# پودمان پنجم

## برنامه‌نویسی اندروید

شاید خود اندی رابین (خالق اندروید / مدیر کنونی تیم توسعه اندروید در شرکت گوگل) هم زمانی که داشت در شرکت کوچک اندروید روی تولید نرم‌افزار برای گوشی‌های موبایل کار می‌کرد، فکرش را نمی‌کرد که روزی شرکت گوگل، غول بزرگ دنیای فناوری و اطلاعات آن را بخرد و بعد آن اندروید به اوج برسد.

# واحد یادگیری
# شایستگی برنامه‌نویسی اندروید

آیا می‌دانید

ـ برنامه‌نویسی کجا کاربرد دارد؟
ـ تفاوت زبان‌های برنامه‌نویسی را می‌دانید؟
ـ با دستورات زبان برنامه‌نویسی جاوا مثل (if-for) آشنایی دارید؟
ـ کاربرد آرایه در برنامه‌نویسی چیست؟
ـ برنامه‌نویسی اندروید را چگونه آغاز کنیم؟
ـ اتصال یک اپلیکیشن با وسایل سخت‌افزاری چگونه می‌باشد؟

**هدف از این شایستگی عبارت‌اند از:**

- آشنایی با برنامه نویسی Java
- بررسی انواع عملکردهای ریاضی ـ مقایسه‌ای ـ منطقی
- معرفی ساختارهای کنترلی در جاوا
- معرفی ساختارهای تکرار در جاوا
- آشنایی با آرایه
- معرفی برنامه نویسی اندروید

## استاندارد عملکرد

پس از اتمام واحد یادگیری و کسب شایستگی برنامه‌نویسی اندروید، هنرجویان قادر خواهند بود برنامه‌های مختلفی را به زبان برنامه‌نویسی جاوا نوشته و همچنین اپلیکیشن‌های جدید تولید کنند.

## مقدمه

این روزها تلفن همراه را می‌توان دست همه آدم‌ها دید. دیگر کسی نیست که در کاربردی بودن آنها شک داشته باشد و استفاده از آنها هر روز فراگیرتر می‌شود. همه کسب و کارها به سمت آنلاین شدن و رایانه‌ای شدن در حرکت هستند و موبایل‌ها و تبلت‌ها امروزه تقریباً از عهدهٔ هر کاری برمی‌آیند.
یکی از ضرورت‌های اصلی و مهارت‌های با ارزش در حال حاضر توانایی برنامه‌نویسی برای دستگاه‌های تلفن همراه است. اگر با اصول اولیه برنامه‌نویسی موبایل آشنا باشید و از آن در کنار سایر مهارت‌هایی که در طول تحصیل به‌دست می‌آورید استفاده کنید، می‌توانید کارهای بسیار جالب و مفیدی انجام بدهید و در آینده شغلی‌تان موفق‌تر باشید.
در این فصل می‌خواهیم با برنامه‌نویسی تلفن همراه آشنا بشویم و یک برنامه تلفن همراه برای کنترل تخت بیمارستان بنویسیم که از راه دور و با کمک بلوتوث بتوان تخت بیمارستان را کنترل کرد.

## سیستم عامل

سیستم عامل برنامه‌ای است که ارتباط بین سخت‌افزار و برنامه‌های کاربردی را برقرار می‌کند. هر وسیله محاسباتی مثل رایانه و گوشی‌های تلفن همراه سیستم عامل دارند. معروف‌ترین و پر استفاده‌ترین سیستم عامل‌های دنیا **ویندوز**، **اندروید** و **لینوکس** هستند. هر کدام از سیستم عامل‌ها برای نوع خاصی از دستگاه‌ها مناسب هستند. معمولاً از ویندوز برای رایانه‌های شخصی، از اندروید برای گوشی‌های موبایل و سایر دستگاه‌های مصرفی و از لینوکس برای رایانه‌های سرور استفاده می‌شود.

## اندروید

اندروید یک سیستم عامل منبع باز و رایگان مبتنی بر هسته لینوکس است که به‌گونه‌ای نوشته شده است که بر روی دستگاه‌های مصرفی با سخت‌افزار محدود و توان پردازش کم به‌خوبی کار کند.


پخش فیلم

منبع باز چیست؟

ویژگی «**منبع باز بودن**» اندروید باعث شده است تا شرکت‌های گوناگون آن را بر روی دستگاه‌های خود نصب کنند و کاربران زیادی در سرتاسر دنیا از دستگاه‌های مجهز به اندروید استفاده کنند. در حال حاضر حدود ۲ میلیارد گوشی موبایل از سیستم عامل اندروید استفاده می‌کنند که تقریباً ۸۵ درصد گوشی‌های موبایل هوشمند در دنیا است.

پخش فیلم

تاریخچه ساخت سیستم عامل اندروید

## برنامه‌نویسی

دلیلی که ما از رایانه‌ها، تلفن همراه‌ها و سایر ابزارهای محاسباتی استفاده می‌کنیم این است که این دستگاه‌ها در انجام محاسبات پیچیده مورد نیاز برای انجام کارهای مختلف به کمک ما بیایند. دستگاه‌های سخت‌افزاری از تعدادی مدارهای الکترونیک ساخته شده‌اند که قادر به درک مفاهیم و زبان انسان نیستند. به همین دلیل ما به کمک زبان‌های برنامه‌نویسی، دستورها و فرمان‌های محاسباتی را به زبان قابل فهم این دستگاه‌ها تبدیل می‌کنیم. به این کار برنامه‌نویسی می‌گویند.

پخش فیلم

انواع شیوه‌های برنامه‌نویسی

در حال حاضر زبان‌های برنامه‌نویسی متعددی وجود دارند که هر کدام از آنها برای نوع خاصی از محاسبات مناسب هستند. با این حال برخی زبان‌های برنامه‌نویسی، زبان‌های همه‌منظوره هستند و از آنها می‌توان به‌صورت عام برای هر نوع برنامه‌نویسی استفاده کرد. یکی از زبان‌های مطرح و بسیار پرکاربرد زبان برنامه‌نویسی جاوا است.

## زبان برنامه‌نویسی جاوا


جاوا یکی از معروف‌ترین و پرکاربردترین زبان‌های برنامه‌نویسی همه‌منظوره است. به‌دلیل پرکاربرد بودن و آشنا بودن برنامه‌نویسان زیادی در سرتاسر دنیا به این زبان برنامه‌نویسی، اندروید هم این زبان برنامه‌نویسی را به‌عنوان زبان خود برگزیده است و متداول‌ترین شیوه برنامه‌نویسی برای اندروید، استفاده از زبان جاوا است.

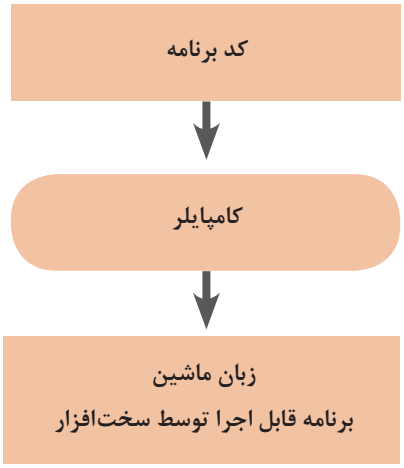
جاوا **زبانی شئ‌گرا** و **مستقل از محیط اجرا** است. برنامه‌های نوشته شده به زبان جاوا را می‌توان روی همه سیستم‌عامل‌های متداول و شناخته شده اجرا کرد.

پخش فیلم

تاریخچه زبان جاوا

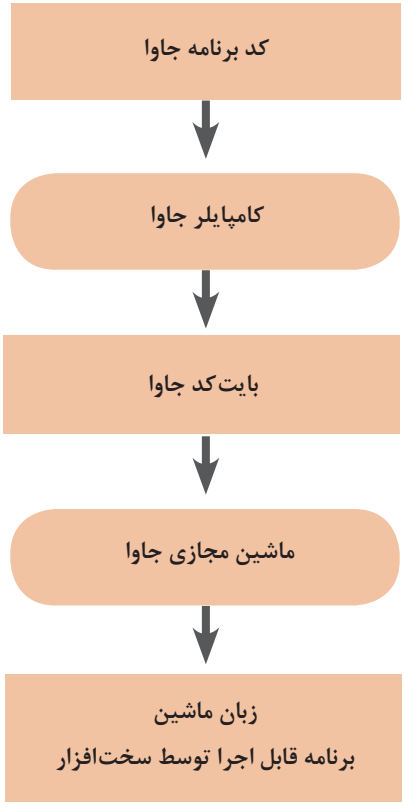
## مراحل برنامه‌نویسی جاوا

به‌طور معمول شیوه نوشتن و اجرای زبان‌های برنامه‌نویسی به این صورت است که ابتدا دستورها و فرمان‌ها در یک فایل نوشته می‌شوند. سپس یک برنامه دیگر به نام کامپایلر برنامه را کامپایل می‌کند و فایلی تولید می‌کند که ترجمه دستورها به زبان سخت‌افزاری که قرار است برنامه را اجرا کند در آن قرار می‌گیرد. پس از آن سیستم عامل آن دستورها را در سخت‌افزار مورد نظر اجرا می‌کند:




مشکل اصلی این روش این است که با توجه به تنوع سخت‌افزارها و سیستم‌عامل‌ها، اگر بخواهیم برنامه را بر روی سخت‌افزار و سیستم‌عامل دیگری اجرا کنیم، باید کل این فرایند را تکرار کنیم. این مسئله مشکلات زیادی را به شرکت‌های نرم‌افزاری تحمیل می‌کند.

جاوا برای حل این مشکل راهکار هوشمندانه‌ای را به‌کار می‌برد: استفاده از ماشین مجازی جاوا. در روش جاوا برنامه‌ها به جای اینکه بعد از کامپایل به برنامه قابل اجرا در سخت‌افزار تبدیل شوند، به «بایت‌کد» تبدیل می‌شوند. این بایت‌کد را «ماشین مجازی جاوا» به کد قابل اجرا در سیستم عامل و سخت‌افزار تبدیل کرده و برنامه اجرا می‌شود:




این روش جاوا هزینه نگهداری برنامه‌ها را به‌شدت کاهش می‌دهد و همین موضوع یکی از دلایل اصلی محبوبیت جاوا در میان برنامه‌نویس‌ها است.

