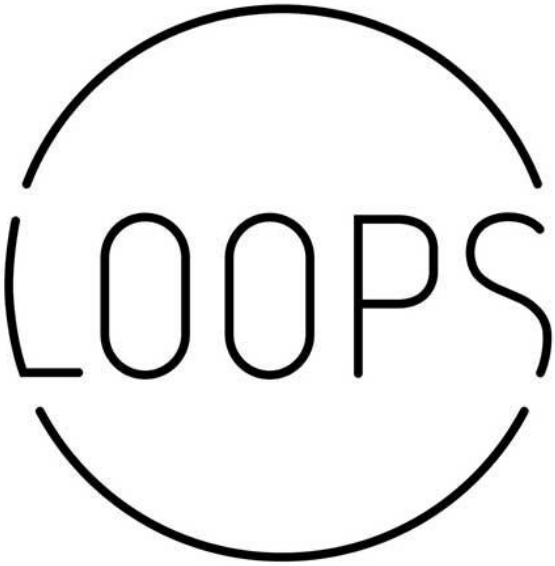
حلقه ها در جاوا



گاهی وقت ها در برنامه نویسی پیش می آید که شما نیاز دارید یک قسمت از کدها چندین بار اجرا بشوند. در حقیقت در زمان اجرا، کدهایی که نوشته اید بصروت خط به خط و پشت سر هم اجرا میشوند. اما در زبان جاوا ساختار هایی وجود دارد که میتوانید با استفاده از آنها کنترل های پیچیده تری روی روند اجرای برنامه داشته باشید. برای یادگیری حلقه ها در جاوا با **برنامه چی** همراه باشید.

در این جلسه سر فصل های زیر بررسی میشوند:

