

Debugowanie

Marcin Kwiatkowski

Agenda

- Czym jest debugowanie?
- Znaczenie bugów
- Debugowanie w IntelliJ
- Zadania, zadania...

Debugowanie



Debugowanie to *proces zmniejszania liczby błędów przez śledzenie działania aplikacji krok po kroku.*

“Everyone knows that debugging is twice as hard as writing a program in the first place. So if you're as clever as you can be when you write it, how will you ever debug it?

Brian Kernighan
(współautor książki Język ANSI C)

“ People also underestimate the time they spend debugging. They underestimate how much time they can spend chasing a long bug. With testing, I know straight away when I added a bug. That lets me fix the bug immediately, before it can crawl off and hide. There are few things more frustrating or time wasting than debugging. Wouldn't it be a hell of a lot quicker if we just didn't create the bugs in the first place? ”

Martin Fowler

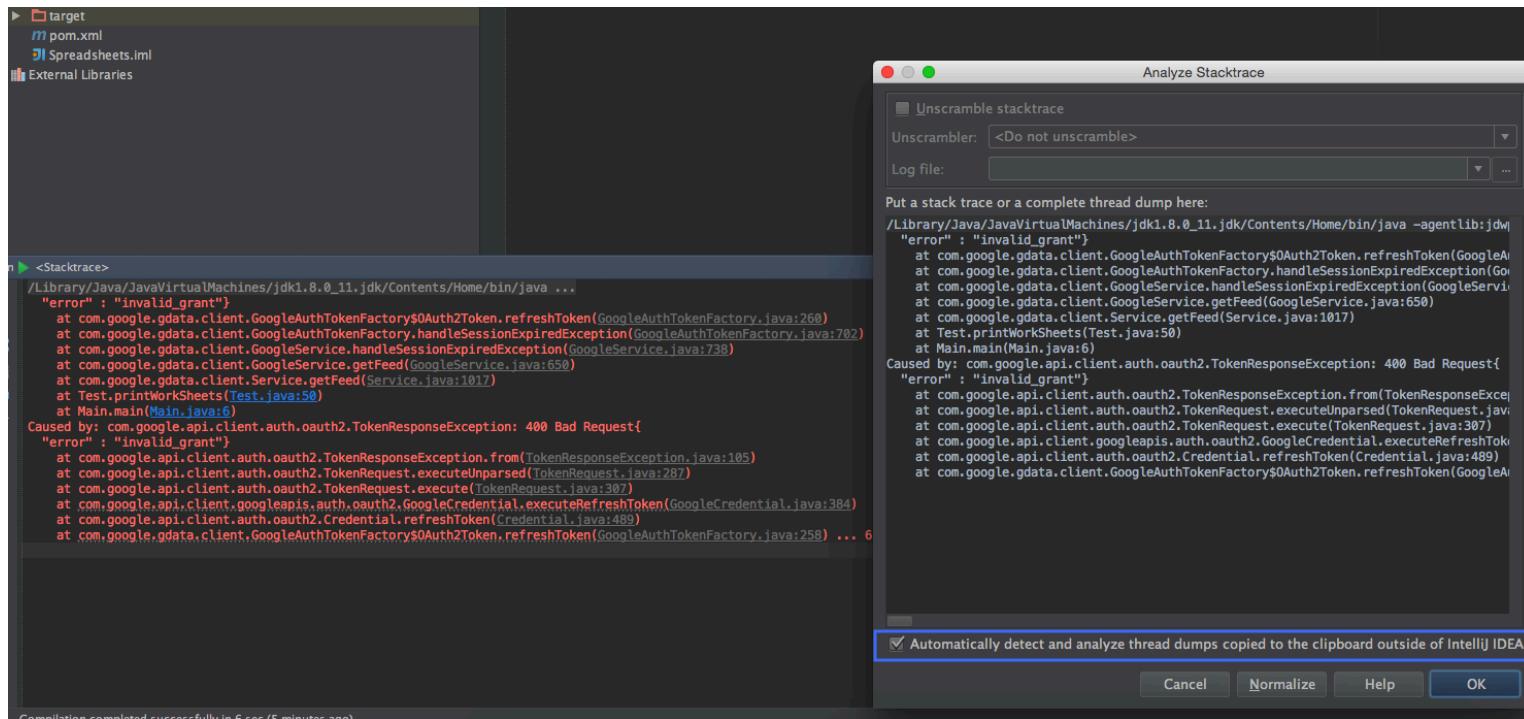
Znaczenie bugów



Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

If you'd like to know more, you can search online later for this error:
SYSTEM_THREAD_EXCEPTION_NOT_HANDLED (nvlddmkm.sys)

Znaczenie bugów



The image shows a Java development environment with two windows open. On the left, the IntelliJ IDEA interface displays a project structure with files like pom.xml and Spreadsheets.iml, and an 'External Libraries' section. Below this is a terminal window showing a stacktrace:

```

target
m pom.xml
3 Spreadsheets.iml
External Libraries

<Stacktrace>
/Library/Java/JavaVirtualMachines/jdk1.8.0_11.jdk/Contents/Home/bin/java ...
"error" : "invalid_grant"
at com.google.gdata.client.GoogleAuthTokenFactory$OAuth2Token.refreshToken(googleAuthTokenFactory.java:260)
at com.google.gdata.client.GoogleAuthTokenFactory.handleSessionExpiredException(GoogleAuthTokenFactory.java:702)