## نصب پیش‌نیازهای برنامه‌نویسی جاوا

پیش از آنکه برنامه‌نویسی جاوا را آغاز کنیم، باید مقدمات آن را فراهم کنیم. پیش از همه‌چیز نیاز به ابزاری داریم که کدهای برنامه را به کمک آن بنویسیم. هرچند که برای این کار می‌توان از هر ویرایشگر متنی استفاده کرد، ولی بهتر است از ابتدا با ابزارهای مناسب کار را شروع کنیم. این ابزارها در مواقع خطا به ما هشدارهای لازم را می‌دهند. ابزار معروفی که ما از آن برای برنامه‌نویسی جاوا استفاده می‌کنیم «آیدیا» نام دارد. یکی دیگر از پیش‌نیازهای برنامه‌نویسی جاوا، کامپایلر جاوا است که همان‌طور که گفتیم وظیفه‌اش کامپایل برنامه ما و تبدیل کدهای برنامه به بایت‌کد جاوا است.

پخش فیلم

نصب جاوا و آیدیا


## یک فنجان جاوا

بعد از نصب کامپایلر جاوا و محیط برنامه‌نویسی آیدیا، حالا نوبت این است که اولین برنامه جاوا را بنویسیم! ساده‌ترین برنامه جاوایی که می‌توان نوشت برنامه‌ای است که این متن ساده را در خروجی چاپ کند:

I love Mechatronics!

متن این برنامه بسیار ساده است:

```
1  public class Helloworld {
2        public static void main (String [] args)  {
3              System.out.printIn (“I Love Mechatronics!”);
4        }
5  }
```

پخش فیلم

اجرای برنامه فوق در آیدیا

## شیءگرایی در جاوا

جاوا یک زبان شیءگرا است. یعنی در زبان جاوا هر چیزی یک شیء است. از کلاس‌های جاوا برای تعریف یک شیء در برنامه استفاده می‌کنیم. اشیا دارای ویژگی‌ها و رفتار هستند. فرض کنید می‌خواهیم برنامه‌ای بنویسیم و با کمک آن برنامه محیط و مساحت یک دایره را محاسبه کنیم. برای این کار یک کلاس می‌سازیم به نام Circle به معنی دایره. ویژگی‌ای که دو دایره را از هم متمایز می‌کند شعاع آن است. پس کلاس Circle ما، یک ویژگی دارد به نام radius به معنی شعاع. با دانستن شعاع یک دایره می‌توان محیط و مساحت آن را محاسبه کرد. برای این کار دو متد به کلاس Circle اضافه می‌کنیم: perimeter یا محیط و area یا مساحت. یک متد دیگر هم به کلاس اضافه می‌کنیم تا هر زمان که خواستیم بتوانیم شعاع دایره را تغییر بدهیم. اسم این متد را setRadius می‌گذاریم. شکل کلی کلاس ما شبیه این خواهد بود:

```
1  public class Circle {
2       private float radius;
3   .
4      // این متد محیط دایره را محاسبه می‌کند
5      public float perimeter ( )  {
6         // همان عدد پی معروف است Math.PI
7         return 2 * Math.PI * radius;
8         }
9
10        //این متد مساحت دایره را محاسبه می‌کند
11         public float area ( )   {
12              return Math.PI * Math.power (radius, 2);
13         }
14
15         // برای تغییر شعاع دایره از این متد استفاده می کنیم
16         public void setRadius (float r)   {
17               radius  = r;
18         }
19              }
```

حالا یک کلاس دیگر به برنامه اضافه می‌کنیم که دارای متد main است:

```
1  public class Main  {
2       public static void main (String [] ars)   {
3       }
4  }
```

متد main متدی است که اجرای برنامه از آن آغاز می‌شود. حالا وقت آن است که از کلاس Circle یک شیء دایره بسازیم و محیط و مساحت آن را حساب کنیم:

```
1  public class Main  {
2      public static void main (String [] args)   {
3          // ایجاد یک شیء دایره
4          Circle c = new Circle ( );
5          //مقداردهی به شعاع دایره
6          c.setRadius (10.0);
7          //چاپ محیط دایره
8          System.out.printIn (c.perimiter ());
9          // چاپ مساحت دایره
10         System.out.printIn (c.area ());
11     }
12 }
```

پخش فیلم

محاسبه محیط و مساحت یک مربع و مستطیل

فعالیت

یک کلاس دیگر به برنامه اضافه کنید که محیط و مساحت لوزی، متوازی‌الاضلاع، ذوزنقه، مثلث متساوی‌الاضلاع، مثلث قائم‌الزاویه، چندضلعی منتظم، قطاع دایره، مکعب، استوانه، استوانه توخالی، هرم، مخروط کره، جرم کره، جرم طولی، جرم سطحی را محاسبه و آن را چاپ کند.

تمرین۱

برنامه‌ای بنویسید که وزن جسمی را بر حسب گرم از ورودی بگیرد آنگاه در خروجی مشخص سازد که چند کیلوگرم و چند گرم است؟

تمرین۲

برنامه‌ای بنویسید که سن شمارا بر حسب روز از ورودی گرفته آنگاه در خروجی مشخص سازد که سن شما چند سال،چند ماه و چند هفته چند روز دارد؟

## انواع داده‌ها در زبان جاوا

برای انجام محاسبات بر روی داده‌ها در برنامه باید بتوانیم داده‌ها را شناسایی کنیم. باید بدانیم آنها از چه نوعی هستند. همچنین باید بدانیم که در هر لحظه هر کدام از آنها چه مقداری دارند. در جاوا هر متغیر سه ویژگی دارد: نوع، نام و مقدار. الگوی کلی تعریف متغیرها در زبان جاوا به‌صورت زیر است:

```
1  Type name;
2  مانند نمونه‌های زیر //
3  int a;
4          Circle c;
```

پس از تعریف یک متغیر باید بتوانیم مقداری را به آن نسبت دهیم. برای این کار از الگوی زیر استفاده می‌کنیم:

```
1  name = value;
2  مانند مثال‌های زیر //
3 a = 5;
4 c = new Circle ( );
```

در صورتی که بخواهیم می‌توانیم دو مرحله تعریف متغیر و مقداردهی به آن را در یک دستور انجام دهیم. برای این کار از الگوی زیر استفاده می‌کنیم:

```
1 Type name = value;
2   مانند مثال‌های زیر //
3   int a = 5;
4 Circle c = new circle ( );
```

پخش فیلم

آموزش انواع داده‌ها در زبان جاوا

اعداد صحیح در جاوا را با نوع int و long تعریف می‌کنیم. تنها تفاوت آنها در گنجایش یا ظرفیت آنها است. نوع int برای نگهداری مقادیر حدود منفی دو میلیارد تا مثبت دو میلیارد مناسب است و نوع long برای نگهداری مقادیر بسیار بزرگ در حدود منفی ۸ میلیون میلیارد تا مثبت ۸ میلیون میلیارد!

اعداد اعشاری در جاوا را با دو نوع float و double تعریف می‌کنیم. نوع float برای نگهداری مقادیر حدود منفی ۳/۴ ضرب در ۱۰ به توان ۳۸ تا مثبت همین مقدار و با دقت ۸ رقم اعشار مناسب است و نوع double برای نگهداری مقادیر حدود منفی ۱/۴ ضرب در ۱۰ به توان ۳۰۸ تا مثبت همین مقدار و با ۱۶ رقم دقت اعشاری مناسب است.

برای تعریف کاراکترها یا حروف در زبان جاوا از نوع char استفاده می‌کنیم. با توجه به اینکه متغیرهای حرفی در جاوا یونی‌کد هستند، از آنها می‌توان برای کلیه حروف کلیه زبان‌ها (از جمله فارسی) استفاده کرد.

## عملگرها و اولویت آنها در جاوا

تاکنون قطعاً عبارت‌های ریاضی فراوانی را دیده‌اید. برای مثال می‌دانید که ۲+۲ یک عبارت ریاضی است. در زبان‌های برنامه‌نویسی مانند جاوا همچنین عبارت‌هایی وجود دارند.در اغلب عبارت‌های ریاضی ما از نمادهایی مانند * (ضرب)، + (جمع) و مانند آن استفاده می‌کنیم. در اصطلاح برنامه نویسی به این نمادهای ویژه، عملگر می‌گویند. عملگرها نمادهایی هستند که برای محاسبات ریاضی و منطقی از آنها استفاده می‌شود.

**عملگرهای ریاضی:** برای انجام محاسبات ریاضی در جاوا از پنج عملگر جمع (+)، تفریق (-)، ضرب (*)، تقسیم (/) و باقیمانده تقسیم (٪) استفاده می‌کنیم. در جدول زیر فهرست این عملگرها را به همراه مثال‌هایی از کاربرد آنها می‌بینید:

| نام عملگر | معنی | مثال | نتیجه عملیات |
|---|---|---|---|
| + | حاصل جمع | ۵ + ۷ | ۱۲ |
| - | حاصل تفریق | ۳ - ۹ | ۶ |
| - | منفی عدد | -۳ | -۳ |
| * | حاصل ضرب | ۵ × ۳ | ۱۵ |
| / | حاصل تقسیم | ۳ / ۱۵ | ۵ |
| ٪ | باقیمانده تقسیم | ۳ ٪ ۸ | ۲ |

### عملگرهای مقایسه‌ای:

جاوا عملگرهایی برای مقایسه مقادیر متغیرها دارد. حاصل این عملگرها یک متغیر منطقی است که همواره دارای مقدار درست یا true و نادرست یا false است. جدول زیر شامل فهرست کامل عملگرهای مقایسه‌ای جاوا و مثال‌هایی از هر کدام است:

| عملگر | معنی | مثال | توضیح |
|---|---|---|---|
| == | تساوی | x == ۵ | اگر x برابر ۵ باشد حاصل عبارت true و در غیر این صورت false است |
| != | نامساوی | x != ۵ | اگر x برابر ۵ باشد حاصل عبارت false و در غیر این صورت true است |
| < | کوچک‌تر | x < ۵ | اگر x کوچک‌تر از ۵ باشد حاصل عبارت true و در غیر این صورت false است |
| > | بزرگ‌تر | x > ۵ | اگر x بزرگ‌تر از ۵ باشد حاصل عبارت true و در غیر این‌صورت false است |
| <= | کوچک‌تر یا مساوی | x <= ۵ | اگر x کوچک‌تر از ۵ یا مساوی ۵ باشد، حاصل عبارت true و در غیر این صورت false است |
| >= | بزرگ‌تر یا مساوی | x >= ۵ | اگر x بزرگ‌تر از ۵ یا مساوی ۵ باشد، حاصل عبارت true و در غیر این صورت false است |

