

به نام خدا

آموزش جاوا به صورت خلاصه (بخش اول)



مقدمه:

این جزوه برای کسانی نوشته شده است که حداقل دانش برنامه نویسی (آشنایی با C یا C#) را دارند.

به همین دلیل همه چیز به صورت کاملاً خلاصه بیان شده است.

آشنایی با جاوا:

جاوا در سال 1990 منتشر شد که در ابتدا نام آن oak بود و بعد از استقبال به java که برگرفته از نام سازندگان اصلی آن می باشد تغییر یافت. انتخاب اسم جاوا در یک شاپ انجام شد و شاید به همین خاطر باشد که logo آن یک فنجان داغ است.

جاوا بسیار شبیه به زبان C می باشد. بنابراین کسانی که با این زبان آشنا هستند کار بسیار آسانی برای فراگیری java دارند.

جاوا دارای کتابخانه های بسیار قوی می باشد که یکی از دلایل برتری آن نسبت به دیگر زبان ها است. دلایل اصلی برتری در اجرا شدن برنامه های جاوا بر روی اکثر سیستم ها می باشد. زیرا JVM باعث شده است که نرم افزار های java بر روی سیستم هایی که معماری متفاوتی دارند به راحتی اجرا شود. دلیل دیگر برتری استانداردهایی مثل j2ee می باشد که به ما اجازه تولید یک برنامه تجاری و حتی ایجاد یک FrameWork را می دهد. سیستم های تجاری مثل ERP ها نیز بیشتر با جاوا کار می شوند. همین دلایل برای یادگیری جاوا کافی است.

نظرات خود را به ایمیل rezajafari2a@gamil.com ارسال نمایید.

با سپاس

نویسنده:

رضا جعفری

برنامه های لازم:

1- نرم افزار Netbeans جهت برنامه نویسی.

<http://dlc.sun.com.edgesuite.net/netbeans/7.4/final/bundles/netbeans-7.4-javaee-windows.exe>

2-دانلود کتاب خانه جاوا یا jdk 1.7

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

3- برای برنامه نویسی تحت وب،جاوا مربوط به مرورگر خود را دانلود کنید.

نصب:

1-ابتدا jdk را نصب کنید

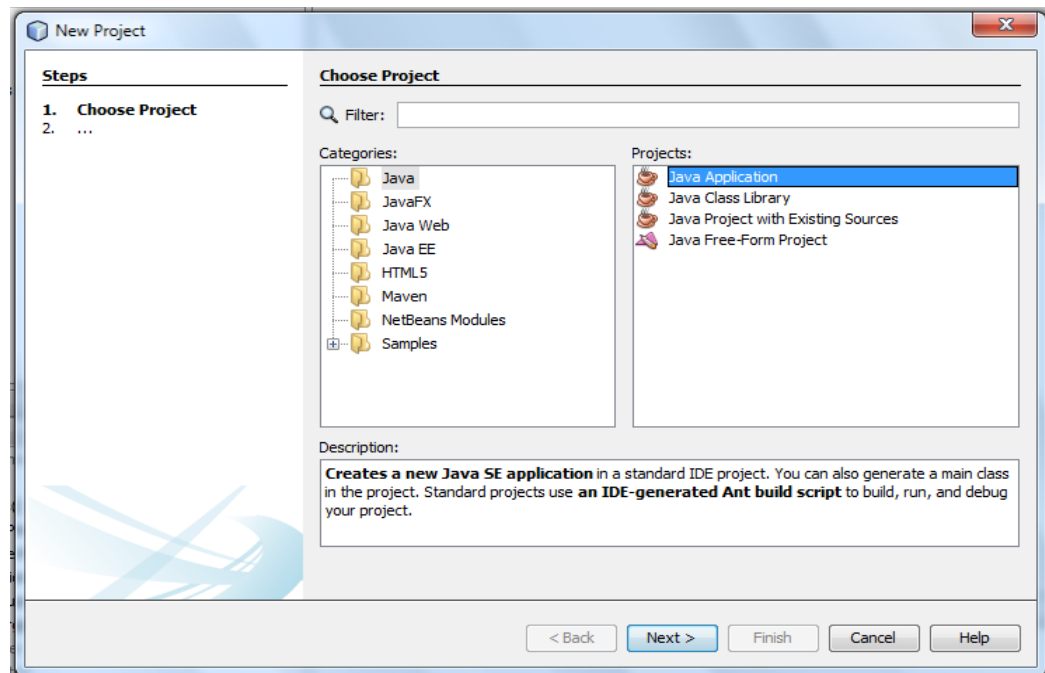
2- بعد نصب مورد بالا،netbeans را نصب کنید.

در هنگام نصب ، آدرس jdk از شما خواسته می شود.اگر آدرس پیش فرض را تغییر نداده باشید آدرس آن برابر مقدار زیر است.

C:\Program Files\Java\jdk1.7.0_51

-----پایان نصب

بعد از باز کردن برنامه از منوی file گزینه new project را انتخاب نمایید و سپس java application



بر روی next کلیک کنید و در صفحه بعد اسم پروژه و محل ذخیره سازی آن را تعیین کنید و سپس finish

--- توضیح قالب برنامه:

`package firstproject;`

یک package می تواند شامل چند کلاس باشد.

`public class FirstProject {`

`}`

هر برنامه اصلی در داخل یک کلاس تعریف می شود.

