

## دستور کار کارگاه برنامه‌نویسی پیشرفته

نیم‌سال دوم ۹۷-۹۸

جلسه پنجم

### آشنایی با اشکال‌یابی در کد<sup>۱</sup>

#### مقدمه

در این جلسه قصد داریم تا با مفاهیم رفع اشکال آشنا شویم. همانطور که می‌دانید، محصولات نرم‌افزاری حاصل از کدهایی هستند که توسط یک یا چند توسعه‌گر نوشته شده‌اند. از آنجایی که عمده‌ی این کدها بوسیله‌ی انسان نوشته می‌شود، همواره احتمال خطا در کد وجود دارد. لذا یکی از قسمت‌های مهم در توسعه محصول بحث کنترل خطاها و کیفیت محصول است. یکی از متغیرهای مهم تعیین کیفیت نرم‌افزار، تعداد خطاهای موجود در آن است. هرچه تعداد خطایی که کاربران هنگام کار با نرم‌افزار با آن مواجه می‌شوند افزایش یابد، کیفیت نرم‌افزار کاهش می‌یابد که نارضایتی کاربران را به دنبال خواهد داشت. لذا کنترل تعداد خطاهای نرم‌افزار اهمیت بسیاری در توسعه محصول دارد. کنترل خطا در پروژه بدین گونه است که با به کارگیری روش‌هایی، سعی در کمینه کردن تعداد خطاهای محصول هنگام رسیدن به مشتری را دارد. یکی از فعالیت‌های این فرآیند بحث رفع اشکال کد یا debug کردن آن است. در این فعالیت خطاهای کشف شده موجود در محصول طی روش‌هایی بدست آمده باید از بین بروند.

#### اشکال‌یابی

یکی از فرآیندهای دشوار توسعه محصول، رفع اشکال آن است. فرآیند دیباگ دو مرحله دارد که اولین آن پیدا کردن منشا خطا و بعد رفع اشکال آن است. هر دو مرحله سختی‌های خاص خود را دارند که باعث سختی این فرآیند شده. در مرحله اول از آنجایی که کدهای محصول تا حدی زده شده و قابلیت دوباره نوشتن آن وجود ندارد، پیدا کردن منشا مشکل نیازمند درک بالایی از کد است. لذا اولین قسمت از فرآیند دیباگ کردن این است که توسعه‌گر خود کد مستندات مربوط به آن قسمت کد را در صورت وجود مطالعه کند تا با کلاس‌های پروژه و متدها و متغیرهای کلاس‌ها آشنایی پیدا کند.

بعد از مطالعه‌ی کد و مستندات آن باید به کمک ابزارهای اشکال‌یابی، منشا خطا را پیدا کرد. منشا خطا در کد جایی است که برنامه هنگام اجرا از روند منطقی خود خارج می‌شود. لذا در گام بعدی

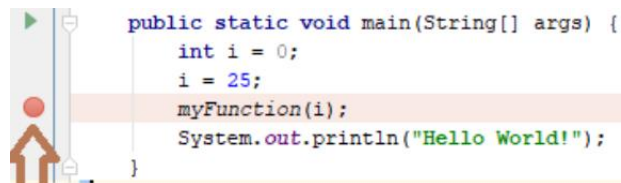
<sup>۱</sup>Debugging

باید کد را اجرا کنید و نتیجه‌ی هر قسمت از کد را با نتیجه مورد نظر خود مقایسه کنید. نرم‌افزار IntelliJ به برنامه‌نویسان این امکان را می‌دهد تا بتوانند برنامه خود را به صورت خط به خط اجرا کرده و مقدار تمامی متغیرهای موجود را در هر لحظه مشاهده کنند. برای استفاده از این امکان، باید یک نقطه وقفه (breakpoint) در برنامه قرار داده شود. پس از قراردادن این نقطه در برنامه، با اجرای برنامه در حالت اشکالیابی می‌توان از امکان اجرای خط به خط برنامه و مشاهده مقادیر متغیرها استفاده نمود.

برنامه زیر را در نظر بگیرید. در این برنامه، تابعی تعریف شده است که مقدار ورودی را یک واحد افزایش می‌دهد و سپس مقدار جدید را به همراه یک رشته کاراکتر چاپ میکند.

```
public class Main {
    public static void myFunction(int i) {
        i++;
        System.out.println("The variable you passed was " + i);
    }
    public static void main(String[] args) {
        int i = 0; i = 25;
        myFunction(i);
        System.out.println("Hello World!");
    }
}
```

فرض کنید می‌خواهیم در کنار تابع myFunction یک نقطه وقفه قرار دهیم. برای قراردادن نقطه وقفه کافی است تا ناحیه مشخص شده در شکل بعد را فشار دهید تا یک علامت دایره قرمز نشان داده شود.



