = new Person("person[" i.ToString() + ", " + j.ToString + "]");
                }
            }
            for (int i=0; i<2; i++){for(in j= 0; j<2; j++){
                Console.WriteLine (arr2[i, j].name); }}
        }
    }
```

CamScanner ile tarandı

3)

3.a) "Ayşe"

3.b) "Ali 2"

3.c)

```
interface IEmployee {
    void showDetail();
}

class Employee : IEmployee {
        public String name;
        public Employee (Strin n){
          this.name =n;
        }
        public void showDetail(){
            Console.WriteLine(this.name);
        }
}
class Engineer : Employee {
        public int dnumber;
        public Engineer(string n, int d): base(n) {
            base.(n);
            this.dnumber=d;
        }
        public void showDetail() {
            Console.WriteLine(name + " "+ this.dnumber);
        }
class Main {
        public static void main(Strig[] args){
          Employee ee = new Engineer("Ayşe",1); ee.showDetail();
          IEmployee ie = new Engineer("Ali",2); ee.showDetail();
}
```

4)

```
public class Person
{
   String name;
   public Person(String n){ name = n;}
   public override String ToString(){
      return name;
   }

   public override bool Equals(object o){
      if (o == null){
         return false;
      }
      if (!(o is Person)){
         return false;
      }
      return (this.name == ((Customer)o).name);
   }
}
```

5)

5.a)

```
public class Complex {
   public int real;
   public int imag;
   public Complex(int r, int i){
      this.real = r;
      this.imag = i;
   }
}
```

5.b)

```
public static Complex operator +(Complex f, Complex s){
      return new Complex(f.real + s.real, f.imag + s.imag);
}
```