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

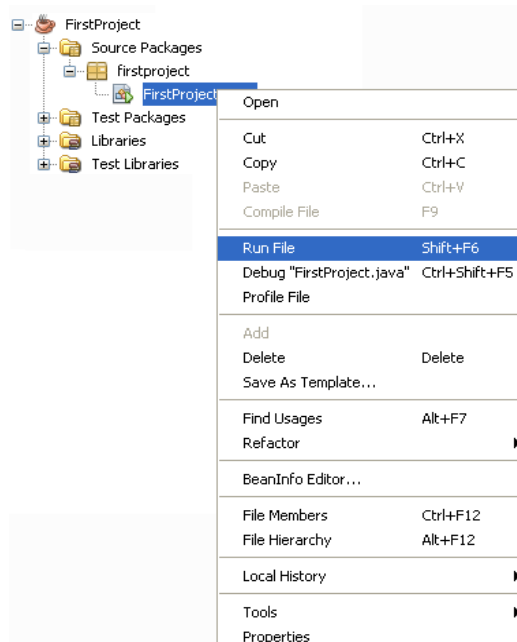
`}`

کدهای اجرایی در این بخش(قالب اصلی) نوشته می شوند.

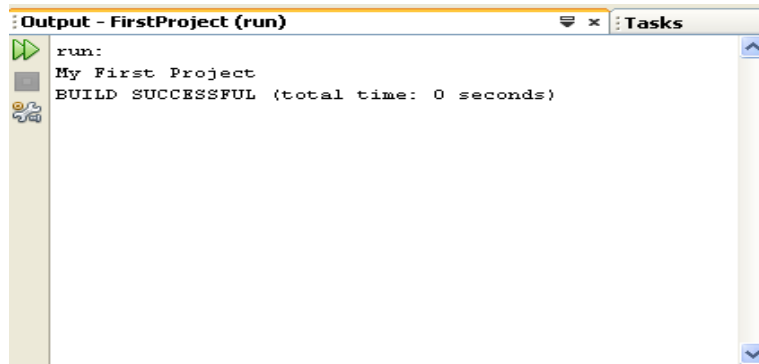
--- نحوه RUN

با استفاده از کلید های : **shift+f6**

توصیه می شود بر روی کلاس کلیک راست شود و سپس گزینه run انتخاب شود.(زیرا وقتی تعداد کلاس ها زیاد باشد، با روش اول، اولین کلاس اجرا می شود)



---خروجی



---آموزش دستورها

1-نظر یا توضیح

comment:

روش اول:	روش دوم:
<pre> /* توضیحات */ </pre>	<pre> //توضیح </pre>

روش سوم:

```

/**
 * توضیحات
 */

```

2- چاپ در خروجی

```

public static void main( String[ ] args ) {

    System.out.println( "رشته خروجی" );

}

```

3-متغیر ها

عدد صحیح // int x=2;

رشته // String str="HI";

کاراکتر // char ch='a';

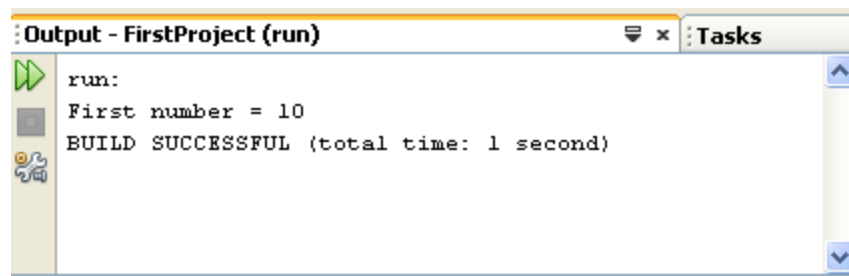
`double first_number=10.0;`//double تعریف نوع

`float first_number= 10.5f;`// float تعریف نوع

مثال: نمایش یک عدد در خروجی.

```
public static void main(String[] args) {
    int first_number;
    first_number = 10;
    System.out.println("My First Project" + first_number);
}
```

خروجی:



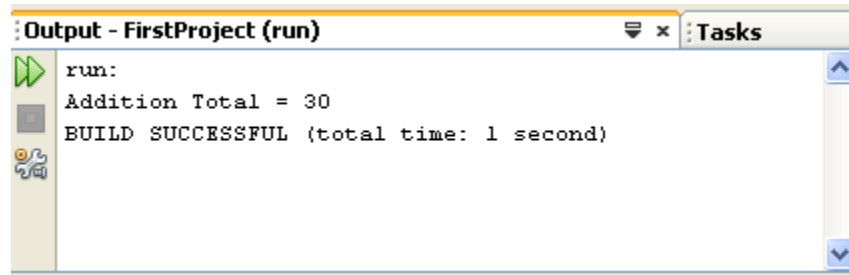
مثال: جمع دو عدد

```
public static void main(String[] args) {
    int first_number, second_number, answer;

    first_number = 10;
    second_number = 20;
    answer = first_number + second_number;

    System.out.println("Addition Total = " + answer );
}
```

خروجی:



مثال : برنامه ای که فرمول زیر را حساب کند

`answer = first_number - (second_number + third_number);`

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

    int first_number, second_number, third_number, answer;

    first_number = 100;
    second_number = 75;
    third_number = 25;
    answer = first_number - (second_number + third_number);

    System.out.println("Total = " + answer );

