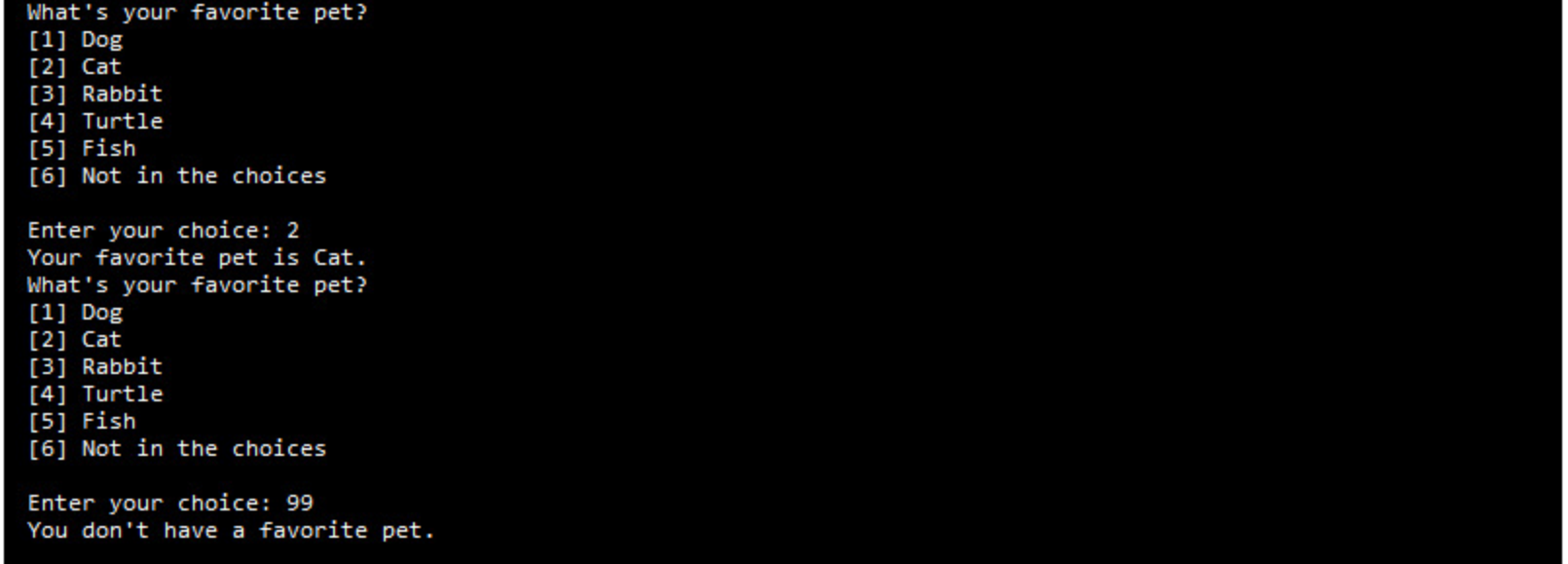


در جاوا ساختاری به نام switch وجود دارد که به شما اجازه می‌دهد که با توجه به مقدار ثابت یک متغیر چندین انتخاب داشته باشید. دستور switch معادل دستور if تو در تو است با این تفاوت که در دستور switch متغیر فقط مقادیر ثابتی از اعداد، رشته‌ها و یا کاراکترها را قبول می‌کند. مقادیر ثابت مقادیری هستند که قابل تغییر نیستند. در زیر نحوه استفاده از دستور switch آمده است :

```
switch (testVar)
{
    case compareVal1:
        code to execute if testVar == compareVal1;
        break;
    case compareVal2:
        code to execute if testVar == compareVal2;
        break;
    .
    .
    .
    case compareValN:
        code to execute if testVer == compareValN;
        break;
    default:
        code to execute if none of the values above match the testVar;
        break;
}
```