### عملگرهای منطقی :

عبارت‌هایی که حاصل آنها یک مقدار منطقی درست (true) یا نادرست (false) است، می‌توانند با هم ترکیب شده و عبارت‌های پیچیده‌تری بسازند. برای ترکیب این عبارت‌ها می‌توان از عملگرهای منطقی استفاده کرد. جدول زیر شامل مهم‌ترین و پرکاربردترین عملگرهای منطقی جاوا است:

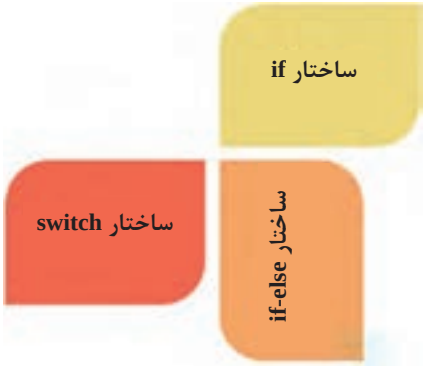
| عملگر | معنی | مثال | توضیح |
|---|---|---|---|
| && | «و» منطقی (and) | a && b | اگر هر دو عبارت منطقی a و b درست (true) باشند، مقدار بازگشتی عبارت، درست و در غیر این صورت مقدار بازگشتی عبارت نادرست (false) است. |
| \|\| | «یا»ی منطقی (or) | a \|\| b | اگر یکی از دو عبارت منطقی درست (true) باشد، مقدار بازگشتی عبارت، درست و در غیر این صورت مقدار بازگشتی عبارت نادرست (false) است. |

فیلم

عملگرها و اولویت آنها در جاوا

## ساختارهای کنترلی در جاوا

برنامه‌های رایانه‌ای نوشته شده با هر زبان برنامه نویسی ـ از جمله جاوا ـ از سه جزء سازنده اصلی تشکیل می‌شوند که اجرای برنامه را کنترل می‌کنند: **توالی، انتخاب** و **تکرار**.




توالی به معنی اجرای دستورهای برنامه به صورت پشت‌سرهم و پیاپی است. در جاوا همه دستورها پشت‌سرهم اجرا می‌شوند.

گاهی مواقع می‌خواهیم مسیر اجرای برنامه را از بین چند مسیر مختلف انتخاب کنیم. مثلاً اگر مقدار متغیر a کمتر از ۵۰ بود یک کار را انجام دهیم و اگر بیشتر از ۵۰ بود کار دیگری را انجام دهیم. جاوا برای ساختار انتخاب از چندین روش مختلف استفاده می‌کند که در این بخش با آنها آشنا خواهید شد:


موارد زیادی پیش می‌آید که می‌خواهیم یک فعالیت را به تعداد زیادی تکرار کنیم. یک حلقه تکرار تعدادی دستور است که چندین بار اجرا می‌شوند. جاوا برای ساختار تکرار از چندین روش مختلف استفاده می‌کند که در این بخش با آنها آشنا خواهید شد:

## ساختار تک انتخابی یا if

اگر بخواهیم یک یا چند دستور را فقط در صورتی اجرا کنیم که یک شرط خاص برقرار باشد، از ساختار if استفاده می‌کنیم. اگر بخواهیم ساختار if را خیلی ساده و به زبان فارسی بیان کنیم به شکل زیر خواهد بود:

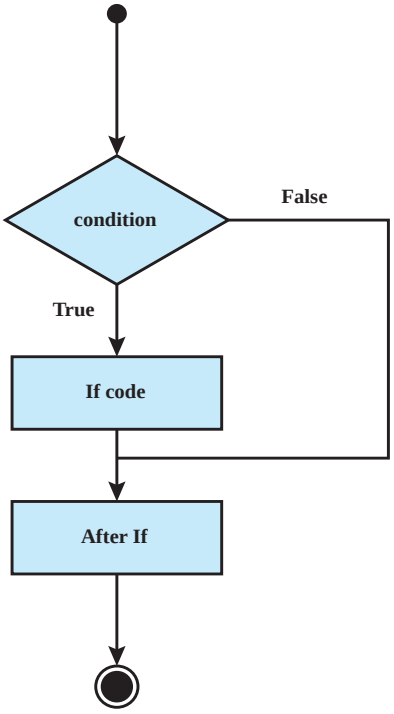
اگر (شرط) برقرار بود آنگاه این دستور (ها) را اجرا کن:
دستور

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

فلوچارت آن به صورت زیر است:
به مثال زیر دقت کنید:

```
1       into a =5;
2       into b = 7;
3       into c = a+b;
4       if (c>10) {
5       System.out.printLn ("a+b is greater than 10!");
6       }
```

فلوچارت آن به‌صورت زیر است:




فعالیت کلاسی ۱

برنامه‌ای بنویسید که تعیین کند آیا یک عدد صحیح زوج است و اگر زوج بود پیغام مناسب چاپ کند.

پخش فیلم

فیلم فعالیت کلاسی ۱

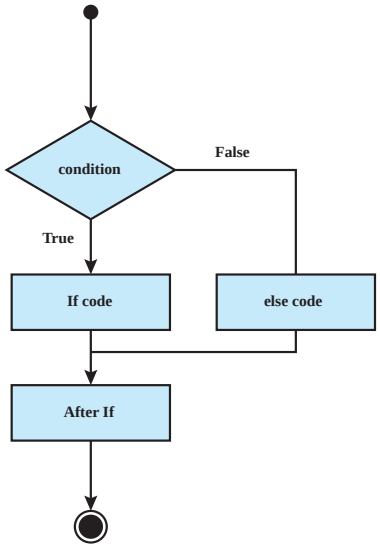
## ساختار دو انتخابی با if–else

در دستور if اگر شرط درست باشد، دستورهای داخل if اجرا می‌شوند و در غیر این صورت، برنامه از روی این دستورها پرش می‌کند و دستورهای بعد از if را اجرا می‌کند. مواقعی پیش می‌آید که می‌خواهیم در صورت غلط بودن شرط دستور if دستور یا دستورهای خاصی را اجرا کنیم. در چنین مواقعی از ساختار if-else استفاده می‌کنیم:

اگر (شرط) برقرار بود آنگاه این دستور(ها) را اجرا کن:
دستورها
در غیر این صورت این دستور(های) زیر را اجرا کن:
دستورها

فلوچارت آن به صورت زیر است:

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




به مثال زیر دقت کنید:
مثال) با فرض اینکه دو عدد برابر نباشند، برنامه‌ای بنویسید که دو عدد را با یکدیگر مقایسه کند و پیغام مناسب دهد؟

```
If(a<b)
System.Out.Println("a is less than b")
Else
System.Out.Println("b is less than a")
```

فعالیت کلاسی ۲

برنامه‌ای بنویسید که مسافت طی شده توسط یک خودرو برحسب کیلومتر و میزان مصرف سوخت آن را بر حسب لیتر گرفته و مصرف سوخت خودرو را در هر صد کیلومتر مشخص کند. اگر مصرف خودرو کمتر از ۷ لیتر در ۱۰۰ کیلومتر بود، پیغام «مصرف سوخت مناسب» و در غیر این صورت پیغام «مصرف سوخت زیاد» چاپ کند.

پخش فیلم

فیلم: فعالیت کلاسی ۲

تمرین ۳

برنامه‌ای بنویسید که ۵ نمره را بگیرد و معدل را محاسبه نماید. اگر معدل بالای ۱۲ باشد پیغام (مشروط نیست) چاپ شود، در غیر این‌صورت پیغام (مشروط است) چاپ شود؟

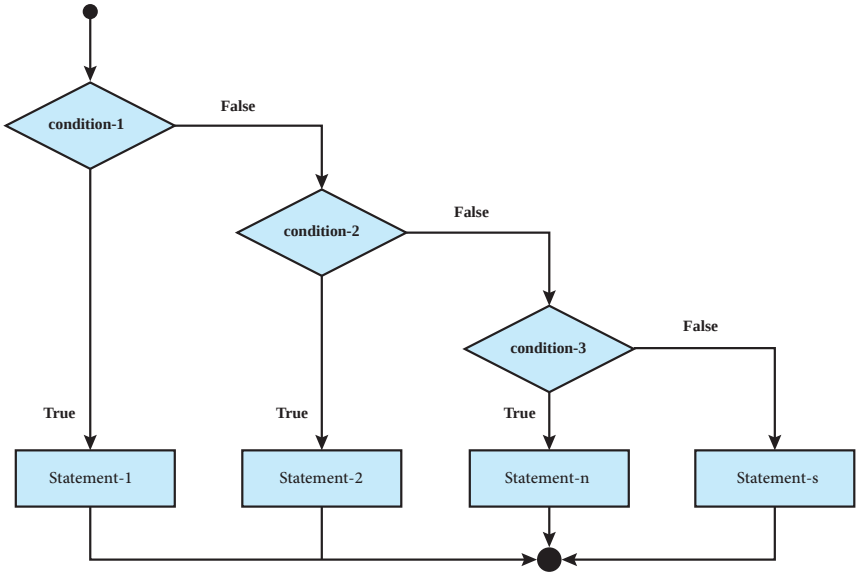
## ترکیب ساختار if و else

اگر «دستور» داخل بخش else، خودش یک if باشد، شکل ساختار if - else این‌گونه می‌شود:

به همین شکل می‌توان چندین ساختار if-else تودرتو را به شکل ساده زیر نوشت:

```
if (شرط (الف))                    اگر شرط (الف) برقرار بود آنگاه این دستور(ها) را اجرا کن:
{
        دستورات                    دستورها
}
else if (شرط (ب))                 در غیر این صورت، اگر شرط (ب) برقرار بود دستور(های) زیر را اجرا کن:
{
        دستورات                    دستورها
}
else if (شرط (ج))                 در غیر این صورت، اگر شرط (ج) برقرار بود این دستور(های) زیر را اجرا کن:
{
        دستورات                    دستورها
}
...                               در غیر این صورت این دستور(های) زیر را اجرا کن:
else
{
        دستورات                    دستورها
}
```

فلوچارت آن به صورت زیر است:



فعالیت کلاسی ۳

برنامه فعالیت کلاسی ۲ را به شکلی تغییر دهید که اگر مصرف سوخت خودرو کمتر از ۵ لیتر در ۱۰۰ کیلومتر بود پیغام «مصرف سوخت بسیار کم» و اگر بین ۵ تا ۷ لیتر بود پیغام «مصرف سوخت مناسب» و اگر بین ۷ تا ۱۰ لیتر بود پیغام «مصرف سوخت زیاد» و اگر بیشتر از ۱۰ لیتر بود پیغام «مصرف سوخت بسیار زیاد» چاپ کند.