}
```

خروجی: صفر

مثال: برنامه نمایش اسم و فامیل.(اسم و فامیل در دو رشته جدا است)

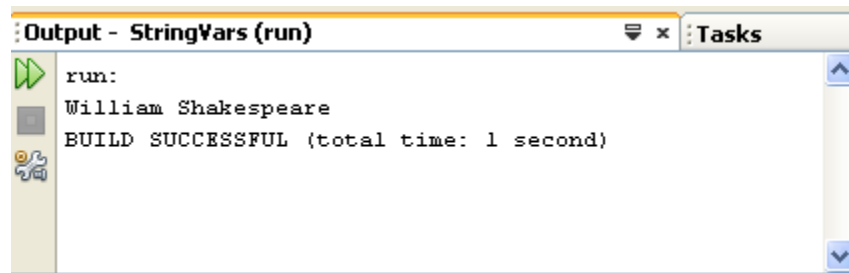
```
public static void main(String[] args) {

    String first_name = "William";
    String family_name = "Shakespeare";

    System.out.println(first_name + " " + family_name);

}
```

خروجی:



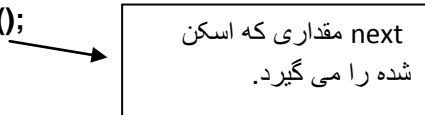
--گرفتن ورودی از کاربر

```
import java.util.Scanner;
```

ابتدا دستور فوق را در زیر `package firstproject;` می نویسیم.

```
Scanner متغیر = new Scanner( System.in );
```

```
نام متغیر.next();
```



مثال: خواندن نام

```
String first_name;
```

```
Scanner user_input = new Scanner( System.in );
```

```
first_name = user_input.next( );
```

توضیح: در خط اول یک رشته تعریف می کنیم تا مقداری که کاربر وارد می کند را بعداً در آن ذخیره کنیم.

خط دوم یک متغیر از نوع `scanner` تعریف کردیم تا ورودی که کاربر وارد می کند را ذخیره کند.

متد `next` برای خواندن مقدار اسکن شده از متغیر می باشد.

زیرا نوع متغیر اسکن و رشته با یک دیگر متفاوت هستند. مثل رشته و عدد صحیح که بدون استفاده از متد های تبدیل نمی توان آنها را به یکدیگر تبدیل کرد.

مثال: خواندن نام ، نام خانوادگی و نمایش در خروجی

```
package stringvars;

import java.util.Scanner;

public class StringVariables {

    public static void main(String[] args) {

        Scanner user_input = new Scanner(System.in);

        String first_name;
        System.out.print("Enter your first name: ");
        first_name = user_input.next();

        String family_name;
        System.out.print("Enter your family name: ");
        family_name = user_input.next();

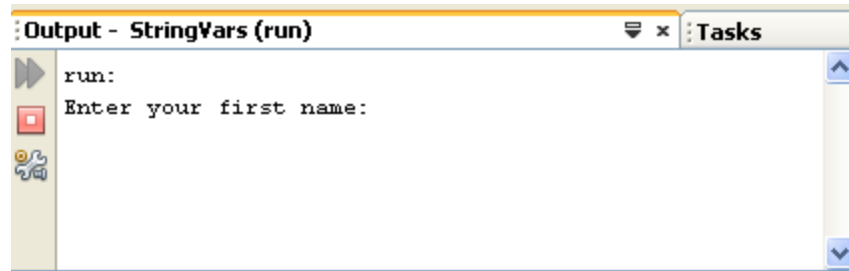
        String full_name;
        full_name = first_name + " " + family_name;

        System.out.println("You are " + full_name);

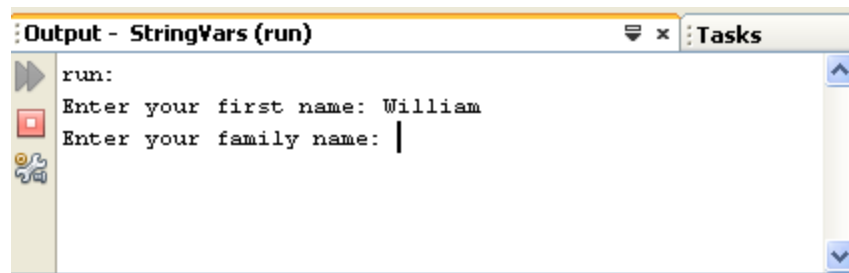
    }
}
```


--خروجی

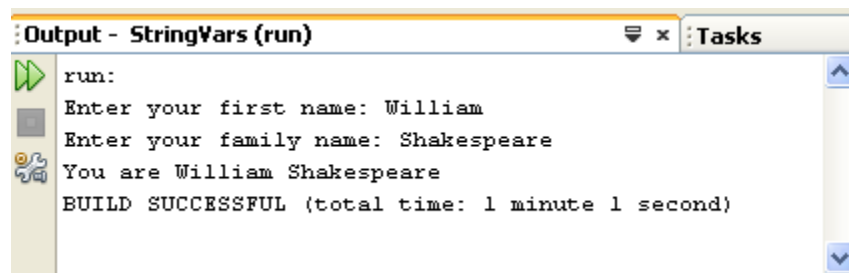
خواندن نام



خواندن نام خانوادگی:

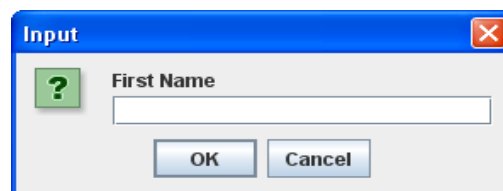


خروجی:



--خواندن با استفاده از Option Panes

Option Panes: یعنی شکل زیر (خواندن به صورت گرافیکی!!!)



این نوع از ورودی در کتاب خانه زیر قرار دارد پس باید آن را import کنیم.