at com.google.gdata.client.GoogleService.handleSessionExpiredException(GoogleService.java:738)
at com.google.gdata.client.GoogleService.getFeed(GoogleService.java:650)
at com.google.gdata.client.Service.getFeed(Service.java:1017)
at Test.printWorkSheets(Test.java:50)
at Main.main(Main.java:6)
Caused by: com.google.api.client.auth.oauth2.TokenResponseException: 400 Bad Request{
"error" : "invalid_grant"
at com.google.api.client.auth.oauth2.TokenResponseException.from(TokenResponseException.java:105)
at com.google.api.client.auth.oauth2.TokenRequest.executeUnparsed(TokenRequest.java:287)
at com.google.api.client.auth.oauth2.TokenRequest.execute(TokenRequest.java:307)
at com.google.api.client.googleapis.auth.oauth2.GoogleCredential.executeRefreshToken(GoogleCredential.java:384)
at com.google.api.client.auth.oauth2.Credential.refreshToken(Credential.java:489)
at com.google.gdata.client.GoogleAuthTokenFactory$OAuth2Token.refreshToken(GoogleAuthTokenFactory.java:259) ... 6

```

At the bottom of the terminal window, a message indicates compilation was successful: "Compilation completed successfully in 6 sec (5 minutes ago)".

On the right, a separate window titled "Analyze Stacktrace" is open. It contains fields for "Unscrambler" (set to "<Do not unscramble>") and "Log file". Below these is a text area with the same stacktrace from the terminal. At the bottom of this window is a checkbox labeled "Automatically detect and analyze thread dumps copied to the clipboard outside of IntelliJ IDEA" which is checked. There are also buttons for "Cancel", "Normalize", "Help", and "OK".

Znaczenie bugów

- **Katastrofa Ariane 5 (4.06.1996 r.)**
- 64-bitowa liczba zmiennoprzecinkowa konwertowana na 16-bitowy integer
- Konwersja się nie udała, bo liczba zmiennoprzecinkowa była spoza zakresu 16-bitowego



Znaczenie bugów



Znaczenie bugów



Zanim zaczniemy...