تمرین ۴

تمرین ۲ را به گونه‌ای تغییر دهید،که اگر سن محاسبه شده کمتر از ۳ سال بود پیغام (خردسال) و اگر بین ۳ تا ۶ بود، پیغام (کودک) و اگر بین ۶ تا ۱۷ بود، پیغام (نوجوان) و اگر بین ۱۷ تا ۳۹ بود پیغام (جوان)، و اگر بین ۳۹ تا ۶۰ بود پیغام (میانسال) و بالاتر از ۶۰ بود پیغام (کهنسال) چاپ شود؟

پخش فیلم

پاسخ فعالیت کلاسی ۳

## ساختار انتخاب چندتایی با switch

در زبان جاوا برای ساختار if-else تودرتو که در بخش قبل دیدید، یک ساختار ساده‌تر وجود دارد به نام switch. ساختار switch در زبان جاوا به شکل زیر است:

```
Switch (متغیر) {                    انتخاب کن بر اساس مقدار متغیر:
 case value۱:                       در صورتی که برابر مقدار اول بود:
          // دستورات                 دستور
    break;                          خروج
 case value۲:                       در صورتی که برابر مقدار دوم بود:
          // دستورات                 دستور
    break;                          خروج
 default                            در صورتی که متغیر برابر هیچ‌کدام از نمونه‌ها نبود //
          // دستورات                 پیش‌فرض دستورات زیر انجام شود //
}
```

فعالیت کلاسی ۴

برنامه‌ای بنویسید که نوع عملیات ریاضی را از کاربر بگیرد و آن را بر روی اعداد صحیح داده شده اعمال کند. از حرف s برای «جمع» و از حرف m برای «تفریق» و از حرف p برای «توان» استفاده کنید. برای مثال اگر کاربر p را به همراه ۴ و ۳ فرستاد، برنامه باید حاصل ۴ به توان ۳ را چاپ کند.

پخش فیلم

پاسخ فعالیت کلاسی ۴

تمرین ۵

برنامه‌ای بنویسید که کاراکتری که نشان دهند رنگی است، از ورودی بگیرد آنگاه به شما بگوید که چه رنگی را می‌خواهید انتخاب کنید؟ (اگر کاربر کلمه red را وارد کرد پیغام رنگ قرمز چاپ شود، اگر کاربر کلمه green را وارد کرد پیغام رنگ سبز چاپ شود، اگر کاربر کلمه blue را وارد کرد پیغام رنگ آبی چاپ شود، اگر کاربر کلمه Yellow را وارد کرد پیغام رنگ زرد چاپ شود) (برنامه با دستور switch نوشته شود)

## ساختار تکرار با while

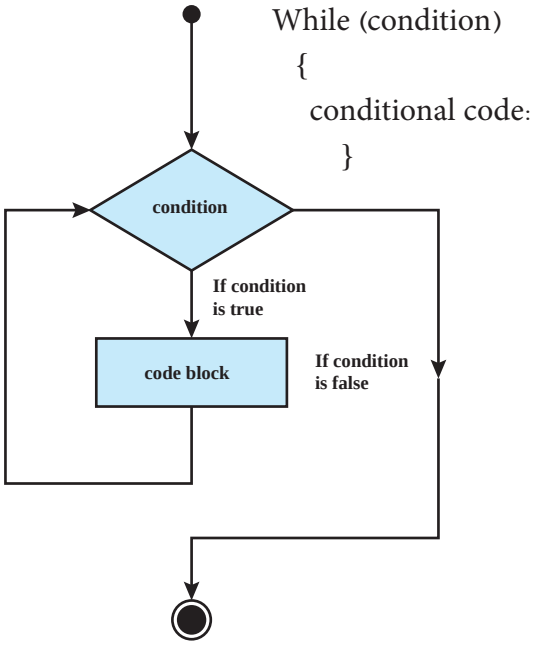
فرض کنید می‌خواهید مجموع و میانگین ۱۰۰ عدد را محاسبه و در خروجی چاپ نمایید. چه می‌کنید؟ آیا ۱۰۰ متغیر تعریف می‌کنید و هر بار مقدار یکی از آنها را از ورودی می‌خوانید؟ یا اینکه یک بار دستور خواندن از ورودی را می‌نویسید و از برنامه می‌خواهید که آن را برای شما ۱۰۰ بار تکرار کند؟ مسلماً روش دوم را استفاده خواهید کرد. بنیادی‌ترین ساختار تکرار در جاوا حلقه while است. ساختار while در جاوا به شکل زیر است:

```
while (condition) {
        // body of loop
}
```

تا زمانی که شرط (while) برقرار است این دستور(ها) را اجرا کن:
دستورات داخل حلقه

فلوچارت آن به صورت زیر است:

**دیاگرام حلقه while**



اگر شرط while در ابتدا برقرار نباشد، دستور یا دستورهای داخل آن هرگز اجرا نمی‌شوند و اگر شرط همیشه درست باشد، هیچ‌وقت از حلقه تکرار خارج نمی‌شویم. بنابراین همیشه در نوشتن حلقه‌های تکرار مراقب شرط حلقه تکرار باشید.

فعالیت کلاسی ۵

برنامه‌ای بنویسید که مجموع اعداد از ۱ تا ۱۰۰ را چاپ کند.

پخش فیلم

پاسخ فعالیت کلاسی ۵

تمرین ۶

برنامه‌ای بنویسید که یک عدد از ورودی دریافت کند آنگاه آن را به صورت معکوس چاپ کند.

## ساختار تکرار با do–while

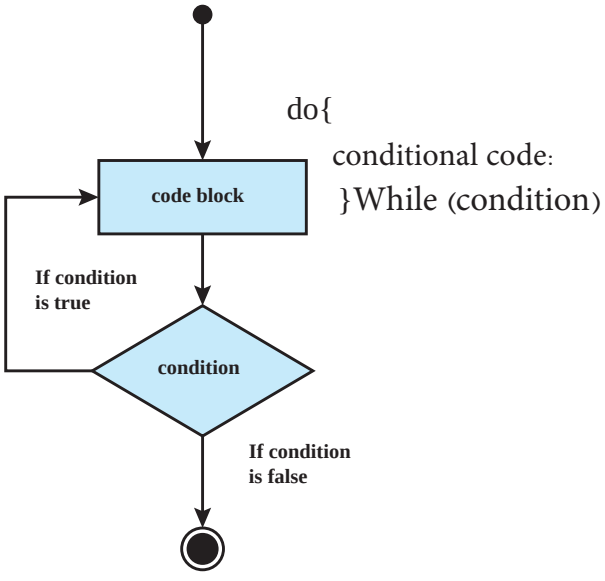
دستور do-while از نظر مفهومی بسیار شبیه به حلقه تکرار while است. به ساختار این دستور دقت کنید:

```
do {
// body of loop
} while (condition);
```

این دستورات را اجرا کن

سپس شرط دستور (while) را بررسی کن و در صورتی که شرط برقرار باشد دوباره دستورات را انجام بده.

فلوچارت آن به صورت زیر است:




دیاگرام حلقه do while

بزرگ‌ترین تفاوت بین حلقه‌های تکرار while و do-while این است که در حلقه while شرط حلقه در ابتدای هر تکرار بررسی می‌شود ولی در حلقه do-while ابتدا یک تکرار انجام شده و سپس شرط حلقه بررسی می‌شود. بنابراین در حلقه do-while بدون توجه به درستی یا نادرستی شرط، دستور(ها) حداقل یک بار انجام می‌شوند.

فعالیت کلاسی ۶

برنامه‌ای که برای switch نوشته بودید را به‌گونه‌ای تغییر دهید که برنامه تا زمانی که کاربر q به معنای «خروج» وارد نکرده است ادامه پیدا کند.

پخش فیلم

پاسخ فعالیت کلاسی ۶

تمرین ۷

برنامه‌ای بنویسید که تا زمانی‌که کاربر عدد صفر را از ورودی وارد نکرده، از ورودی عدد دریافت کند آنگاه مجموع اعداد دریافت شده را چاپ نماید؟

## ساختار تکرار با for

حلقه تکرار for یک ساختار تکرار با شمارنده یا نگهبان است. اگر بخواهیم کاری را به تعداد مشخصی تکرار کنیم، می‌توانیم از حلقه تکرار for استفاده کنیم. حلقه for از اجزای زیر تشکیل شده است:

**شمارنده و مقدار اولیه آن:** شمارنده یک متغیر است که برای کنترل تعداد تکرارهای حلقه از آن استفاده می‌شود. این شمارنده باید یک مقدار اولیه داشته باشد.

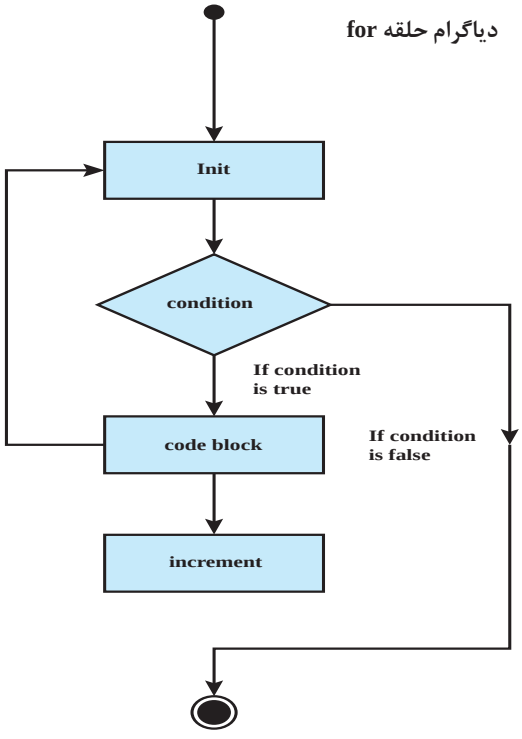
**گام پرش شمارنده:** مقداری که در هر بار تکرار دستور یا دستورهای حلقه به شمارنده اضافه شده یا از آن کم می‌شود.

