

به نام خدا

قسمت هفدهم

همانطور که در جلسات گذشته دیدیم، چنانچه داده ورودی به برنامه با متدی که در کلاس Scanner به کار گرفته شده بود هم خوانی نداشت برنامه با خطا مواجه می شد و این در حالی است که ما این اجازه را نداریم تا برنامه ای طراحی کنیم که اگر کاربر چیزی به اشتباه وارد برنامه کرد برنامه در وسط اجرا برنامه Crash کند یا از ادامه کار باز ایستد.

برای این منظور نیاز داریم تا از متدهایی در کلاس Scanner استفاده کنیم که از نوع boolean باشند که چنانچه کاربر یک داده اشتباه وارد کرد به او بازخورد دهد. بنابراین یک پروژه جدید تحت عنوان 17th Session به معنی "جلسه هفدهم" ایجاد کرده و کلاسی به اسم Project به معنی "پروژه" در آن ایجاد می کنیم.

اکنون نیاز است تا یک سناریو تعریف کنیم تا بر آن اساس پروژه خود را تکمیل کنیم: فرض کنیم می خواهیم یک برنامه تحت وب طراحی کنیم که کاربران به منظور استفاده از این برنامه می بایست در ابتدا رمز خود را وارد کنند. کاری که ما قرار است در این برنامه انجام دهیم این است که شرطی بگذاریم که اگر کاربر هر داده ای به غیر عدد صحیح وارد کرد برنامه از وی ایراد بگیرد چرا که رمزهای این برنامه فقط بر اساس یک عدد صحیح چند رقمی طراحی شده اند در ضمن اگر کاربر رمز خود را به اشتباه وارد کرد، برنامه از وی بخواهد که مجدداً اطلاعات خود را وارد کند.

در ابتدا یک شیء از روی کلاس Scanner تحت عنوان passwordInput به معنی "ورودی رمز عبور" می سازیم. سپس می خواهیم جمله ای به این مضمون روی صفحه مانیتور نمایش دهیم که "لطفا رمز عبور خود را وارد کنید". برای این منظور کد خود را به شکل زیر تکمیل می کنیم:

```
import java.util.Scanner;

class Project {
    public static void main(String[] args) {
        Scanner passwordInput = new Scanner(System.in);
        System.out.print("Please enter your password: ");
    }
}
```

اکنون پس از آنکه کاربر رمز عبور خود را وارد برنامه کرد، نیاز است تا اولاً چک کنیم ببینیم که آیا داده ورودی از جنس یک عدد صحیح است یا خیر ثانیاً چک کنیم ببینیم که آیا رمز عبور درست وارد شده است یا خیر. متدی که در اینجا مورد استفاده قرار خواهد گرفت متدی از جنس boolean است. به عبارت دیگر متدهایی که از این جنس هستند صرفاً دو خروجی خواهند داشت به این صورت که اگر شرطی که این متد گذاشته برآورده شود جواب true خواهد داشت و در غیر این صورت جواب false خواهیم گرفت.

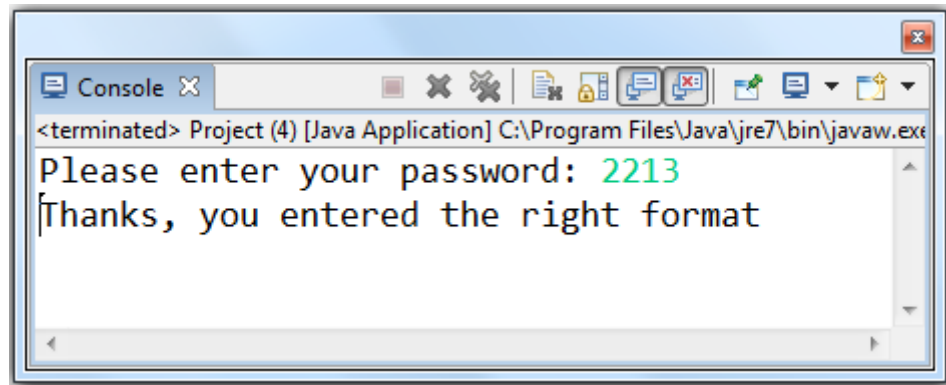
متدهای boolean که در ارتباط با کلاس Scanner مورد استفاده قرار می گیرند و از اهمیت بیشتری برخوردارند عبارتند از hasNextLine() برای بررسی این مسئله که ببینیم آیا داده ای از جنس یک string یا "متن" وارد شده است یا خیر، hasNextDouble() برای بررسی این مسئله که ببینیم آیا داده ای از جنس یک عدد اعشاری وارد شده است یا خیر و در نهایت hasNextInt() برای بررسی این مسئله که ببینیم آیا داده ای از جنس یک عدد صحیح وارد شده است یا خیر که در برنامه خود در این پروژه از متد آخر استفاده خواهیم کرد چرا که در سناریوی این پروژه تعریف کرده ایم که رمز برنامه می بایست از داده ای از جنس عدد صحیح باشد.