```
import javax.swing.JOptionPane;
```

```
JOptionPane.showInputDialog("First Name");
```

مثال: خواندن نام

```
String first_name;
first_name = JOptionPane.showInputDialog("First Name");
```

خط دوم برای خواندن و ذخیره در رشته **first_name** است.

نمایش خروجی با استفاده از **JOptionPane**:

مثال **messagebox.show()** در **c#**

```
JOptionPane.showMessageDialog( null, "رشته" );
```

چرا null؟ زیرا message box با چیز دیگری در برنامه در ارتباط نیست. (وابسته نیست)

نکته: بهتر است وقتی از **joption** برای نمایش خروجی استفاده می شود، در آخر برنامه از کد زیر استفاده شود تا برنامه به طور کامل خاتمه یابد و عمل **G.C** به طور کامل انجام شود.

G.C: Garbage Collection

یعنی مدیریت حافظه و برگرداندن فضای اشغال شده به رم

مثال: خواندن نام و نام خانوادگی و نمایش

```
package userinput;
import javax.swing.JOptionPane;

public class InputBoxes {

    public static void main(String[] args) {

        String first_name;
        first_name = JOptionPane.showInputDialog("First Name");

        String family_name;
        family_name = JOptionPane.showInputDialog("Family Name");

        String full_name;
        full_name = "You are " + first_name + " " + family_name;

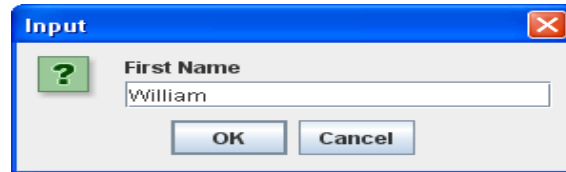
        JOptionPane.showMessageDialog(null, full_name);
        System.exit(0);

    }

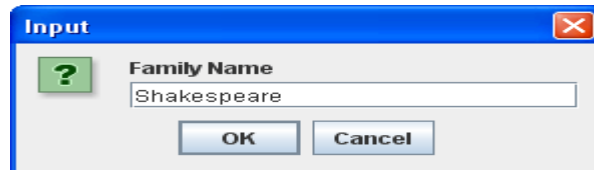
}
```

خروجی:

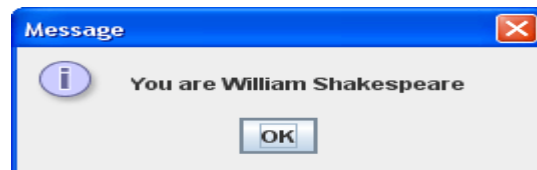
خواندن نام:

An input dialog box titled "Input" with a close button (X) in the top right corner. It contains a green question mark icon, a label "First Name", a text field containing "William", and "OK" and "Cancel" buttons at the bottom.

خواندن نام خانوادگی:

An input dialog box titled "Input" with a close button (X) in the top right corner. It contains a green question mark icon, a label "Family Name", a text field containing "Shakespeare", and "OK" and "Cancel" buttons at the bottom.

نمایش نام و نام خانوادگی:

A message dialog box titled "Message" with a close button (X) in the top right corner. It contains an information icon (i), the text "You are William Shakespeare", and an "OK" button at the bottom.

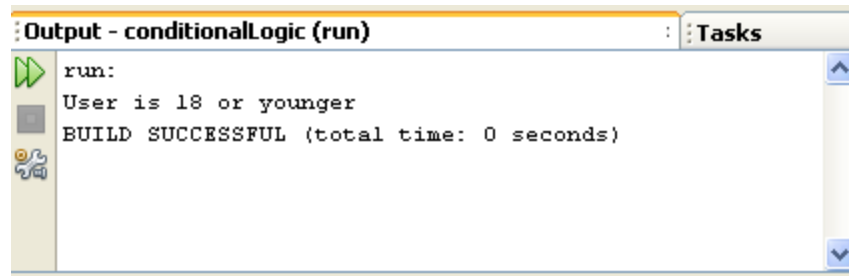
---دستور شرطی if

```
if ( شرط ) {  
    دستورات  
}
```

مثال: بگوید سن کاربر از 18 بزرگتر مساوری یا کوچکتر است

```
public static void main(String[] args) {  
  
    int user = 18;  
  
    if (user <= 18) {  
        System.out.println("User is 18 or younger");  
    }  
  
    if (user > 18) {  
        System.out.println("User is older than 18");  
    }  
  
}
```

خروجی:



---دستور if else

```
if ( شرط ) {
    //دستورات
}
else {
    //دستورات
}
```

مثال: تعیین حدود سن با شروط. >> کوچکتر مساوی 18 << >> بزرگتر از 18 و کوچکتر از 40 << >> بزرگتر از 40

```
public static void main(String[] args) {
    int user = 21;

    if (user <= 18) {
        System.out.println("User is 18 or younger");
    }
    else if (user > 18 && user < 40) {
        System.out.println("User is between 19 and 39");
    }

    else {
        System.out.println("User is older than 40");
    }
}
```

خروجی: شرط دوم اجرا می شود.

اسم متغیر

---تعریف متغیر boolean یا پرچم

`boolean flage = true;`

مثال: برنامه ای بنویسید که مشخص است کند مقدار `flag` صحیح (`true`) یا اشتباه (`false`) است.

`boolean flage = true;`