حلقه ها در جاوا چه کاری میکنند؟

چه حلقه هایی در جاوا وجود دارند؟

*حلقه while در جاوا

*حلقه for در جاوا

*حلقهdo while

*حلقه بهبود یافتهfor

**قواعد سینتکس

دستورات کنترل حلقه ها در جاوا

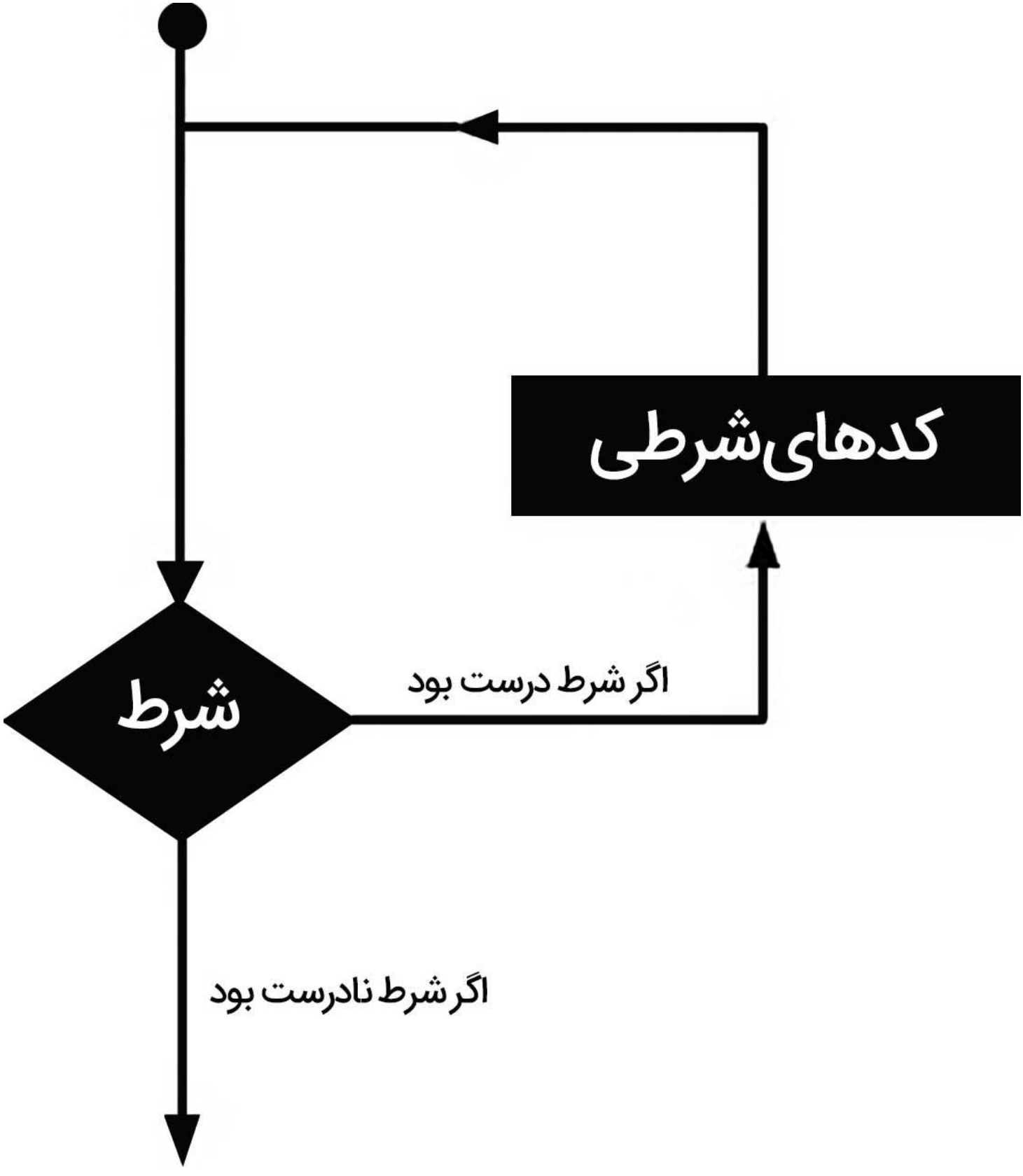
*دستور break در جاوا

*دستور continue در جاوا

دیدن این مقاله در سایت

حلقه ها در جاوا چه کاری میکنند؟

حلقه ها در جاوا به شما اجازه میدهند که یک دستور یا تعدادی از دستورات را چندبار پشت سر هم اجرا کنید. در زیر میتوانید فرم اصلی یک حلقه را مشاهده کنید که در اکثر زبان های برنامه نویسی هم به همین شکل است.



چه حلقه هایی در جاوا داریم؟

زبان برنامه نویسی جاوا انواع حلقه های زیر را شامل میشود که میتوانید از آنها برای کنترل برنامه خودتان استفاده کنید. برای آشنایی دقیقتر و جزئیات بیشتر میتوانید روی اسم حلقه ها کلیک کنید. **در زبان جاوا حلقه های زیر وجود دارند:**

ردیف	نام حلقه و توضیح کوتاه
1	**حلقه while در جاوا** حلقه while در جاوا یک سری دستورات مشخص را بصورت تکراری اجرا میکند و این روند را تا زمانی ادامه میدهد که شرطی که برای آن مشخص کرده ایم، true بشود.
2	**حلقه for در جاوا** حلقه for یک ساختار کنترل کننده تکرار شونده است که به شما تکرار میکند یک حلقه داشته باشید که به تعداد دقیق و مشخصی تکرار بشود. پس یعنی هنگامی که شما بدانید یک عمل خاص را دقیقا میخواهید دقیقا چندبار تکرار کنید، این نوع خاص از حلقه ها در جاوا ممکن است به دردتان بخورد.
3	**حلقه do while** حلقه do...while شبیه به حلقه while میباشد. تنها فرق آن این است که در این حلقه ضمانت میشود که حداقل برای یک مرحله، کدهای حلقه اجرا میشوند.

حلقه بهبود یافته for

بعد از انتشار نسخه 5 جاوا، حلقه بهبود یافته for معرفی شد. مورد استفاده اصلی آن برای پیمایش روی مجموعه ای از المان ها، مانند آرایه ها، میباشد.