**شرط اتمام تکرار:** for هم مثل سایر حلقه‌های تکرار برای اتمام تکرار دستورهای خود نیاز به یک شرط دارد و تا زمانی‌که آن شرط برقرار باشد دستور یا دستورهای داخل آن تکرار می‌شوند.

for (شمارنده و مقدار اولیه; شرط اتمام تکرار; گام پرش شمارنده)
دستور (ها)

فلوچارت آن به صورت زیر است:

دیاگرام حلقه for



نمونه‌ای از یک حلقه تکرار for برای چاپ اعداد زوج کمتر از ۱۰۰ شبیه قطعه کد زیر خواهد شد:

```
1   // شمارنده و مقدار اولیه آن: int i = 0
2   // شرط اتمام تکرار و خروج از حلقه: i < 100
3   // گام پرش شمارنده: i = i + 2
4   for(int i = 0; i < 100; i = i + 2)  {
5           System.out.printIn(i);
6                                                   }|
```

در این مثال شمارنده متغیر i و مقدار اولیه آن هم صفر است. گام پرش شمارنده ۲ است و در هر بار تکرار به شمارنده دو واحد اضافه می‌شود. شرط اتمام این است که شمارنده از ۱۰۰ کمتر باشد.

تمرین ۸

برنامه‌ای بنویسید که بین ۱۰۰ عدد بزرگ‌ترین و کوچک‌ترین را پیدا کند؟

تمرین ۹

برنامه‌ای بنویسید که یک عدد از ورودی گرفته آنگاه مشخص نماید عدد اول است یا خیر؟

تمرین ۱۰

برنامه‌ای بنویسید که دو عدد از ورودی بگیرد و اعداد بین آنها را چاپ نماید؟

تمرین ۱۱

برنامه‌ای بنویسید که شماره دانشجویی و معدل ۱۰ دانشجو را بگیرد سپس مشخص کند کدام شماره دانشجویی بالاترین معدل را دارد؟

## حلقه‌های تکرار تو در تو

در هر ساختار تکرار می‌توانیم تعدادی دستور جاوا را اجرا کنیم، ولی آیا امکان دارد که در یک حلقه تکرار، یک حلقه تکرار دیگر داشته باشیم؟ قطعاً بله، البته با دقت و احتیاط! حلقه‌های تکرار تودرتو کاربرد بسیار زیادی در الگوریتم‌ها و محاسبات دارند.

فعالیت کلاسی ۷

برنامه‌ای بنویسید که جدول ضرب را در تولید و در خروجی نمایش دهد.

پخش فیلم

فیلم: فعالیت کلاسی ۷

تمرین ۱۲

برنامه‌ای بنویسید که خروجی زیر را تولید نماید؟

```
+
**
+++
****
+++++
```

تمرین ۱۳

برنامه‌ای بنویسید که خروجی زیر را تولید نماید؟

```
  *
 ***
*****
 ***
  *
```

تمرین ۱۴

برنامه‌ای بنویسید که خروجی زیر را تولید نماید؟

```
   ۱
  ۲۲۲
 ۳۳۳۳۳
۴۴۴۴۴۴۴
```

## آرایه‌ها

فرض کنید که می‌خواهید نمره‌های دانشجویان یک کلاس را گرفته و میانگین آنها را محاسبه کنید. تعداد دانشجویان کلاس ۲۵ نفر است. آیا ۲۵ متغیر از نوع اعشاری تعریف می‌کنید؟ اگر تعداد دانشجویان کلاس ۵۰ نفر بود چه کار می‌کردید؟ برای ۲۰۰ نفر چه می‌کردید؟ همان‌طور که حدس می‌زنید راه‌حل، استفاده از تعداد زیادی متغیر نیست. اگر می‌توانستیم تعداد زیادی متغیر همنوع را با یک نام ذخیره کنیم و با استفاده از یک اندیس به آنها دسترسی داشته باشیم مشکل حل می‌شد. مثلاً می‌گفتیم فهرست نمره‌های دانشجویان که شامل ۲۵ متغیر float است و بعد می‌گفتیم نمره دانشجوی اول ۱۵ و دانشجوی دوم ۱۶ و ... است. به چنین نوع داده‌ای آرایه گفته می‌شود. آرایه (Array) مجموعه‌ای از متغیرها است که عنصر (element) یا جزء (Component) نامیده می‌شوند و همگی از یک نوع (type) هستند. یک آرایه با چند چیز شناخته می‌شود: نام آن، تعداد متغیرهایی که نگهداری می‌کند که طول آرایه نامیده می‌شود و نوع متغیرهایی که آرایه در خود نگه می‌دارد. بنابراین برای تعریف یک آرایه که نمره‌های دانشجویان را نگهداری کند به شکل زیر عمل می‌کنیم:

```
1    float [25] grades;
```

به اجزای تعریف فوق دقت کنید. float نوع آرایه را مشخص می‌کند. در واقع می‌گوید که آرایه فوق عناصری از نوع float را نگهداری می‌کند. [۲۵] اعلام می‌کند که طول آرایه ۲۵ است. [] نشان دهنده آرایه است و عدد صحیحی که درون آن قرار می‌گیرد طول آرایه را مشخص می‌کند. grades هم که نام آرایه است. حال اگر بخواهیم عناصر این آرایه را مقداردهی کنیم به شکل زیر عمل می‌کنیم:

```
1    grades [0]=12.3f;
2    grades [1]=15.5f;
3    //...
4    grades [24]=16f;
```

بعد از آشنایی با ساختارهای زبان جاوا، با مثال‌ها و تمرین‌های کاربردی، با آرایه‌ها بیشتر آشنا خواهید شد.

## آرایه‌ها و حلقه‌های تکرار

پیش از این درباره آرایه‌ها گفتیم که آرایه‌ها مجموعه‌ای از متغیرهای همنوع هستند. از آرایه‌ها برای ذخیره کردن اطلاعات در تعداد زیاد استفاده می‌شود، مثلاً لیست دانش‌آموزان یک کلاس، لیست حروف یک جمله، لیست پیام‌های ارسال یا دریافت شده، لیست حقوق کارمندان یک شرکت، لیست خودروهای پارک شده در یک پارکینگ عمومی، لیست شماره تلفن‌های دفتر تلفن و ... معمولاً ما این اطلاعات را نگهداری می‌کنیم تا بتوانیم بر روی آنها محاسبات انجام دهیم. مثلاً از لیست کارمندان و ساعت‌های کاری آنها استفاده می‌کنیم تا بتوانیم حقوق آنها را محاسبه کنیم. همان‌طور که حدس می‌زنید، برای کار کردن بر روی آرایه‌ها باید بتوانیم عناصر آرایه را یکی یکی بررسی کنیم و محاسبات لازم را بر روی آنها انجام دهیم و برای این کار، بهترین ابزار ما ساختارهای تکرار هستند.

به علت کاربرد بسیار فراوان آرایه‌ها در الگوریتم‌های برنامه‌نویسی و در مسائل محاسباتی، شیوه‌های مرتب کردن عناصر آرایه‌ها و نیز جست‌وجو برای پیدا کردن یک عنصر خاص در آرایه‌ها اهمیت فراوانی دارد.

فعالیت کلاسی ۸

برنامه‌ای بنویسید که پرتاب تاس را شبیه‌سازی کند. ببینید بعد از ده هزار پرتاب تاس چند بار اعداد ۱ تا ۶ تکرار شده‌اند.

پخش فیلم

فیلم: فعالیت کلاسی ۸

تمرین ۱۵

برنامه‌ای بنویسید که ۱۰ عدد را از کاربر دریافت کند آنگاه اعداد را از انتها به ابتدا چاپ نماید؟

تمرین ۱۶

برنامه‌ای بنویسید که معدل دانشجویان یک کلاس ۲۰ نفره را دریافت کند سپس مشخص کند چند نفر معدل بالای ۱۸ و چند نفر معدل بین ۱۴ تا ۱۸ و چند نفر معدل زیر ۱۸ را دارند، همچنین ۳ معدل برتر را چاپ نماید؟

تمرین ۱۷

برنامه‌ای بنویسید که ۵ عدد را از کاربر دریافت کرده آنگاه آنها را چاپ می‌کند،سپس بزرگ‌ترین عدد را پیدا کند؟

تمرین ۱۸

برنامه‌ای بنویسید که حقوق تعدادی از کارکنان مؤسسه‌ای را دریافت کند آنگاه آنها را بر اساس تعرفه زیر، مالیات حقوق آنها را محاسبه کند و به خروجی ببرد همچنین حقوق خالصی دریافتی فرد را چاپ نماید و مشخص کند بین کارمندان چه مبلغی بیشترین حقوق است؟

| | |
|---|---|
| حقوق <=۲۰۰۰۰۰۰ | معاف از مالیات |
| حقوق >۲۰۰۰۰۰۰ <=۳۰۰۰۰۰۰ | ۱۰ درصد مالیات |
| حقوق ≤۳۰۰۰۰۰۰ | ۱۷ درصد مالیات |

## مرتب کردن آرایه

مرتب کردن یعنی قرار دادن عناصر آرایه با یک ترتیب خاص در کنار یکدیگر. برای مثال یک آرایه از اعداد صحیح را می‌توان به صورت صعودی (کمترین مقدار در ابتدای آرایه و بیشترین مقدار در انتهای آرایه) یا نزولی (بیشترین مقدار در ابتدای آرایه و کمترین مقدار در انتهای آرایه) مرتب کرد. برای مرتب کردن آرایه‌ها الگوریتم‌های فراوانی وجود دارد که یکی از ساده‌ترین آنها «الگوریتم مرتب‌سازی حبابی» است. فرض کنید می‌خواهیم آرایه را به‌صورت صعودی مرتب کنیم. از ابتدای آرایه شروع می‌کنیم و عناصر را دو به دو با هم مقایسه می‌کنیم. اگر عنصر دوم کوچک‌تر از عنصر اول بود، جای آنها را با هم عوض می‌کنیم و بعد به سراغ عنصر بعدی می‌رویم و همین کار را تکرار می‌کنیم ... در اولین اجرای این الگوریتم بزرگ‌ترین عضو آرایه به انتهای آرایه منتقل می‌شود. اگر یک بار دیگر این الگوریتم را تکرار کنیم، دومین عضو بزرگ آرایه به یکی مانده به آخر آرایه منتقل می‌شود. اگر این الگوریتم را به تعداد عناصر آرایه تکرار کنیم، آرایه به‌طور کامل مرتب می‌شود. برای مثال فرض کنید آرایه ما به شکل زیر است:

| ۴ | ۵ | ۱ | ۷ | ۳ |
|---|---|---|---|---|

ابتدا ۴ و ۵ را با هم مقایسه می‌کنیم. از آنجایی که ۵ بزرگ‌تر است نیازی به جابه‌جایی آنها نیست. بعد ۵ را با یک مقایسه می‌کنیم و جای آنها را عوض می‌کنیم:

| ۴ | ۱ | ۵ | ۷ | ۳ |
|---|---|---|---|---|

سپس ۵ را با ۷ مقایسه می‌کنیم و می‌بینیم که نیازی به جابه‌جایی آنها نیست. حالا ۷ را با ۳ مقایسه می‌کنیم و جای آنها را با هم عوض می‌کنیم:

| ۴ | ۵ | ۱ | ۳ | ۷ |
|---|---|---|---|---|

همان‌طور که می‌بینید بزرگ‌ترین عضو آرایه به انتهای آرایه منتقل شده است. اگر برگردیم و یک‌بار دیگر همه این مراحل را تکرار کنیم، این بار ۵ به جایگاه یکی مانده به آخر منتقل می‌شود. اگر این روند را حداکثر ۵ بار تکرار کنیم، کل عناصر آرایه به ترتیب صعودی مرتب می‌شوند.