بنابراین می بایست کد خود را به شکل زیر تکمیل کنیم:

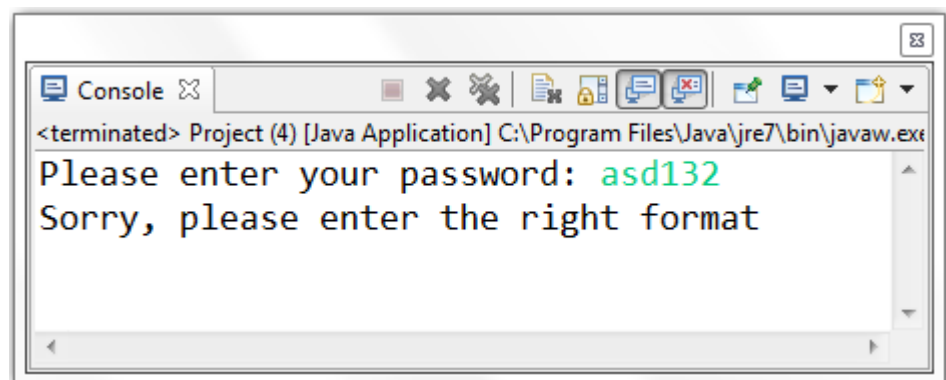
```
import java.util.Scanner;

class Project {
    public static void main(String[] args) {
        Scanner passwordInput = new Scanner(System.in);
        System.out.print("Please enter your password: ");
        if (passwordInput.hasNextInt()) {
            System.out.println("Thanks, you entered the right format");
        } else {
            System.out.println("Sorry, please enter the right format");
        }
    }
}
```

شرط اول برنامه این است که چک کنیم ببینیم که آیا کاربر داده ای از جنس عدد صحیح به عنوان رمز وارد کرده است یا خیر. برای این منظور همانطور که در کد فوق مشاهده می شود یک شرط if وارد برنامه کرد و داخل پرانتز نام شیئی ساخته شده از روی کلاس Scanner را نوشته سپس متد hasNextInt() را به آن ضمیمه می کنیم. به طور خلاصه این شرط حاکی از آن است که برنامه ببیند آیا داده ورودی از طریق کیبورد از جنس یک عدد صحیح است یا خیر و این در حالی است که هیچ فرقی نمی کند این عدد صحیح چه مقداری داشته باشد چرا که در شرط دوم خود این مسئله را خواهیم سنجید. از آنجا که می خواهیم اگر این شرط true بود عبارت Thanks, you entered the right format به معنی "با تشکر، شما فرمت صحیح را وارد کردید" روی صفحه مانیتور به نمایش درآید، پس جمله خود را در دستور System.out.println(); در داخل { } مقابل if می نویسیم. از آنجا که می خواهیم اگر فرمت رمز چیزی به غیر از داده ای از جنس عدد صحیح بود برنامه به کاربر اخطار دهد، داخل دستور else با استفاده از دستور System.out.println(); جمله Sorry, please enter the right format به معنی "متأسفیم، لطفا فرمت صحیح را وارد نمایید" را وارد می کنیم. اکنون یک بار برنامه خود را اجرا می کنیم و رمزی از جنس یک عدد صحیح مثلاً عدد ۲۲۱۳ وارد می کنیم:



می بینیم که برنامه داده ورودی را به درستی تشخیص می دهد. مجدد برنامه را اجرا کرده این بار هر نوع داده ای به غیر از یک عدد صحیح وارد می کنیم مثلاً asd132:



همانطور که ملاحظه می شود از آنجا که جواب شرط false است برنامه دستور داخل else را اجرا می کند. در ادامه برنامه می خواهیم ببینیم که اگر رمز داده ای از جنس عدد صحیح بود، حال آیا این رمز با رمز تعبیه شده در برنامه هم خوانی دارد یا خیر. برای این منظور کد خود را به صورت زیر بازنویسی می کنیم:

```

import java.util.Scanner;

class Project {
    public static void main(String[] args) {
        Scanner passwordInput = new Scanner(System.in);
        System.out.print("Please enter your password: ");

        int builtInPassword = 2244176;

        if (passwordInput.hasNextInt()
            && passwordInput.nextInt() == builtInPassword) {
            System.out.println("Thanks, you entered the right format"
                               + " \nand also your password is correct");
        } else {
            System.out.println("Sorry, please enter the right format"
                               + " \nor the right password");
        }
    }
}

```

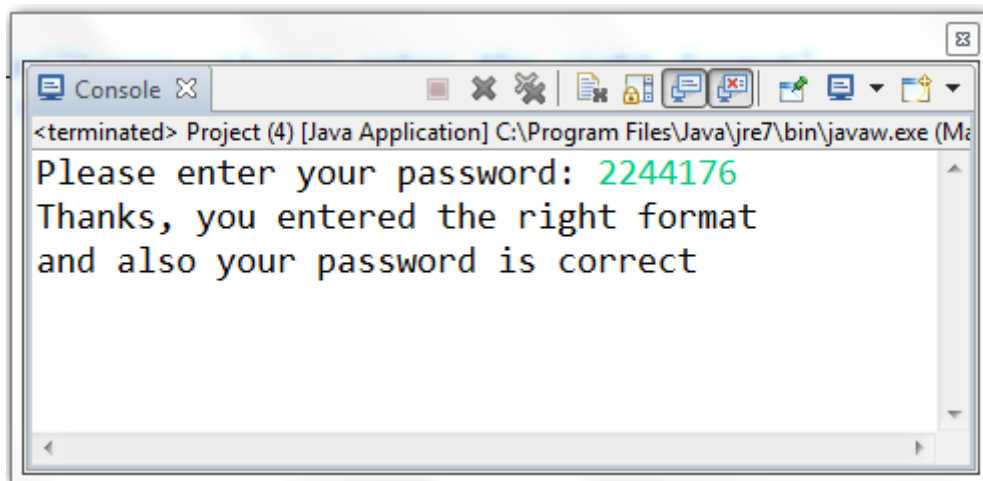
همانطور که مشاهده می شود یک متغیر از جنس int تحت عنوان builtInPassword به معنی رمز عبور تعبیه شده در سیستم ایجاد می کنیم و مقدار آن را معادل با 2244176 قرار می دهیم (این عدد به دلخواه انتخاب شده است). به طور خلاصه اگر کاربر داده ای از جنس عدد صحیح از یک سو، و همچنین عدد صحیحی معادل با 2244176 از سوی دیگر وارد کند خواهد توانست وارد حساب کاربری خود شود. به طور خلاصه، داخل شرط if یک علامت && قرار داده ایم که در زبان برنامه نویسی جاوا به معنی "و" می باشد. در واقع هر موقع که بخواهیم بیش از یک شرط را در برنامه خود بگنجانیم از این ساختار استفاده می کنیم (به خاطر داشته باشیم که در شرط قرار گرفته در سمت چپ && متد hasNextInt به شیئی ساخته شده از روی کلاس Scanner ضمیمه شده در حالیکه در شرط قرار گرفته در سمت راست علامت && متد nexInt به این شیئی ضمیمه شده است). سپس این شرط را می گذاریم که اگر شیئی ساخته شده از روی کلاس Scanner که همان passwordInput است به انضمام متد nextInt() که وظیفه دارد مقدار متغیر از جنس عدد صحیح را در خود ذخیره سازد دارای مقداری برابر با مقدار متغیر builtInPassword بود، عبارت Thanks, you entered the right format and also your password is correct به معنی "با تشکر، شما فرمت صحیح را وارد