```
if (flage == true) {
    System.out.println("it's true");
}
else {
    System.out.println("it's false");
}
```

نکته: دو دستور زیر معاداً اند.

<code>if (! flage)</code>	<code>if (flage ==false)</code>
-----------------------------	---------------------------------

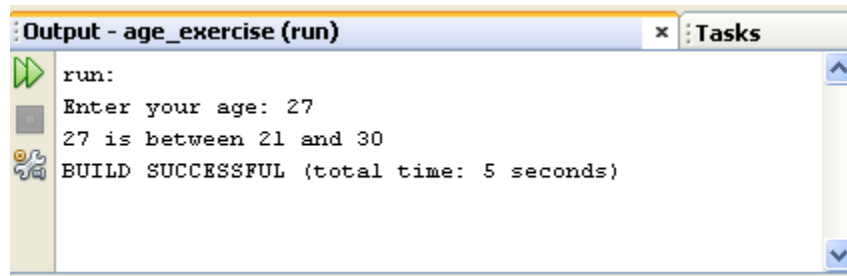
---دستور switch

```
{ (متغیری که می خواهیم آن را تست کنیم)
case مقدار تست:
    دستور;
break;
case مقدار تست:
    دستور;
break;
default//حالتی که هیچ کدام یک از موارد بالا اجرا نشود
    دستور;
}
```

مثال: برنامه ای که بگوید سن شما 18 یا 19 یا 20 و یا هیچ کدام است!!!

```
public static void main(String[] args) {
    int user = 18;
    switch ( user ) {
        case 18:
            System.out.println("You're 18");
            break;
        case 19:
            System.out.println("You're 19");
            break;
        case 20:
            System.out.println("You're 20");
            break;
        default:
            System.out.println("You're not 18, 19 or 20");
    }
}
```

خروجی:



```
Output - age_exercise (run)
run:
Enter your age: 27
27 is between 21 and 30
BUILD SUCCESSFUL (total time: 5 seconds)
```

---حلقه for: همان حلقه c است.

for (**گام**; **شرط پایان**; **مقدار شروع**) {

دستورات

}

مثال: نمایش اعداد صفر تا 11

```
package loops;

public class ForLoops {

    public static void main(String[] args) {

        int loopVal;
        int end_value = 11;

        for (loopVal = 0; loopVal < end_value; loopVal++) {

            System.out.println("Loop Value = " + loopVal);

        }

    }

}
```

خروجی:

```

Output - loops (run)
run:
Loop Value = 0
Loop Value = 1
Loop Value = 2
Loop Value = 3
Loop Value = 4
Loop Value = 5
Loop Value = 6
Loop Value = 7
Loop Value = 8
Loop Value = 9
Loop Value = 10
BUILD SUCCESSFUL (total time: 0 seconds)

```

حلقه while: همان حلقه c است.

while (شرط) {

//دستورات

}

حلقه do while: همان حلقه c است.

do {
//دستورات
}

while (شرط خاتمه);

int[] aryNums = new int[6];

اسم متغیر

تعداد خانه های آرایه

---آرایه در جاوا:

روش های مقدار دهی به آرایه:

aryNums[1] = 14;

//-----روش دوم

int[] aryNums = { 1, 2, 3, 4 };

مثال از انواع دیگر آرایه:

//آرایه به صورت رشته

String[] aryStrings = {"Autumn", "Spring", "Summer", "Winter"};

//آرایه از نوع boolean یا پرچم

boolean[] aryBools = {false, true, false, true};

به دست آوردن طول آرایه:

اسم آرایه.length;

مثال: برنامه ای بنویسید که آرایه با گام های های حلقه مقدار دهی شود.

```
package prjarrays;

public class ArraysTest {

    public static void main(String[] args) {

        int[] lottery_numbers = new int[49];
        int i;

        for (i=0; i < lottery_numbers.length; i++) {
            lottery_numbers[i] = i + 1;
            System.out.println( lottery_numbers[i] );
        }

    }
}
```

--مرتب سازی آرایه: تابع مرتب سازی در جاوا موجود است.

نکته: به منظور استفاده از تمام ویژگی های آرایه ها دو دستور زیر را در ابتدا بنویسید

```
import java.util.Arrays;
```

```
import java.util.Collections;
```

```
-----//
```

```
Arrays.sort(اسم آرایه); //برای مرتب سازی
```


مثال: آرایه ای تعریف کرده سپس آن را مرتب کنید؟

```
package prjarrays;

import java.util.Arrays;

public class ArraysTest {

    public static void main(String[] args) {

        int[] aryNums;
        aryNums = new int[6];

        aryNums[0] = 10;
        aryNums[1] = 14;
        aryNums[2] = 36;
        aryNums[3] = 27;
        aryNums[4] = 43;
        aryNums[5] = 18;

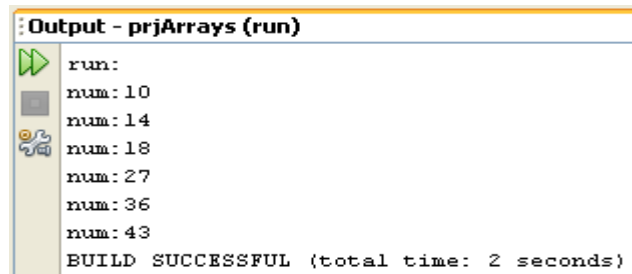
        Arrays.sort(aryNums);

        int i;

        for (i=0; i < aryNums.length; i++) {
            System.out.println("num:" + aryNums[i]);
        }

    }
}
```

خروجی:



```
Output - prjArrays (run)
run:
num: 10
num: 14
num: 18
num: 27
num: 36
num: 43
BUILD SUCCESSFUL (total time: 2 seconds)
```

مثال: آرایه ای از جنس رشته تعریف کرده و بعد از مرتب کردن، آن را نمایش دهید.