فعالیت کلاسی ۹

برنامه‌ای بنویسید که یک آرایه از نوع اعداد صحیح با صد عضو تصادفی (رندوم) بسازد و سپس آن را با الگوریتم مرتب سازی حبابی مرتب کند.

پخش فیلم

فیلم: فعالیت کلاسی ۹

## جست‌وجو در آرایه

پیدا کردن یک مقدار در میان عناصر آرایه یکی از مهم‌ترین کارهایی است که بر روی یک آرایه انجام می‌شود. فرض کنید به دنبال یک واژه خاص در یک واژه‌نامه می‌گردید یا شماره تلفن یک فرد در دفتر تلفن. ساده‌ترین روش جست‌وجو در یک آرایه، «جست‌وجوی خطی» است. در جست‌وجوی خطی مقدار مورد نظر را با همه عناصر آرایه مقایسه می‌کنیم تا آن را پیدا کنیم. در آرایه‌های کوچک این الگوریتم به سادگی جست‌وجو را انجام می‌دهد اما اگر تعداد عناصر آرایه زیاد باشد، ممکن است روش مقرون به صرفه‌ای نباشد. روش دیگر جست‌وجو، الگوریتم «جست‌وجوی دودویی» است. جست‌وجوی دودویی فقط بر روی آرایه‌های مرتب شده کار می‌کند بنابراین لازم است پیش از جست‌وجو آرایه را مرتب کنیم. حالا مقداری را که می‌خواهیم جست‌وجو کنیم با عنصر وسط آرایه مقایسه می‌کنیم. اگر برابر بود که جست‌وجو تمام شده است. اگر مقدار از عنصر میانی آرایه کوچک‌تر بود، جست‌وجو را در نیمه ابتدایی و اگر بزرگ‌تر بود در نیمه انتهایی آرایه تکرار می‌کنیم. فرض کنیم می‌خواهیم ببینیم مقدار ۱۹ در آرایه زیر وجود دارد یا نه:

| ۴ | ۱۳ | ۱۶ | ۱۹ | ۳۲ | ۴۸ | ۵۱ | ۵۹ | ۶۸ |
|---|---|---|---|---|---|---|---|---|

ابتدا ۱۹ را با عنصر میانی آرایه یعنی ۳۲ مقایسه می‌کنیم. چون ۱۹ از ۳۲ کمتر است، جست‌وجو را به نیمه ابتدایی آرایه محدود می‌کنیم:

| ۴ | ۱۳ | ۱۶ | ۱۹ |
|---|---|---|---|

حالا چون تعداد عناصر آرایه زوج است می‌توانیم ۱۳ یا ۱۶ را به عنوان عضو میانی انتخاب کنیم. ۱۹ را با ۱۳ مقایسه می‌کنیم، چون ۱۹ از ۱۳ بزرگ‌تر است، محدودهٔ جست‌وجوی ما باز هم نصف می‌شود:

| ۱۶ | ۱۹ |
|---|---|

حالا ۱۶ را به عنوان عنصر میانی در نظر می‌گیریم و با مقداری که به دنبال آن بودیم مقایسه می‌کنیم. چون مقدار مورد نظر ما یعنی ۱۹ از ۱۶ بزرگ‌تر است جست‌وجو را محدود می‌کنیم به آخرین عنصر باقیمانده آرایه و از آنجایی که این دو مقدار برابر هستند جست‌وجوی ما با موفقیت به پایان می‌رسد.

همان‌طور که می‌بینید در هر بار اجرای این الگوریتم، ما نیمی از آرایه را دور می‌ریزیم و این باعث می‌شود که جست‌وجو در یک آرایه مرتب شده بسیار بزرگ به سرعت انجام شود. با استفاده از این روش، برای جست‌وجو در یک آرایه با هزار عضو، حداکثر ۱۰ مقایسه لازم است و در یک آرایه با یک میلیون عنصر حداکثر بیست مقایسه انجام می‌شود در حالی‌که اگر می‌خواستیم به صورت خطی جست‌وجو کنیم حداکثر یک میلیون مقایسه باید انجام می‌دادیم.

فعالیت کلاسی ۱۰

برنامه فعالیت کلاسی ۹ را به صورتی تغییر دهید که کاربر بتواند آرایه مرتب شده را برای پیدا کردن یک مقدار خاص به صورت دودویی جست‌وجو کند.

پخش فیلم

فیلم: فعالیت کلاسی ۱۰

## برنامه‌نویسی اندروید

همان‌طور که پیش از این گفتیم اندروید یک سیستم عامل منبع باز براساس لینوکس است. برای برنامه‌نویسی برای این سیستم عامل دو روش اصلی وجود دارد: روش «بومی» و روش «ترکیبی».

در روش بومی برنامه‌های اندروید به زبان جاوا نوشته می‌شوند و توسط ابزارهایی که اندروید در اختیار ما می‌گذارد کامپایل شده و در گوشی‌های موبایل اندرویدی اجرا می‌شوند.

در روش ترکیبی برنامه‌ها به زبان HTML و جاوا اسکریپت که مخصوص برنامه‌نویسی وب هستند نوشته شده و سپس به کمک یک مرورگر وب در گوشی اجرا می‌شوند. مزیت برنامه‌نویسی بومی سرعت بسیار بالاتر اجرای برنامه‌ها است.

## نصب اندروید استودیو

برنامه‌های بومی اندروید را در یک محیط برنامه‌نویسی ویژه به نام «اندروید استودیو» می‌نویسیم. اندروید استودیو نسخه‌ای از «آیدیا» است که ابزارهای مخصوص برنامه‌نویسی اندروید به آن اضافه شده است. بنابراین تفاوت زیادی در کار با آن احساس نخواهید کرد.

پخش فیلم

دانلود و نصب اندروید استودیو

## اجرای برنامه‌های اندروید

برای آزمایش و رفع خطاهای برنامه‌های اندروید باید بتوانید آنها را در دستگاه‌های اندروید نصب کنید. برای این کار می‌توانید از گوشی‌ها و تبلت‌های اندرویدی استفاده کنید یا شبیه‌ساز اندروید را بر روی رایانه خود نصب کرده و برنامه را روی شبیه‌ساز تست کنید.

پخش فیلم

آماده‌سازی گوشی تلفن همراه برای تست برنامه‌های اندروید

پخش فیلم

نصب شبیه‌ساز اندروید و تست برنامه‌های اندروید در آن

## اجزای برنامه‌های اندروید


یک برنامه اندروید از یک یا چند صفحه تشکیل شده است که هر کدام از آنها یک کار یا فعالیت را انجام می‌دهند. به این صفحه‌ها در برنامه‌نویسی اندروید «اکتیویتی» می‌گوییم. هر اکتیویتی از دو بخش اصلی تشکیل شده است: طراحی گرافیکی صفحه که یک فایل xml است و به آن «چیدمان» می‌گوییم و کد اکتیویتی که اطلاعات را از چیدمان می‌گیرد، آنها را پردازش می‌کند و در صفحه نمایش می‌دهد و... کد اکتیویتی به زبان جاوا نوشته می‌شود.

## چیدمان اکتیویتی

هر صفحه در رابط کاربری اندروید، از عناصر فراوانی تشکیل شده است. به عکس‌های زیر دقت کنید:


همان‌طور که در عکس می‌بینید، هر صفحه رابط کاربری یک برنامه از عناصر و اجزای زیادی ساخته شده است که نحوه قرار گرفتن آنها در کنار یکدیگر، ممکن است به شیوه‌های متفاوتی باشد. مثلاً در عکس فوق، ممکن است یکی بگوید بهتر است عکس‌های آگهی‌ها در سمت راست باشد نه در سمت چپ. یا مثلاً بگوید جای دکمه‌های جست‌وجو و اضافه کردن عوض شود. در ضمن به علت محدودیت‌های فضای صفحه نمایش در موبایل‌ها و تبلت‌ها، قسمتی از رابط کاربری در نگاه اول به کاربر نمایش داده نمی‌شود. عکس روبه‌رو ببینید،


در این عکس، منویی که کاربر از طریق آن شهر محل سکونت خود و نیز دسته‌بندی آگهی‌های مورد علاقه خود را انتخاب می‌کند، در یک منوی کشویی قرار داده شده و کاربر باید آن را از سمت راست بکشد تا نمایش داده شود. همه اینها به ما نشان می‌دهد که برنامه‌نویس باید بتواند رابط کاربری مورد نظر خود را به هر شکل که می‌خواهد بسازد و برای این کار اندروید ابزاری دارد به نام مدیر چیدمان. با کمک این ابزار می‌توانید مثلاً بگویید دکمه جست‌وجو کجای صفحه قرار می‌گیرد. از میان انواع مدیر چیدمان‌ها دو «چیدمان خطی» و «چیدمان وابسته» از کارایی بیشتری برخوردارند و با ترکیب آنها با هم می‌توان تقریباً هر رابط کاربری پیچیده‌ای را ساخت.

## چیدمان خطی

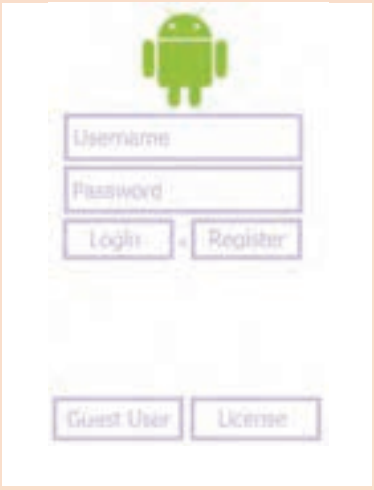
در چیدمان خطی عناصر صفحه پشت سر هم قرار می‌گیرند. اگر جهت چیدمان خطی «افقی» باشد، عناصر از سمت چپ صفحه به راست چیده می‌شوند و اگر جهت آن «عمودی» باشد، عناصر از بالا به پایین چیده می‌شوند.