قواعد سینتکس

در زیر نحوه نوشتن یک حلقه بهبود یافته for را میتوانید ببینید:

```
for(declaration : expression) {
  // Statements
}
```

- **declaration:** در این قسمت یک متغیر تعریف میکنید که درون بلوک حلقه for به کار برده بشود. پس یعنی داخل بلوک حلقه با این متغیر کار میکنیم. (اگر تا الان کدنویسی های بزرگ را تجربه نکرده باشید باید به شما بگویم که این کار برای جلوگیری از پیچیده شدن برنامه به شدت کاربردی است). متغیر جدیدی که در این بلوک کد تعریف میکنید، با آرایه ای که میخواهید با آن کار کنید، نوعی هماهنگی دارد. متغیر درون بلوک for قابل دسترس خواهد بود و مقدار آن دقیقا برابر با همان المنت آرایه است.
- **expression:** این قسمت درون آرایه ای که میخواهید حلقه آن را پیمایش کند، ذخیره میشود. این expression میتوانید یک متغیر آرایه ای باشد یا یک متد که یک آرایه را برمیگرداند. (به زبان ساده تر: یعنی فرض کنین یه آرایه دارین که میخواین یه سری مقدار ها رو توش بریزین. آرایه رو تو قسمت statement مینویسین و حلقه for روی تک تک خونه های آرایه یکی یکی جلو میره و مقداری که شما توی expression تعریف میکنید رو یکی یکی توی آرایه ای که میخواین ذخیره میکنه. به همین راحتی!).
- **Statement:** دستوراتی که میخواهید درون حلقه اجرا بشوند را اینجا مینویسید.

مثال

```
public class Test {
  public static void main(String args[]) {
    int [] numbers = {10, 20, 30, 40, 50};

    for(int x : numbers ) {
      System.out.print( x );
      System.out.print(",");
    }

    System.out.print("\n");

    String [] names = {"James", "Larry", "Tom", "Lacy"};

    for( String name : names ) {
      System.out.print( name );
      System.out.print(",");
    }
  }
}
```

کدهای بالا خروجی زیر را تولید میکنند.

خروجی

```
۱۰, ۲۰, ۳۰, ۴۰, ۵۰,
James, Larry, Tom, Lacy,
```

همانطوری که در مثال بالا میبینید، داخل بلوک for ما یک متغیر به نام x را تعریف کرده ایم و یک آرایه از جنس int به نام number هم وجود دارد که مقادیر آن یکی یکی درون x قرار میگیرند. حالا درون بلوک for تعریف کرده ایم که مقدار درون x روی صفحه نمایش نشان داده بشود. همانطور که در خروجی مشاهده میکنید هربار که یک مرتبه از حلقه اجرا میشود، یکی یکی مقادیر آرایه number درون متغیر x قرار میگیرند و کارهایی که میخواهید روی متغیر x، درون بلوک for اجرا میشوند.

دستورات کنترل حلقه ها در جاوا

دستورات کنترل حلقه ها در جاوا مسیر نرمال اجرای حلقه ها را به هم میریزد. وقتی که روند اجرا از محدوده حلقه خارج میشود، همه عمل های اتوماتیک که درون آن تعریف شده بودند نیز نابود میشوند. زبان جاوا از دستورات زیر برای کنترل حلقه ها پشتیبانی میکند. میتوانید برای دیدن جزییات بیشتر روی اسم هرکدام از آنها کلیک کنید.

ردیف	نام دستور و توضیح کوتاه
1	**دستور break در جاوا** دستور break برای حلقه ها در جاوا دو کاربرد زیر را دارند: • وقتی که روند اجرای دستورات حلقه به دستور break میرسد، بلافاصله حلقه از بین میرود و ادامه دستوراتی که بعد از حلقه نوشته شده اند اجرا خواهند شد. • در دستور switch هم میتوانید break را برای از بین بردن اجرای دستورات در قسمت case ها به کار ببرید. (در درس بعدی درباره این ساختار صحبت خواهیم کرد).
2	**دستور continue در جاوا** • دستور continue میتواند در همه حلقه ها در جاوا استفاده بشود. Continue باعث میشود که اجرای دستورت به خط بعدی خود حلقه پرش بکند. • در حلقه های for کلمه کلیدی continue باعث میشود حلقه بلافاصله به مرحله update منتقل بشود. • در حلقه while یا حلقه do while کنترل حلقه مستقیما به Boolean expression پرش بکند.

در جلسه بعد چه چیزی یاد میگیریم؟

در قسمت بعدی از دوره آموزش جاوا درباره دستورات تصمیم گیری در زبان برنامه نویسی جاوا بحث میکنیم. با ما همراه باشید.

جلسه قبل: عملگرهای جاوا

جلسه بعد: تصمیم سازی در جاوا