```
package strings1;
import java.util.Arrays;

public class StringArrays {

    public static void main(String[] args) {

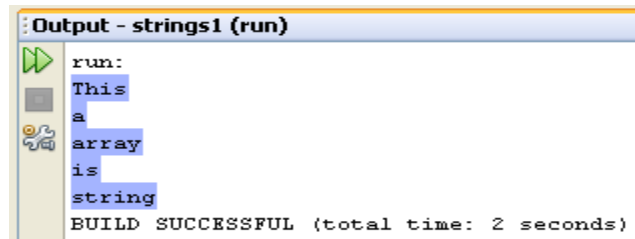
        String[] aryString = new String[5] ;

        aryString[0] = "This";
        aryString[1] = "is";
        aryString[2] = "a";
        aryString[3] = "string";
        aryString[4] = "array";

        Arrays.sort(aryString);

        int i;
        for (i=0; i < aryString.length; i++){
            System.out.println( aryString[i] );
        }
    }
}
```

خروجی:



```
Output - strings1 (run)
run:
This
a
array
is
string
BUILD SUCCESSFUL (total time: 2 seconds)
```

---تعریف آرایه دوبعدی:

```
int[ ][ ] aryNumbers = new int[6][5];
```

مثال: یک آرایه 2 بعدی را با استفاده از گامهای حلقه مقدار دهی کنید؟

```
int rows = 6;
int columns = 5;

int i, j;

for (i=0; i < rows ; i++) {

    for (j=0; j < columns ; j++) {
        System.out.print( aryNumbers[i][j] + " " );
    }
    System.out.println( "" );
}
```

خروجی:

```
Output - prjArrays (run)
run:
10 12 43 11 22
20 45 56 1 33
30 67 32 14 44
40 12 87 14 55
50 86 66 13 66
60 53 44 12 11
BUILD SUCCESSFUL (total time: 2 seconds)
```

--لیست آرایه: از کتابخانه زیر در ابتدا استفاده شود.

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

تعریف لیست آرایه ای:

```
ArrayList listTest = new ArrayList( );
```

افزودن به لیست:

```
listTest.add( "first item" );
listTest.add( "second item" );
listTest.add( "third item" );
listTest.add( 7 );
```

خواندن از لیست :

```
listTest.get( 3 );
```

گرفتن مقدار خانه شماره 3 از لیست آرایه ای

حذف از لیست:

```
listTest.remove(2);
```

حذف مقدار خانه شماره 2 از لیست آرایه ای

حذف از لیست با استفاده از مقدار درون لیست.

```
listTest.remove( "second item" );
```

حذف خانه از لیست آرایه ای که مقدارش برابر رشته زیر است

استخراج کل لیست آرایه ای: افزودن کتابخانه زیر به برنامه

```
import java.util.Iterator;
```

نوعی از داده می باشد که با استفاده از آن می توان به همه داده های لیست آرایه ای دسترسی داشت

```
Iterator it = listTest.iterator( );
```

اتصال لیست آرایه ای به متغیر مربوطه

```
while ( it.hasNext( ) ) {  
System.out.println( it.next( ) );  
}
```

تا وقتی که مقداری در لیست است دستور زیر را انجام بده

گرفتن مقدار از لیست. هر بار مقدار بعدی را می خواند.

استخراج:

--تبدیل رشته به حروف بزرگ و کوچک

تبدیل رشته به حروف بزرگ:

```
str.toUpperCase( );
```

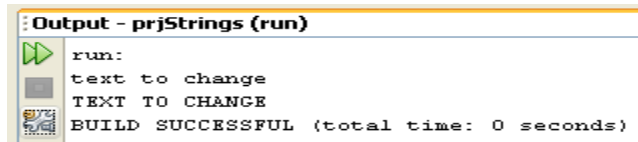
تبدیل رشته به حروف کوچک:

```
str.toLowerCase ( );
```

مثال: رشته ای را به حروف بزرگ تبدیل کنید؟

```
package prjstrings;  
  
public class StringManipulation {  
  
    public static void main(String[] args) {  
  
        String changeCase = "text to change";  
        System.out.println( changeCase );  
  
        String result;  
        result = changeCase.toUpperCase();  
  
        System.out.println( result );  
    }  
}
```

خروجی:



--مقایسه رشته ها: دستور زیر دو رشته را از هم کم می کند. حاصل اگر صفر شود برابراند. اگر بزرگتر صفر باشد، رشته اول بزرگتر است و اگر کوچکتر از صفر باشد رشته دوم بزرگتر است.

Word1.compareTo(Word2);

مثال: دو کلمه را مقایسه کند و حالات آنها را مقایسه کند؟