پخش فیلم

آشنایی با چیدمان خطی

فعالیت کلاسی ۱۱

با استفاده از چیدمان خطی، صفحه‌ای شبیه عکس روبه‌رو را بسازید:




## چیدمان وابسته

در چیدمان وابسته، موقعیت هر عنصر در صفحه نسبت به سایر عناصر تعریف می‌شود. مثلاً یک عنصر صفحه می‌تواند بالا، پایین، چپ یا راست یک عنصر دیگر قرار بگیرد.

پخش فیلم

آشنایی با چیدمان وابسته

فعالیت کلاسی ۱۲

سعی کنید با چیدمان وابسته، فعالیت کلاسی ۱۱ را انجام دهید. مشکلات کار را بنویسید.

## ترکیب چیدمان‌ها

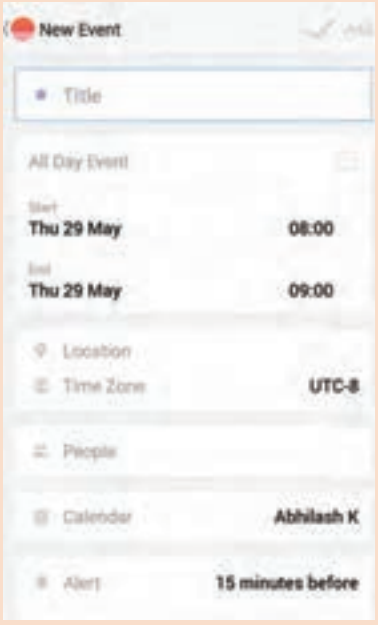
یک قانون طلایی درباره چیدمان‌ها وجود دارد: «چیدمان‌ها به هر تعداد و با هر شکل می‌توانند با هم ترکیب شوند». مثلاً می‌توانید داخل یک چیدمان خطی عمودی، تعدادی چیدمان خطی دیگر (افقی یا عمودی) و یک یا چند چیدمان وابسته داشته باشید.

پخش فیلم

آشنایی با ترکیب چیدمان‌ها

فعالیت کلاسی ۱۳

سعی کنید با ترکیب مناسبی از چیدمان‌های خطی و وابسته، صفحه‌ای شبیه عکس روبه‌رو را بسازید. این عکس از یک برنامه واقعی برداشته شده است.




## ارتباط چیدمان با کد جاوا

منطق برنامه در کدهای جاوای داخل کلاس اکتیویتی نوشته می‌شود و طراحی رابط کاربری آن در فایل‌های xml باید راهی باشد تا بتوان بین این دو ارتباط برقرار کرد. مثلاً اگر کاربر بخواهد با کلیک کردن بر روی یک دکمه اطلاعات داخل فرم به سرور برنامه ارسال شود، باید راهی باشد تا در کد جاوا بتوان به اطلاعاتی که کاربر در فرم وارد کرده است دسترسی پیدا کرد.

کلاس Activity که کلاس مافوق همه اکتیویتی‌های اندروید است متدی به نام setContentView دارد که چیدمان اکتیویتی را مشخص می‌کند. بعد از اینکه چیدمان اکتیویتی را مشخص کردیم، باید بتوانیم به عناصری که در چیدمان قرار دارند دسترسی داشته باشیم. برای این کار از متد findViewById استفاده می‌کنیم. مثلاً اگر بخواهیم در جاوا به متنی که کاربر در یک EditText وارد کرده است دسترسی داشته باشیم، ابتدا یک EditText در اکتیویتی تعریف می‌کنیم و بعد با استفاده از متد findViewById مشخص می‌کنیم که این

EditText کدام EditText در چیدمان است و بعد مقداری که کاربر وارد کرده است را می‌خوانیم:

```
1   public class Main Activity extends Activity   {
2       private EditText nameEditText;
3
4       public voide onCreate (Bundle savedInstanceState)   {
5            super .onCreate (savedInstanceState);
6            // تعیین چیدمان اکتیویتی
7           setContentView (R.layout.activity_main);
8           // پیدا کردن ادیتور متن
9           nameEditText = (EditText) findViewById(R.id. nameEditText);
10          // گرفتن متنی که در ادیتور نوشته شده است
11           String name = nameEditText.getText( ) . toString ( );
12           // ...
13      }
14
15       // ... سایر توابع
16          }|
17
```

فعالیت کلاسی ۱۴

برنامه اندرویدی بنویسید که نام و نام خانوادگی کاربر را بگیرد و با زدن دکمه به کاربر پیغام خوش‌آمدگویی نشان دهد.

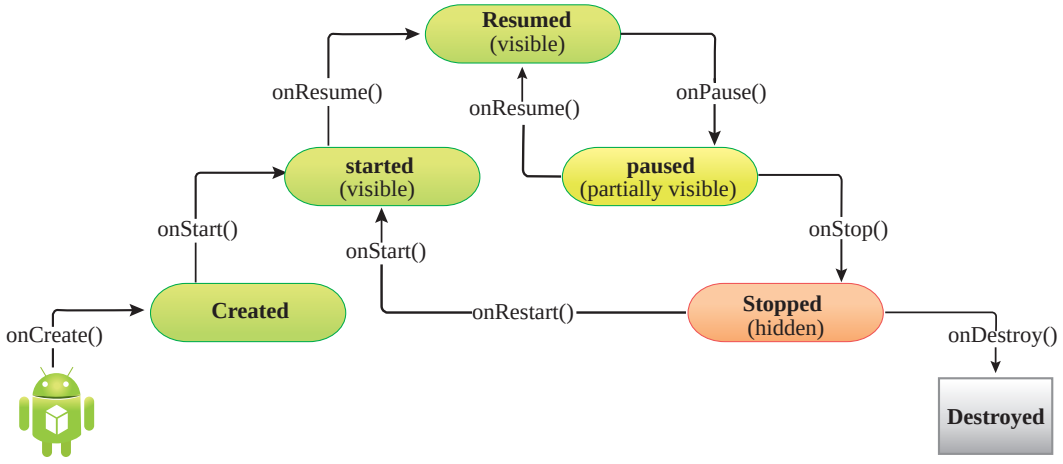
پخش فیلم

ارتباط چیدمان با کد جاوا و پاسخ فعالیت کلاسی ۱۴

## چرخه زندگی اکتیویتی

کل فرایند به‌وجود آمدن و از بین رفتن اکتیویتی‌ها ـ که آن را چرخه زندگی می‌نامیم- توسط سیستم عامل اندروید مدیریت می‌شود. هر زمان که اکتیویتی وارد یکی از مراحل چرخه زندگی خود می‌شود برنامه ما از آن مطلع می‌شود. مثلاً زمانی که اکتیویتی در مرحله اول ایجاد است، متد onCreate صدا زده می‌شود و ما می‌توانیم با فراخوانی متد setContentView چیدمان اکتیویتی را مشخص کنیم. چرخه زندگی یک اکتیویتی بسیار پیچیده است ولی مهم‌ترین مراحل آن اینها است:

- onCreate: اکتیویتی در مراحل آغازین ساخته شدن است.
- onStart: اکتیویتی ساخته شده ولی هنوز به کاربر نمایش داده نشده است.
- onResume: اکتیویتی به کاربر نمایش داده شده است.
- onPause: اکتیویتی دیگر به کاربر نمایش داده نمی‌شود ولی همچنان زنده است. در این حالت اگر کاربر به اکتیویتی برگردد، اکتیویتی دوباره به مرحله onResume می‌رود.
- onStop: اکتیویتی در حال نابودی است. اگر کاربر در این مرحله به اکتیویتی برگردد، اکتیویتی دوباره وارد مرحله onStart می‌شود.
- onDestroy: اکتیویتی به‌طور کامل نابود شده است.




فعالیت کلاسی ۱۵

برنامه‌ای بنویسید که در همه مراحل چرخه زندگی اکتیویتی پیغامی را در لاگ چاپ کند. ببینید که یک اکتیویتی چه مراحلی را در طول دوره زندگی خود طی می‌کند.

پخش فیلم

آشنایی با چرخه زندگی اکتیویتی و پاسخ فعالیت کلاسی ۱۵

## بلوتوث

بلوتوث استاندارد ارتباط بی‌سیم برد کوتاه برای تبادل اطلاعات است. یکی از مهم‌ترین ویژگی‌های بلوتوث مصرف بسیار کم آن است و همین ویژگی باعث شده است که در تلفن همراه و دستگاه‌های مصرف با توان محدود بسیار از آن استقبال شود. بلوتوث انتقال هر نوع اطلاعات و صدا و تصویر را پشتیبانی می‌کند و به همین دلیل می‌توان از آن برای ارسال فرمان و دستور به دستگاه‌های جانبی هم استفاده کرد.

یکی از مهم‌ترین استفاده‌های بلوتوث در حال حاضر استفاده از آن در اینترنت اشیا است. اینترنت اشیا به معنی اتصال بسیاری از چیزها یا اشیا اطراف ما به اینترنت است. این اشیا متصل به اینترنت را می‌توان به کمک برنامه‌های تلفن همراه یا تبلت کنترل کرد. در اینترنت اشیا دستگاه‌ها اطلاعات محیط را جمع‌آوری و از طریق بلوتوث به تلفن همراه می‌فرستند و موبایل هم پس از پردازش این اطلاعات آنها را به سرورهای خدمات دهنده می‌فرستند. کاربران می‌توانند با استفاده از ارتباط بین تلفن همراه و دستگاه‌های متصل، فرمان‌ها یا دستورها را به آنها ارسال کنند و به این ترتیب کاربران می‌توانند یک ارتباط و تعامل دوطرفه بین این دستگاه‌ها برقرار کنند. از این ارتباط ساده و کم هزینه می‌توان برای کنترل کردن تلویزیون از طریق تلفن همراه تا نظارت بر زیرساخت‌های شهری و ترافیک استفاده کرد.

پخش فیلم

اینترنت اشیا و کاربردهای آن در زندگی روزمره

**برنامه‌نویسی ارتباط بلوتوثی در اندروید:** اندروید از همان اولین نسخه پشتیبانی مناسبی از بلوتوث داشته است و برنامه‌نویسی بلوتوث در اندروید بسیار ساده است. برای برقراری یک ارتباط بلوتوثی بین دو دستگاه مختلف باید ابتدا به دنبال دستگاه‌های بلوتوثی فعال در محدوده دستگاه بگردید و سپس دستگاه خودتان را با دستگاه دیگر هماهنگ کنید. بعد از آن یک مسیر ارتباطی بین دو دستگاه ایجاد کنید و از طریق آن به ارسال و دریافت اطلاعات بپردازید.

