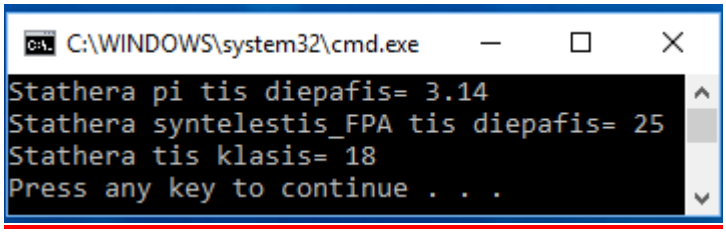
ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΛΥΣΕΙΣ ΑΣΚΗΣΕΩΝ
Διεπαφές / Διασυνδέσεις *(Interfaces)*

ΑΣΚΗΣΗ-1η (Ορισμός σταθερών σε Διεπαφή)

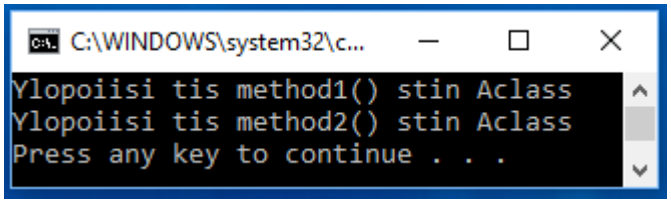
```
interface OrismosConstants {
  double pi=3.14;
  int syntelestis_FPA=25; }

class TestConstants implements OrismosConstants {
  public static final int CONST = 18;

  public static void main(String args[])  {
     System.out.println("Stathera pi tis diepafis= "+pi);
     System.out.println("Stathera syntelestis_FPA tis diepafis=
                    "+syntelestis_FPA);
     System.out.println("Stathera tis klasis= "+CONST);
}}
```



ΑΣΚΗΣΗ-2η (Υλοποίηση μεθόδων της διεπαφής (κλασική χρήση), σε κλάση που υλοποιεί την διεπαφή)

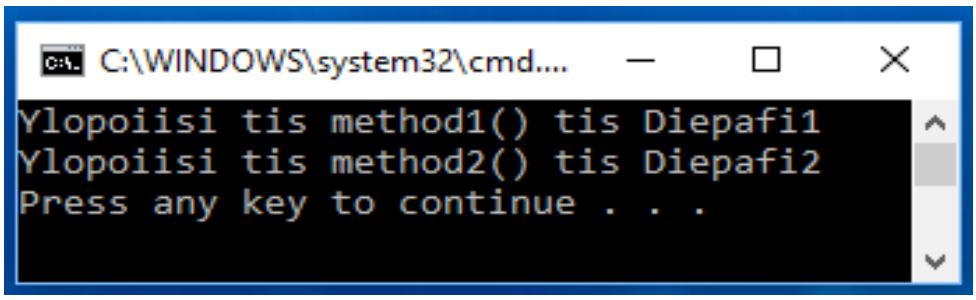
```
interface MyDemoInterface {
  public void method1();
  public void method2();}
class Aclass implements MyDemoInterface {
 public void method1() {
   System.out.println("Ylopoiisi tis method1() stin Aclass"); }
```

```
public void method2() {
  System.out.println("Ylopoiisi tis method2() stin Aclass"); }

public static void main(String arg[]){
    MyDemoInterface obj = new Aclass();
    obj. method1();
    obj. method2(); } }
```



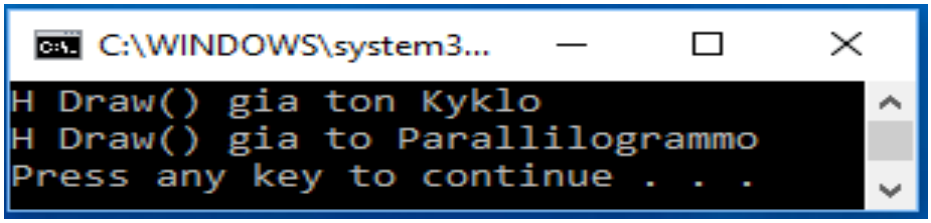
ΑΣΚΗΣΗ-3η (Κληρονομικότητα διεπαφής από άλλη διεπαφή)

```
interface Diepafi1{
  public void method1();}

interface Diepafi2 extends Diepafi1 {
  public void method2();}

class Demo implements Diepafi2{

 public void method1(){
   System.out.println("Ylopoiisi tis method1() tis Diepafi1");}

 public void method2(){
   System.out.println("Ylopoiisi tis method2() tis Diepafi2");}}

class TestInterface2 {
 public static void main(String args[]) {
  Demo obj = new Demo();
  obj. method1();
  obj. method2(); }}
```



ΑΣΚΗΣΗ-4η (Υλοποίηση μεθόδων διεπαφής – Παραλλαγή άσκησης - 4)

```
interface Shape
  {public void draw();}

 class Circle implements Shape {
   public void draw()
    {System.out.println("H Draw() gia ton Kyklo");}}

 class Rectangle implements Shape {
   public void draw()
    {System.out.println("H Draw() gia to Parallilogrammo");}}

 class ShapeConstruction {
   public Shape getShape(String s){
    if (s.equals("Circle")){return new Circle();}
    if (s.equals("Rectangle")){return new Rectangle();}
    return null; }}

class TestInterfaces {
    public static void main(String args[]) {
     ShapeConstruction sc=new ShapeConstruction();
     Shape sh1=sc.getShape("Circle");
     sh1.draw();
     Shape sh2=sc.getShape("Rectangle");
     sh2.draw();
} }
```



ΑΣΚΗΣΗ-5η (Υλοποίηση μεθόδων διεπαφής)

```
interface Emvadon {
  float computeEmvadon(float x, float y);}

class Rectangle implements Emvadon {
 public float computeEmvadon(float x, float y){
```

```
  return(x * y);}}

class Triangle implements Emvadon {
 public float computeEmvadon(float x, float y) {
  return(x * y/2);}}

class InterfaceEmvadon {
 public static void main(String args[]){
  Rectangle rect = new Rectangle();
  Triangle tri = new Triangle();
  Emvadon emv;  //αναφορά
  emv = rect;
  System.out.println("To Emvadon tou Rectangle(2,4)= "+ emv.computeEmvadon(2,4));
  emv = tri;
  System.out.println("To Emvadon tou Triangle(6, 8)= "+  emv.computeEmvadon(6,8));
} }
```