```
package prjstrings;

public class StringManipulation {

    public static void main(String[] args) {

        int result;
        String Word1 = "Ape";
        String Word2 = "App";

        result = Word1.compareTo(Word2);

        if (result < 0) {
            System.out.println("Word1 is less than Word2");
        }
        else if (result > 0) {
            System.out.println("Word1 is more than Word2");
        }
        else if (result == 0) {
            System.out.println("The same word");
        }
    }
}
```

--متد **indexOf**: این متد یک کاراکتر را می گیرد و موقعیت آن را در رشته مشخص می کند. اگر کاراکتر در رشته موجود نبود مقدار 1- را بر می گرداند.

char ampersand='a';

str.indexOf(ampersand);

موقعیت a را در رشته مشخص می کند.

مثال: برنامه ای بنویسید که مشخص کنید ایمیل موجود در رشته معتبر است یا خیر (بررسی کند آیا @ در رشته است یا خیر)

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

    char ampersand = '@';
    String dotCom = ".com";
    String email_address = "meme@me.com";

    int atPos = email_address.indexOf( ampersand );
    int result = email_address.indexOf(dotCom, atPos);

    if (result== -1 ) {
        System.out.println( "Invalid Email Address" );
    }
    else {
        System.out.println( "Email Address OK " + result );
    }
}
```

--زیر رشته یا substring جدا کردن قسمتی خاص از رشته، که با دو آرگومان تعداد و خانه شروع کار می کند.

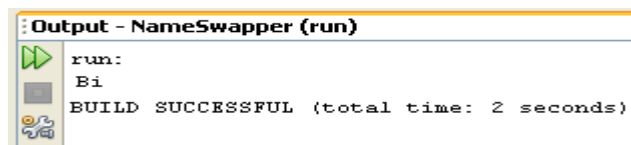
String FullName = "Bill Gates";
String FirstNameChars = "";

از خانه شماره صفر

2 حرف را جدا کن

FirstNameChars = FullName.substring(0, 2);

خروجی:



--برابری دو رشته: اگر مقدار تابع زیر true باشد دو رشته برابرند.

str1.equals(str2);

مثال: مقایسه کنید آیا دو رشته دلخواه برابر اند؟

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

    String email_address1 = "meme@me.cob";
    String email_address2 = "meme@me.com";
    Boolean isMatch = false;

    isMatch = email_address1.equals(email_address2);

    if (isMatch == true) {
        System.out.println( "Email Address Match ");
    }
    else {
        System.out.println("Email addresses don't match");
    }
}
```

--گرفتن یک char از رشته با استفاده از شماره خانه. مثال زیر کاراکتر خانه شماره 4 را می دهد

str.charAt(4);

مثال: با استفاده از اسکنر یک رشته را بخواند و بگوید حرف اول آن y است یا n؟

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

    Scanner user_input = new Scanner(System.in);

    System.out.println( "Quit Y/N");

    String aString = user_input.next();

    char aChar = aString.charAt(0);

    if (aChar == 'Y') {
        System.out.println( "OK, BYE BYE");
    }
    else {
        System.out.println( "Not Quitting");
    }
}
```

--عوض کردن یک کاراکتر در رشته، دستور زیر علامت # را در رشته به @ تبدیل می کند.

aString.replace('#', '@');

نکته: در برخی موارد در ابتدا یا انتهای رشته فاصله گذاشته می شود. برای حذف این فاصله از متد زیر استفاده می شود.

برای درک بهتر به مثال زیر توجه کنید.

`string password="mypass`

`";`

`password.Trim();`

فاصله را حذف می کند

خروجی: "mypass"

--متد در جاوا(تابع):

اسم تابع و نوع آن

آرگومان و نوع آن

`int total(int a)`

`{`

`int c=a+20;`

`return c;`

`}`

نکته: نوع تابع و نوع خروجی هر دو باید یکسان باشد. به طور مثال یک تابع از نوع عدد صحیح نمی تواند یک رشته را برگرداند.

نکته: استفاده از متد که در کلاس جدا گانه ای است در یک کلاس دیگر.

ابتدا باید متغیری از جنس کلاسی که متد در آن است را در کلاسی که می خواهیم از متد استفاده کنیم بسازیم.

اسم کلاسی که متد در آن است

اسم متغیر

=

new

اسم کلاسی که متد در آن است

`myclass test =new myclass();`

مثال: متدی برای جمع در یک کلاس نوشته و از آن در کلاس دیگر استفاده نماییم.

متد:

```
int total(int aNumber) {
    int a_Value = aNumber + 20;

    return a_Value;
}
```


کلاس:

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

    MyMethods test1 = new MyMethods();

    int aVal = test1.total();

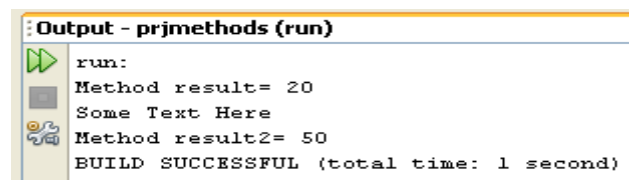
    System.out.println( "Method result= " + aVal );

    test1.print_text();

    int aVal2 = test1.total(30);

}
```

خروجی:



```
Output - prjmethods (run)
run:
Method result= 20
Some Text Here
Method result2= 50
BUILD SUCCESSFUL (total time: 1 second)
```

--تعریف یک کلاس

اسم کلاس **public class**

```
{
}
```

مثال: تعریف کلاس با چند متغیر؟

```
package exams;

public class StudentResults {

    private String Full_Name;
    private String Exam_Name;
    private String Exam_Score;
    private String Exam_Grade;

}
```

constructor چیست: تعریف یک متد درون کلاس با هدف استفاده از متغیرهای **private** را گویند.

زیرا از متغیرهای **private** فقط در محدوده همان کلاس میتوان استفاده کرد.

مثال: یک constructor تعریف کنید.