حال برای اجرای برنامه در حالت اشکالیابی کافی است مطابق با شکل زیر، در قسمت بالا و سمت راست صفحه کلیک کنید.



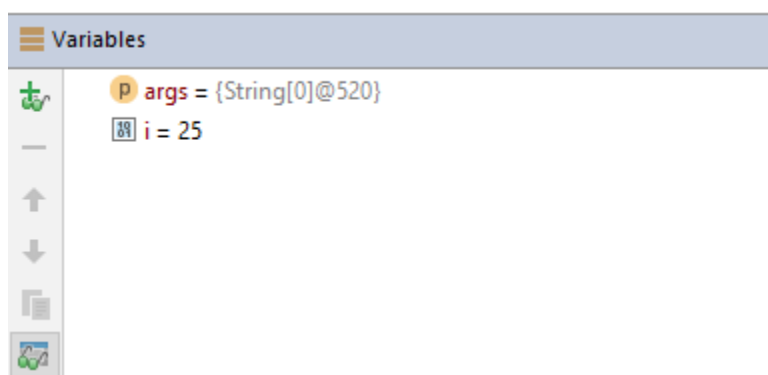
با کلیک کردن بر روی این دکمه، برنامه در حالت debugging اجرا میشود و بر روی خط دارای نقطه وقفه توقف میکند و صفحه Editor مقدار تمامی متغیرهای موجود در صفحه را نشان خواهد داد.



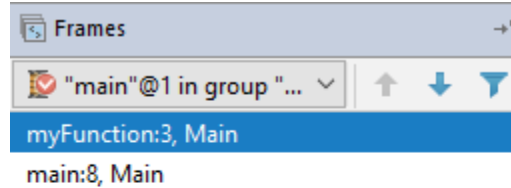
در ادامه، در جدولی به معرفی هر یک از قسمت‌های نوار ابزار اشکالیابی می‌پردازیم.

کلید میان‌بر	کارکرد	دستور موجود در نوار ابزار
Alt + F10	خط فعلی اجرای دستورات را نشان دهد.	Show Execution Point
F8	برنامه را تا خط بعدی اجرا کرده و توابع بینابینی را رد می‌کند.	Step Over
F7	به داخل تابعی که در حال حاضر Debugger روی آن قرار دارد می‌رود.	Step Into
Shift + Alt + F7	دستور Step Into بعضی اوقات دستورات مربوط به خود SDK را وارد نمی‌شود. با این کار شما Debugger را مجبور به ورود به این دستورات خواهید کرد.	Force Step Into
Shift + F8	با اجرای این دستور Debugger از تابع فعلی خارج خواهد شد و به خط بعدی نقطه فراخوانی آن خواهد رفت.	Step Out
Alt + F9	تا اجرای دستور در نقطه‌ی مکان‌نما در Editor پیش خواهد رفت.	Run to Cursor

در قسمت variables می‌توانید متغیرهای ایجاد شده در کد به همراه مقدار کنونی آن‌ها را مشاهده کنید.



در قسمت frames نیز می‌توانید ترتیب متدهای وارد شده را نیز مشاهده کنید.



## انجام دهید

فایل Lab5.zip را دانلود کنید. یک پروژه جدید ساخته و فایل‌ها را کپی کنید. این کدها پیاده‌سازی محیط یک رستوران در جاوا هست که به عنوان تمرین به دانشجویان ترم پیش داده شده. توضیح کلاس‌ها و متدهای مربوط به هر کلاس در فایل Description.pdf داده شده است.

در فایل Test.java یک سناریو از رستوران را پیاده‌سازی کردیم. در این سناریو ابتدا چندین ماده اولیه ساختیم. سپس برای رستوران یک غذا تعریف کردیم که از تعدادی مواد اولیه تهیه شده بود. سپس غذاها را به رستوران اضافه کردیم.

در ادامه یک مشتری به رستوران خود اضافه کردیم. سپس برای او یک درخواست غذا ایجاد کردیم (Order). در ابتدا مشتری دو غذای f1 سفارش داده. سپس تصمیم می‌گیرد که یک غذای f2 نیز به غذاهای خود اضافه کند.

بعد از آنکه مشتری با رستوران تصفیه حساب کرد (به وسیله متد restaurant.settle) مقدار fund رستوران به درستی چاپ نشده. با استفاده از روش debug در IntelliJ مقدار درست fund مشتری را بعد از تصویه حساب محاسبه کنید.