دستور continue در جاوا

در این آموزش یاد می گیرید که از دستور continue در جاوا استفاده کنید. دستور continue از تکرار فعلی حلقه جلوگیری می کند.

فرض کنید در حال کار با حلقه ها هستید. گاهی اوقات می خواهید از برخی دستورات داخل حلقه پرش کنید یا حلقه را فورا بدون چک کردن شرط خاتمه دهید.

در چنین مواردی از عبارتهای break و continue استفاده می شود.

دستور continue از تکرار فعلی یک حلقه (for) ، while، و (do … while جلوگیری می کند.

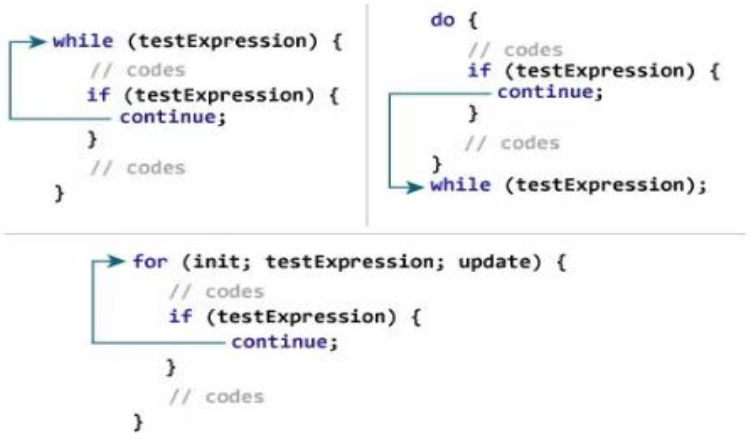
هنگامی که دستور continue اجرا شد ، کنترل برنامه به انتهای حلقه می پرد. سپس شرطی که حلقه را کنترل می کند ارزیابی می شود. در مورد حلقهfor ، قبل از ارزیابی شرط ، عبارت به روزرسانی متغیر اجرا می شود.

اکثر اوقات با ساختار های تصمیم گیری استفاده می شود.(if … else)

نحو دستور continue به شکل زیر است:

```
continue;
```

دستور continue به چه شکل کار می کند؟



مثال ۱ : دستور continue در جاوا

```
1.  class Test {
2.  public static void main(String[] args) {
3.
4.  for (int i = 1; i <= 10; ++i) {
5.  if (i > 4 && i < 9) {
6.  continue;
7.  }
8.  System.out.println(i);
9.  }
10. }
11. }
```

وقتی مقدار i بیشتر از ۴ شود و کمتر از ۹ شود ، دستور continue اجرا می شود و از اجرای System.out.println (i)جلوگیری می کند.

هنگام اجرای برنامه ، خروجی برابر خواهد بود با:

```
۱
۲
۳
۴
۹
۱۰
```

مثال ۲ : دستور continue در جاوا

برنامه زیر مجموع حداکثر ۵ عدد مثبت که توسط کاربر وارد شده است را محاسبه می کند. اگر کاربر عدد منفی یا صفر وارد کند ، از محاسبه رد می شود.

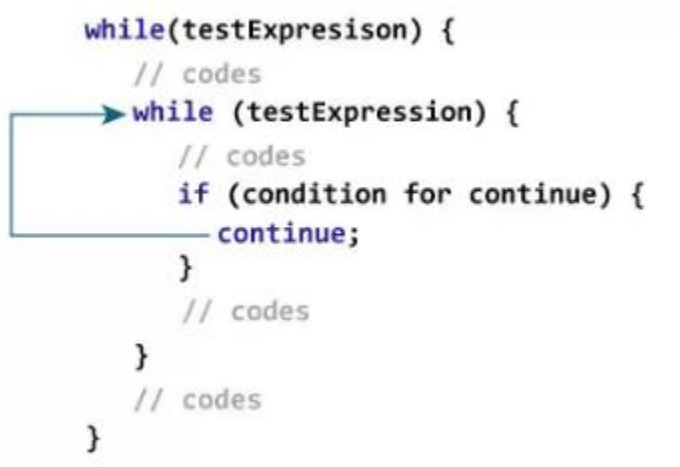
برای گرفتن ورودی از کاربر ، از شی Scanner استفاده می شود.

```
1.  import java.util.Scanner;
2.  class AssignmentOperator {
3.  public static void main(String[] args) {
4.
5.  Double number, sum = 0.0;
6.  Scanner input = new Scanner(System.in);
7.
8.  for (int i = 1; i < 6; ++i) {
9.  System.out.print(“Enter a number: “);
10. number = input.nextDouble();
11.
12. if (number <= 0.0) {
13. continue;
14. }
15.
16. sum += number;
17. }
18. System.out.println(“Sum = ” + sum);
19. }
20. }
```

خروجی:

```
Enter a number: 2.2
Enter a number: 5.6
Enter a number: 0
Enter a number: -2.4
Enter a number: -3
Sum = 7.8
```

در صورت وجود حلقه های تو در تو ، continueبه ابتدای حلقه درونی می پرد.

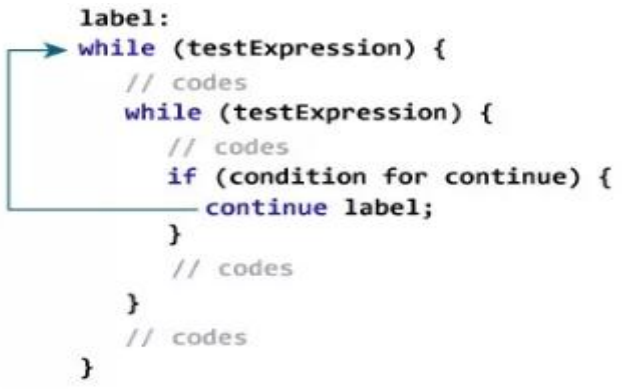


دستور continue برچسب دار

دستور continue که تاکنون در موردش صحبت کردیم، بدون برچسب است، که از اجرای دستورات باقی مانده درونی ترین حلقه for ، whileو while ... do جلوگیری می کند.

شکل دیگری از دستور continue ، فرم دارای برچسب آن است، که می تواند برای پرش از اجرای دستورات که درون حلقه بیرونی قرار دارند ، استفاده شود.

دستور continue برچسب دار چگونه کار می کند؟



در اینجا label شناسه است.

مثال ۳ : دستور continue برچسب دار

```
1.  class LabeledContinue {
2.  public static void main(String[] args) {
3.
4.  label:
5.  for (int i = 1; i < 6; ++i) {
6.  for (int j = 1; j < 5; ++j) {
7.  if (i == 3 || j == 2)
8.  continue label;
9.  System.out.println("i = " + i + "; j = " + j);
10. }
11. }
12. }
13. }
```

خروجی

```
i = 1; j = 1
i = 2; j = 1
i = 4; j = 1
i = 5; j = 1
```

معمولا از استفاده continue برچسب دار صرف نظر می شود زیرا درک کد را سخت می کند. اگر مجبور به استفاده از continue برچسب دار هستید ، کد خود را از نو پالایش کنید و سعی کنید آن را به روشی متفاوت بنویسید تا خوانا تر شود.