```
public class StudentResults {

    private String Full_Name;
    private String Exam_Name;
    private String Exam_Score;
    private String Exam_Grade;

    StudentResults( String name, String grade ) {

        Full_Name = name;
        Exam_Grade = grade;
    }
}
```

StudentResults در مثال فوق یک constructor است.

نکته: اگر تغییری به صورت public تعریف شود می توان به آن در کلاس دیگر دسترسی داشت. تنها کافی است تغییری از آن کلاس را در کلاس دیگر new کنیم. در کلاس بالا اگر Full_Name با Public باشد داریم.

```
StudentResults test = new StudentResults ();
```

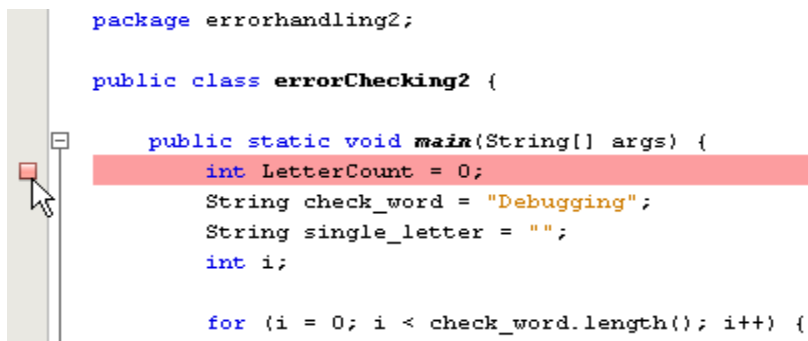
```
test.Full_Name;
```

--مدیریت خطا در جاوا

1- خود برنامه خطاهایی را که از نوع syntax است را تشخیص داده و برای اصلاح آن راه حل می دهد.

2- خطاهایی از نوع logic که با trace کردن می توان متوجه آنها شد.

نحوه trace در جاوا: ابتدا با breakpoint خط مورد نظر را مشخص کرده (کلیک بر روی خط مورد نظر)



```
package errorhandling2;

public class errorChecking2 {

    public static void main(String[] args) {
        int LetterCount = 0;
        String check_word = "Debugging";
        String single_letter = "";
        int i;

        for (i = 0; i < check_word.length(); i++) {
```

سپس debug کرده و با استفاده از F7 خط به خط برنامه را اجرا کنید.

خواندن از فایل:

ابتدا کتابخانه های زیر را اضافه می کنیم.

```
import java.io.IOException;
import java.io.FileReader;
import java.io.BufferedReader;
```

ابتدا آدرس فایل را در یک رشته ذخیره کرده

```
String file_name = "C:/test.txt";
```

سپس یک متغیر از نوع **ReadFile** تعریف کرده و آدرس فایل را که در رشته ریخته ایم را به عنوان آرگومان به آن می دهیم.

```
ReadFile file = new ReadFile( file_name );
```

فایل را باز کرده و در یک رشته ذخیره می کنیم.

```
String[ ] aryLines = file.OpenFile( );
```

فایل خوانده شد.

مثال: برنامه ای بنویسید که یک فایل را بخواند؟

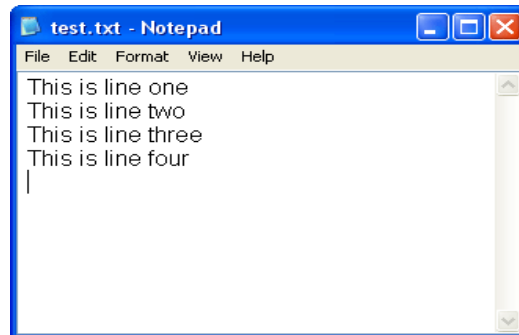
```
public static void main(String[] args) throws IOException {

    String file_name = "C:/test.txt";

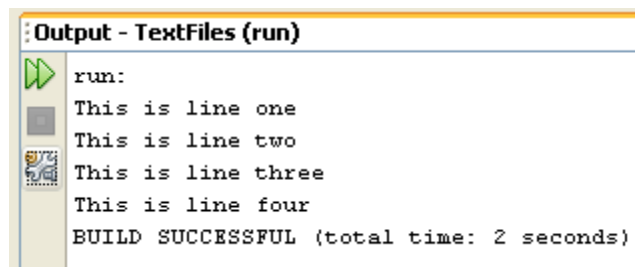
    try {
        ReadFile file = new ReadFile(file_name);
        String[] aryLines = file.OpenFile();

        int i;
        for (i=0; i < aryLines.length; i++) {
            System.out.println(aryLines[i]);
        }
    }
    catch (IOException e) {
        System.out.println( e.getMessage() );
    }
}
```

فایل:



خروجی:



نوشتن در یک فایل:

ابتدا کتابخانه های زیر را اضافه می کنیم.

```
import java.io.FileWriter;
import java.io.PrintWriter;
import java.io.IOException;
```

ابتدا آدرسی که قصد ذخیره فایل در آنجا را داریم می نویسیم

```
String path = file_path;
```

یک متغیر را از نوع **boolean** یا پرچم در نظر می گیریم و مقدار **true** را به آن می دهیم.

```
boolean f = true ;
```

سپس یک متغیر از نوع **WriteFile** تعریف می کنیم .

آدرس فایل و مقدار **true** را به عنوان آرگومان به متد می دهیم.

```
WriteFile data = new WriteFile(file_path , f );
```

رشته مورد نظر را به متد **writeToFile** می دهیم.

```
data.writeToFile( "This is another line of text" );
```

پایان بخش اول (در بخش دوم با محیط GUI و کار با database آشنا خواهید شد)