**مجوز دسترسی به بلوتوث:** برای دسترسی به بلوتوث دستگاه، نیاز به اجازه کاربر دارید. کاربر باید تأیید کند که برنامه می‌تواند به بلوتوث و لیست دستگاه‌های هماهنگ شده دسترسی داشته باشد و اطلاعات ارسال یا دریافت کند. برای این کار برنامه باید مجوز «android.permission.BLUETOOTH» را از کاربر بگیرد. برای جست‌وجو و دیدن لیست دستگاه‌های هماهنگ شده هم برنامه به مجوز «android.permission.BLUETOOTH_ADMIN» نیاز دارد. از آنجایی که استفاده از بلوتوث و اتصال به یک دستگاه ممکن است موقعیت مکانی کاربر را افشا کند، برنامه به مجوز «android.permission.ACCESS_COARSE_LOCATION» هم نیاز دارد. قبل از شروع کار باید این مجوزها را از کاربر بگیرید.

پخش فیلم

نحوه تعریف مجوزهای دسترسی در اندروید

**بررسی روشن یا خاموش بودن بلوتوث:** برای بررسی وضعیت روشن یا خاموش بودن بلوتوث دستگاه تلفن همراه، کافی است متد «getDefaultAdapter» از کلاس «BluetoothAdapter» را صدا بزنیم. این متد یک شیء از نوع «BluetoothAdapter» به ما می‌دهد. برای چک کردن فعال یا غیر فعال بودن بلوتوث دستگاه به شکل زیر عمل می‌کنیم:

```
1  BluethtoothAdapter adapter = BluethtoothAdapter.getdefaultAdapter ( );
2  boolean enabled = adapter .isEnabled ( );
```

اگر بلوتوث دستگاه خاموش باشد باید کاربر را به صفحه تنظیمات هدایت کنیم تا بلوتوث خود را روشن کند.

پخش فیلم

بررسی روشن یا خاموش بودن بلوتوث

**جست‌وجو برای دستگاه‌های بلوتوث:** بعد از اخذ مجوزهای لازم از کاربر، و اطمینان از روشن بودن بلوتوث، حالا نوبت آن است که دستگاه بلوتوثی که می‌خواهیم به آن متصل شویم را پیدا کنیم. اگر پیش از این دو دستگاه به این دستگاه هماهنگ شده باشند کافی است لیست دستگاه‌های هماهنگ شده را از سیستم بگیریم و در میان آنها جست‌وجو کنیم:

```
1  Set< BluethtoothDevice> pairedDevices = adapter .getBondedDevices ();|
```

اما اگر دستگاه مورد نظر در این لیست نباشد، باید جست‌وجو برای دستگاه‌های بلوتوث روشن و قابل کشف در محدوده را شروع کنیم:

```
1  adapter.startDiscovery ( );
```

این متد شروع به جست‌وجو برای دستگاه‌های بلوتوث می‌کند و هر بار که یک دستگاه را پیدا می‌کند به برنامه اطلاع می‌دهد.

پخش فیلم

جست‌وجو برای دستگاه‌های بلوتوث

**اتصال به دستگاه بلوتوث:** یک اتصال بلوتوثی مثل همه اتصال‌های دیگر کار می‌کند. در این ارتباط یک طرف سرور یا کارگزار و طرف دیگر مشتری (client) است. ارتباط بین کارگزار و مشتری با پروتکل RFCOMM برقرار می‌شود که در اندروید با کلاس BluetoothSocket پیاده‌سازی شده است. تقریباً تمام چیزهایی که برای برقراری ارتباط به آنها نیاز داریم در اندروید پیاده‌سازی شده است.

اتصال به عنوان مشتری یا کلاینت اصول ساده‌ای دارد: ابتدا از دستگاه بلوتوثی که می‌خواهید به آن متصل

شوید یک سوکت RFCOMM می‌گیرید و یک شناسه UUID (شناسه ۱۲۸ بیتی یکتا) به دستگاه می‌فرستید:

```
1  UUID uuid = UUID.randomUUID( );
2  BluethtoothDevice device// = ...
3  BluethtoothSocket socket = device.createRfcommSocket ToServiceRecord (uuid);
4   socket.connect ( );|
```

به‌دلیل اینکه فرایند جست‌وجو و اتصال به دستگاه بلوتوث، فرایند سخت‌افزاری پیچیده‌ای است و ممکن است باعث اختلال در عملکرد برنامه شود، کل این فرایند را در thread جداگانه اجرا می‌کنیم.

پخش فیلم

آموزش اتصال به دستگاه بلوتوث

**ارسال دستور به دستگاه بلوتوث:** بعد از اینکه روشن بودن بلوتوث را تست کردیم و لیست دستگاه‌های مجاور را پیدا کردیم و به دستگاه مورد نظر وصل شدیم، نوبت آن است که فرمان‌های مورد نظر را به دستگاه بفرستیم. برای این کار از سوکت RFCOMM که در مرحله قبل از دستگاه بلوتوث گرفتیم استفاده می‌کنیم. ابتدا یک شیء جریان خروجی یا OutputStream از سوکت می‌گیریم و سپس فرمان مورد نظر را در این جریان خروجی می‌نویسیم. برای مثال اگر بخواهیم فرمان A0 را به دستگاه بفرستیم به شکل زیر عمل می‌کنیم:

```
1  OutputStream output = socket.getoutputStream ();
2  String command =“A0”;
3   byte[] bytes = command.getBytes();
4  output.write(bytes);|
```

البته کل این فرایند را باید در یک Thread جدا انجام بدهیم تا مانع عملکرد عادی برنامه نشود.

پخش فیلم

آموزش ارسال دستور به دستگاه بلوتوث

فعالیت کارگاهی

پروژه: برنامه‌ای بنویسید که به تخت بیمارستان متصل شده و فرمان‌های مختلف برای کنترل آن را ارسال کند.

| عنوان پودمان | تکالیف عملکردی (واحدهای یادگیری) | استاندارد عملکرد (کیفیت) | نتایج مورد انتظار | شاخص تحقق | نمره |
|---|---|---|---|---|---|
| پودمان ۵: برنامه‌نویسی اندروید | ۱ـ تحلیل مفاهیم پایه زبان برنامه نویسی برای سیستم عامل اندروید | بررسی و تحلیل سیستم عامل اندروید و برنامه‌نویسی به زبان جاوا برای کنترل دستگاه‌های جانبی توسط تلفن همراه | بالاتر از حد انتظار | ایجاد برنامه ساده برای محاسبات، تعیین انواع داده‌ها و عملگرها، استفاده از ساختارهای کنترلی، استفاده از حلقه‌های تکرار، استفاده از ساختارهای تکرار تودرتو، استفاده از دستورات شرطی، استفاده از متغیرهای آرایه‌ای، اجرای برنامه‌های اندروید، استفاده از انواع چیدمان اشیا در برنامه، استفاده از بلوتوث و ارسال دستور و ارتباط آن با برنامه | ۳ |
| | ۲ـ ایجاد برنامه به زبان جاوا و کنترل سیستم‌های جانبی توسط آن از طریق ارتباط بلوتوث | | درحد انتظار | ایجاد برنامه ساده برای محاسبات، تعیین انواع داده‌ها و عملگرها، استفاده از ساختارهای کنترلی، استفاده از حلقه‌های تکرار، استفاده از ساختارهای تکرار تودرتو، استفاده از دستورات شرطی، استفاده از متغیرهای آرایه‌ای، اجرای برنامه‌های اندروید | ۲ |
| | | | پایین‌تر از حد انتظار | ایجاد برنامه ساده برای محاسبات، تعیین انواع داده‌ها و عملگرها، استفاده از ساختارهای کنترلی، استفاده از حلقه‌های تکرار، استفاده از ساختارهای تکرار تودرتو، استفاده از دستورات شرطی | ۱ |
| | نمره مستمر از ۵ | | | | |
| | نمره شایستگی پودمان | | | | |
| | نمره پودمان از ۲۰ | | | | |

سازمان پژوهش و برنامه‌ریزی آموزشی جهت ایفای نقش خطیر خود در اجرای سند تحول بنیادین در آموزش و پرورش و برنامه درسی ملی جمهوری اسلامی ایران، مشارکت معلمان را به‌عنوان یک سیاست اجرایی مهم دنبال می‌کند. برای تحقق این امر در اقدامی نوآورانه سامانه تعاملی بر خط اعتبارسنجی کتاب‌های درسی راه‌اندازی شد تا با دریافت نظرات معلمان دربارهٔ کتاب‌های درسی نونگاشت، کتاب‌های درسی را در اولین سال چاپ، با کمترین اشکال به دانش‌آموزان و معلمان ارجمند تقدیم نماید. در انجام مطلوب این فرایند، همکاران گروه تحلیل محتوای آموزشی و پرورشی استان‌ها، گروه‌های آموزشی و دبیرخانهٔ راهبری دروس و مدیریت محترم پروژه آقای محسن باهو نقش سازنده‌ای را بر عهده داشتند. ضمن ارج نهادن به تلاش تمامی این همکاران، اسامی دبیران و هنرآموزانی که تلاش مضاعفی را در این زمینه داشته و با ارائهٔ نظرات خود سازمان را در بهبود محتوای این کتاب یاری کرده‌اند به شرح زیر اعلام می‌شود.

**اسامی دبیران و هنرآموزان شرکت کننده در اعتبارسنجی کتاب دانش فنی تخصصی رشته مکاترونیک ـ کد ۲۱۲۴۷۷**

| ردیف | نام و نام خانوادگی | استان محل خدمت | ردیف | نام و نام خانوادگی | استان محل خدمت |
|---|---|---|---|---|---|
| ۱ | مهدی شهروز | خوزستان | ۶ | محمدرضا هاشمی | خراسان رضوی |
| ۲ | محمد چشفر | شهر تهران | ۷ | محمدرضا پایا | اصفهان |
| ۳ | علی ایمانیان | اصفهان | ۸ | گل‌دوست لیاولی | قزوین |
| ۴ | مسعود محمدی چاهکی | قم | ۹ | علی پورشجاع | مرکزی |
| ۵ | بهزاد محسنی | شهر تهران | ۱۰ | محمد افتخاری مقدم | خراسان رضوی |