

A decorative graphic on the left side of the slide, consisting of a network of white lines and small circles on a dark blue background, resembling a circuit board or a neural network.

JAVA INTERFACE

ПРЕПОДАВАТЕЛ: ИНЖ. В. МАРИНОВА

Java Interface

Друг начин за абстракции в Java, е използването на интерфейс. Интерфейсът е напълно абстрактен клас, който се използва да групира свързани методи с празно тяло.

// интерфейс

```
interface Animal {
```

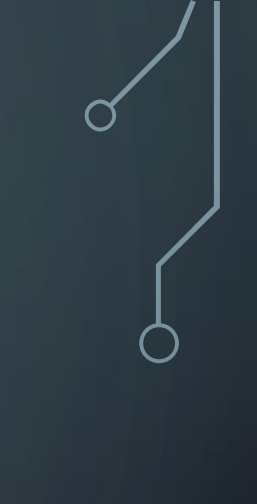

// интерфейсен метод (няма описание на тялото на метода)

```
public void animalSound();
```

// интерфейсен метод (няма описание на тялото на метода)

```
public void run();
```

```
}
```



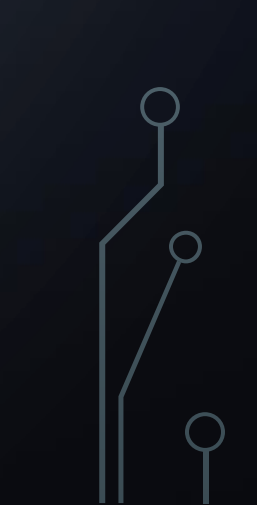

Достъпът до интерфейсният метод става като интерфейсът трябва да бъде имплементиран от друг клас (наследен).

Използва се служебната дума : **implements**

Тялото на интерфейсният метод е празно и се описва в имплементиращия клас.

Както и при абстрактните класове, интерфейсът не може да се използва за създаване на обекти.

При имплементирането на интерфейс, ние трябва да опишем всички негови методи.







Интерфейсните методи по подразбиране са : `abstract` и `public`.

Интерфейсните атрибути по подразбиране са: `public`, `static`, `final`.

Интерфейсът не може да съдържа конструктор (тъй като интерфейсът не може да бъде използван за създаване на обекти).

Интерфейсът се използва за скриване на детайлите и показване само на важните елементи и методи от обекта.



// интерфейс

```
interface Animal {  
    public void animalSound(); // интерфейсен метод  
    public void sleep(); // интерфейсен метод  
}
```

// Pig имплементира интерфейса Animal

```
class Pig implements Animal {  
    public void animalSound() {  
        // тялото на animalSound() се описва тук  
        System.out.println("The pig says: wee wee");  
    }  
    public void sleep() {  
        // тялото на sleep() се описва тук  
        System.out.println("Zzz");  
    }  
}
```



//Главната програма :

```
class MyMainClass {  
    public static void main(String[] args) {  
  
        // създаваме обект Pig  
        Pig myPig = new Pig();  
  
        // извикваме метода animalSound()  
        myPig.animalSound();  
  
        // извикваме метода sleep()  
        myPig.sleep();  
    }  
}
```

Multiple Interfaces

Можем да имплементираме повече от един интерфейс в даден клас, като се използва изброяване със запетая.

Java не поддържа "multiple inheritance" (класът може да бъде наследник само на един superclass).

Множественото наследяване се реализира чрез интерфейси, тъй като клас може да имплементира множество интерфейси.

//пример:

```
interface FirstInterface {  
    public void myMethod(); // интерфейсен метод  
}
```

```
interface SecondInterface {  
    public void myOtherMethod(); // интерфейсен метод  
}
```

```
class DemoClass implements FirstInterface, SecondInterface {  
    public void myMethod() {  
        System.out.println("Some text..");  
    }  
    public void myOtherMethod() {  
        System.out.println("Some other text...");  
    }  
}
```





```
class MyMainClass {
```

```
    public static void main(String[] args) {
```

```
        // създаваме обект myObj от DemoClass
```

```
        DemoClass myObj = new DemoClass();
```

```
        // извикваме метода myMethod() от FirstInterface
```

```
        myObj.myMethod();
```

```
        // извикваме метода myOtherMethod() от SecondInterface
```

```
        myObj.myOtherMethod();
```

```
    }
```

```
}
```



The image features a dark blue background with a subtle gradient. In the corners, there are decorative white line art elements resembling circuit boards or neural network connections, with small circles at the end of the lines. The central text is in a bold, orange, sans-serif font, with a slight reflection effect below it.

Благодаря за вниманието!