دوره آموزش جاوا

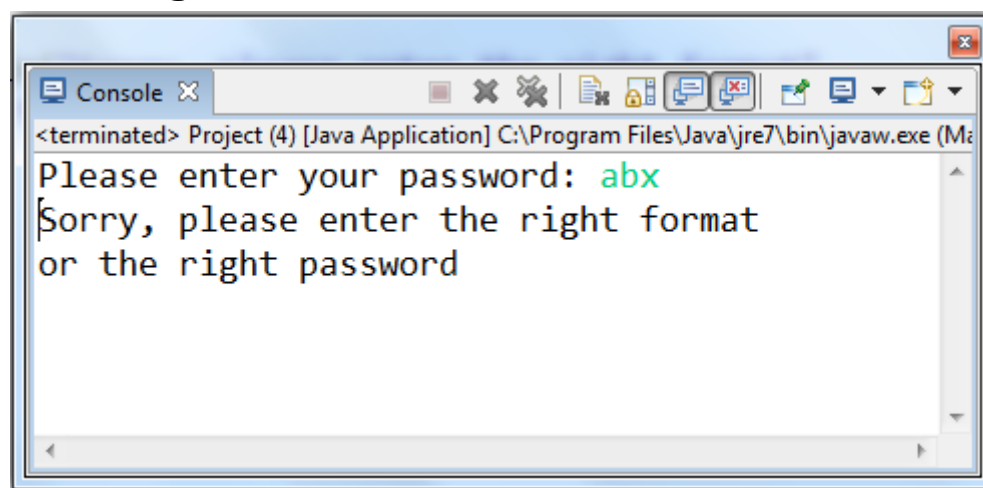
کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

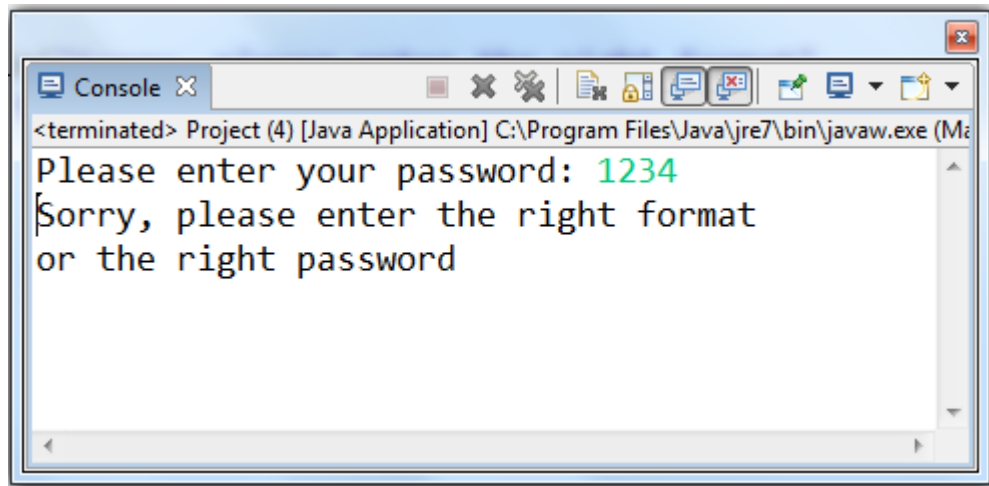
کردید و رمز عبور هم صحیح است" نمایش داده خواهد شد و در صورتیکه جواب به این دو شرط یا یکی از آنها false بود، عبارت Sorry, please enter the right format or the right password به معنی "متأسفیم، لطفا فرمت صحیح یا رمز عبور صحیح را وارد نمایید" نمایش داده خواهد شد. اکنون می توانیم برنامه خود را اجرا کنیم و نتیجه را ببینیم:



از آنجا که داده ورودی از یک سو یک عدد صحیح است و از سوی دیگر معادل با مقدار متغیر builtInPassword است، جواب هر دو شرط true بوده و دستور مرتبط با آن اجرا می شود. مجدداً برنامه را اجرا کرده ولی این بار داده ای به غیر از عدد صحیح وارد می کنیم:



می بینیم از آنجا که داده ورودی یک عدد صحیح نیست از این رو جواب به شرط passwordInput.hasNextInt() معادل با false خواهد بود و دستور مرتبط با false اجرا می شود. مجدداً برای آخرین بار برنامه را اجرا کرده و این بار یک عدد صحیح وارد می کنیم اما مقدار این عدد صحیح با مقدار متغیر builtInPassword یکی نیست:



از آنجا که در مثال فوق فقط یکی از شرط های برنامه برآورده شده است، به عبارت دیگر جواب به شرط `passwordInput.nextInt() == builtInPassword` با `false` است، پس در کل جواب به شرط `false` بوده و متعاقباً دستور مرتبط با جواب `false` اجرا می شود.

در آموزش هجدهم یا دستورات شرطی `switch` در قالب پروژه ای تحت عنوان "ماه تولد" آشنا خواهیم شد.