ابتدا یک مقدار در متغیر switch که در مثال بالا testVar است قرار می‌دهید. این مقدار با هر یک از عبارتهای case داخل بلوک switch مقایسه می‌شود. اگر مقدار متغیر با هر یک از مقادیر موجود در دستورات case برابر بود کد مربوط به آن case اجرا خواهد شد. به این نکته توجه کنید که حتی اگر تعداد پط کدهای داخل دستور case از یکی بیشتر باشد نباید از آکولاد استفاده کنیم. آخر هر دستور case با کلمه کلیدی break تشخیص داده می‌شود که باعث می‌شود برنامه از دستور switch خارج شده و دستورات بعد از آن اجرا شوند. اگر این کلمه کلیدی از قلم بیوفتد برنامه با خطا مواجه می‌شود. دستور switch یک بخش default دارد. این دستور در صورتی اجرا می‌شود که مقدار متغیر با هیچ یک از مقادیر دستورات case برابر نباشد. دستور default اختیاری است و اگر از بدنه switch حذف شود هیچ اتفاقی نمی‌افتد. مکان این دستور هم مهم نیست اما بر طبق تعریف آن را در پایان دستورات می‌نویسند. به مثالی در مورد دستور switch توجه کنید :

```
1 package myfirstprogram;
2
3 import java.util.Scanner;
4
5 public class MyFirstProgram
6 {
7     public static void main(String[] args)
8     {
9         Scanner input = new Scanner(System.in);
10
11         int choice;
12
13         System.out.println("What's your favorite pet?");
14         System.out.println("[1] Dog");
15         System.out.println("[2] Cat");
16         System.out.println("[3] Rabbit");
17         System.out.println("[4] Turtle");
18         System.out.println("[5] Fish");
19         System.out.println("[6] Not in the choices");
20         System.out.print("Enter your choice: ");
21
22         choice = input.nextInt();
23
24         switch (choice)
25         {
26             case 1:
27                 System.out.println("Your favorite pet is Dog.");
28                 break;
29             case 2:
30                 System.out.println("Your favorite pet is Cat.");
31                 break;
32             case 3:
33                 System.out.println("Your favorite pet is Rabbit.");
34                 break;
35             case 4:
36                 System.out.println("Your favorite pet is Turtle.");
37                 break;
38             case 5:
39                 System.out.println("Your favorite pet is Fish.");
40                 break;
41             case 6:
42                 System.out.println("Your favorite pet is not in the choices.");
43                 break;
44             default:
45                 System.out.println("You don't have a favorite pet.");
46                 break;
47         }
48     }
49 }
```



برنامه بالا به شما اجازه انتخاب حیوان مورد علاقه‌تان را می‌دهد. به اسم هر حیوان یک عدد نسبت داده شده است. شما عدد را وارد می‌کنید و این عدد در دستور switch با مقادیر case مقایسه می‌شود و با هر کدام از آن مقادیر که برابر بود پیغام مناسب نمایش داده خواهد شد. اگر هم با هیچ کدام از مقادیر case ها برابر نبود دستور default اجرا می‌شود. یکی دیگر از ویژگیهای دستور switch این است که شما می‌توانید از دو یا چند case برای نشان داده یک مجموعه کد استفاده کنید. در مثال زیر اگر مقدار number، 1، 2 یا 3 باشد یک کد اجرا می‌شود. توجه کنید که case ها باید پشت سر هم نوشته شوند.

```
switch(number)
{
    case 1:
    case 2:
    case 3:
        System.out.println("This code is shared by three values.");
        break;
}
```

همانطور که قبلاً ذکر شد دستور switch معادل دستور if تو در تو است. برنامه بالا را به صورت زیر نیز می‌توان نوشت :

```
if (choice == 1)
    System.out.println("Your favorite pet is Dog.");
else if (choice == 2)
    System.out.println("Your favorite pet is Cat.");
else if (choice == 3)
    System.out.println("Your favorite pet is Rabbit.");
else if (choice == 4)
    System.out.println("Your favorite pet is Turtle.");
else if (choice == 5)
    System.out.println("Your favorite pet is Fish.");
else if (choice == 6)
    System.out.println("Your favorite pet is not in the choices.");
else
    System.out.println("You don't have a favorite pet.");
```

کد بالا دقیقاً نتیجه‌ای مانند دستور switch دارد. دستور default معادل دستور else می‌باشد. حال از بین این دو دستور (switch و if else) کدامیک را انتخاب کنیم. از دستور switch موقعی استفاده می‌کنیم که مقداری که می‌خواهیم با دیگر مقادیر مقایسه شود ثابت باشد. مثلاً در مثال زیر هیچگاه از switch استفاده نکنید.

```
int myNumber = 5;
int x = 5;

switch (myNumber)
{
    case x:
        System.out.println("Error, you can't use variables as a value" +
                             " to be compared in a case statment.");
        break;
}
```

مشاهده می‌کنید که با اینکه مقدار x عدد 5 است و به طور واضح با متغیر myNumber مقایسه شده است برنامه خطا می‌دهد چون x یک ثابت نیست بلکه یک متغیر است یا به زبان ساده‌تر، قابلیت تغییر را دارد. اگر بخواهید از x استفاده کنید و برنامه خطا ندهد باید از کلمه کلیدی final به صورت زیر استفاده کنید.

```
int myNumber = 5;
final int x = 5;

switch (myNumber)
{
    case x:
        System.out.println("Error has been fixed!");
        break;
}
```

از کلمه کلیدی final برای ایجاد ثابتها استفاده می‌شود. توجه کنید که بعد از تعریف یک ثابت نمی‌توان مقدار آن را در طول برنامه تغییر داد. به یاد داشته باشید که باید ثابتها را حتماً مقداردهی کنید. دستور switch یک مقدار را با مقادیر case ها مقایسه می‌کند و شما لازم نیست که به شکل زیر مقادیر را با هم مقایسه کنید :

```
switch (myNumber)
{
    case x > myNumber:
        System.out.println("switch staments can't test if a value is less than " +
                             "or greater than the other value.");
        break;
}
```