Debugowanie w IntelliJ

Część praktyczna

Przygotowanie do zajęć

- Ściągnij pliki z repozytorium
- Utwórz swojego brancha według wzoru: *imie-nazwisko*
- Pamiętaj, aby *commitować* wprowadzane podczas ćwiczeń zmiany i poprawki w projektach.
- Nie zapominaj o odpowiednich opisach przy każdym *commicie!*

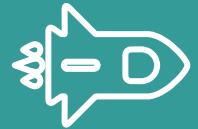
Breakpoint

Breakpoint określa miejsce, w którym aplikacja się zatrzyma oraz umożliwia podejrzenie wartości zmiennych.

```
// Read number b from standard input
System.out.print("b = ");
int b = scanner.nextInt();  b: 3  scanner: "java.util.Scannr

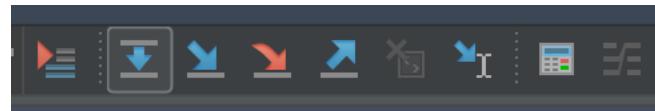
// Output a + b
int c = a + b;  a: 4  b: 3
System.out.println("a + b = " + c);
```

Zadanie 1



- Otwórz projekt 01-scanner
- Postaw breakpoint i uruchom aplikację w trybie debug

Sterowanie aplikacją

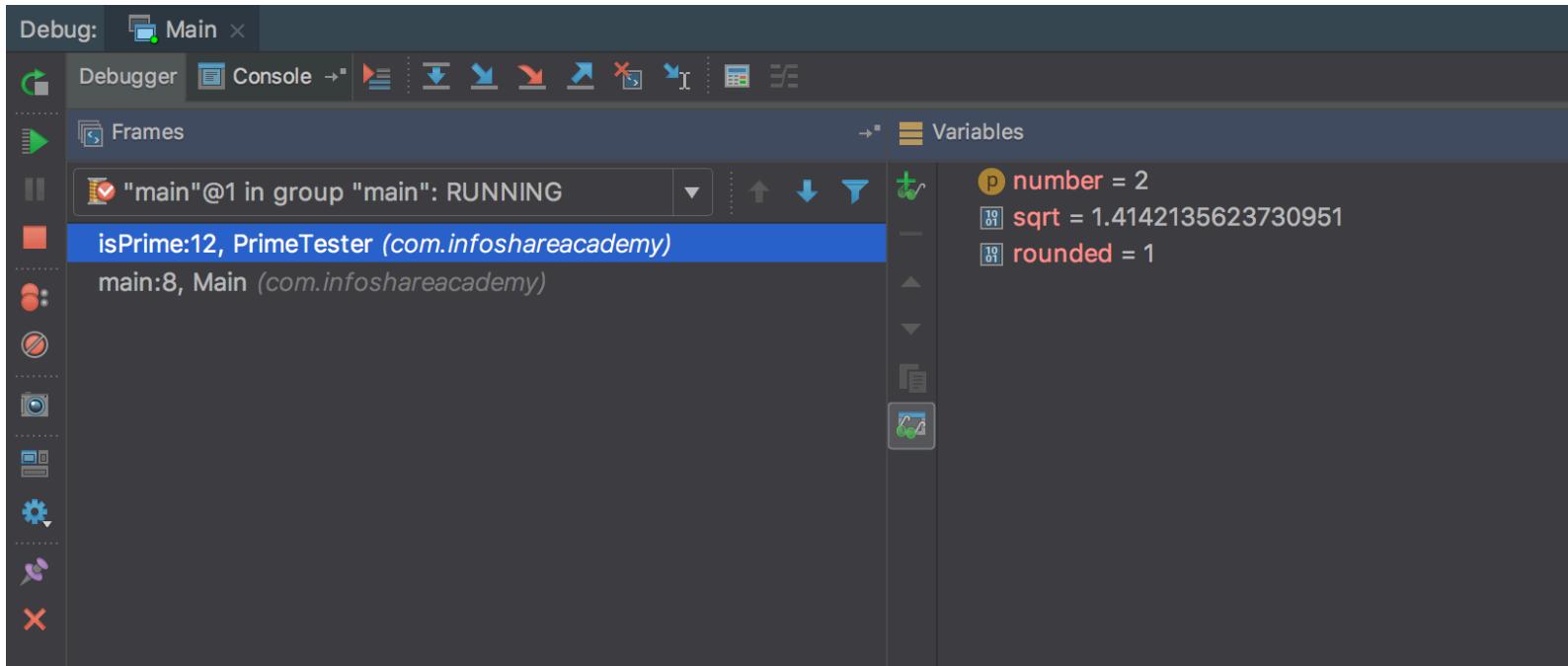


Step over - przejście do następnej linii kodu (bez wchodzenia do środka funkcji)

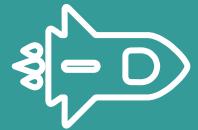
Step into - wejście do środka funkcji, na której zatrzymany jest breakpoint

Run to cursor - kontynuowanie działania aplikacji, zatrzymanie na zaznaczonej linii w edytorze

Podglądanie zmiennych

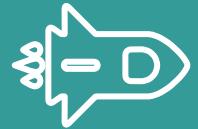


Zadanie 2



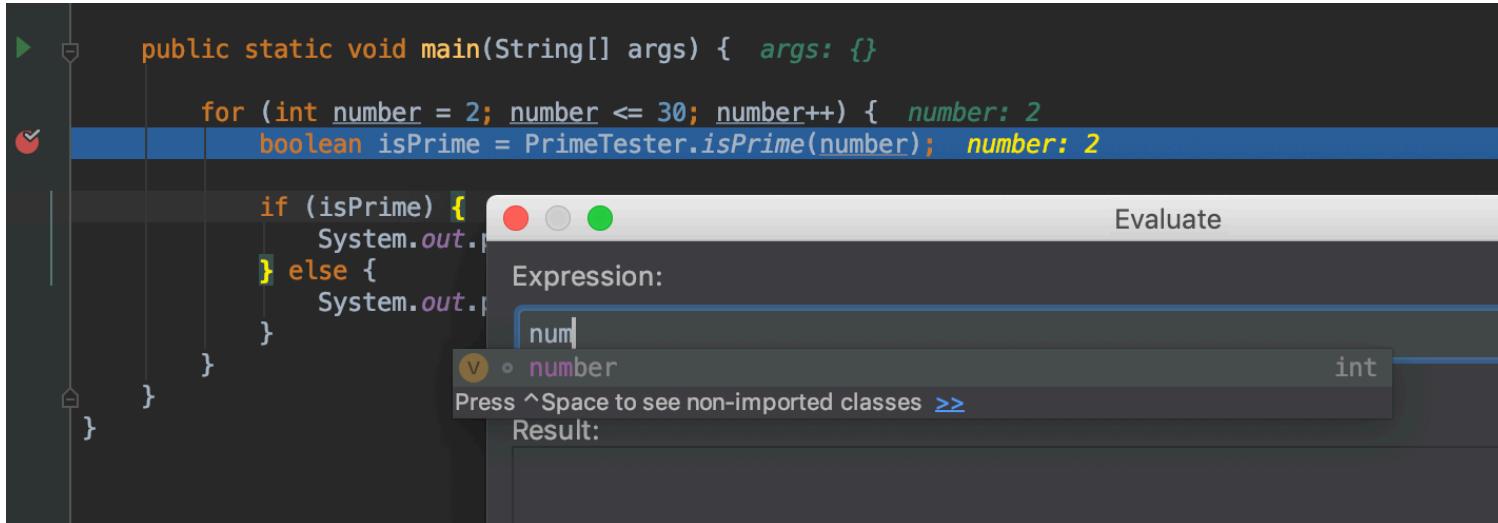
- Otwórz projekt 02-fibonacci
- Program ma wypisać n liczb Fibonacciego

Zadanie 3



- Otwórz projekt 03-prime-test
- Program ma wypisywać liczby od 2 do 30 i określać czy są liczbami pierwszymi

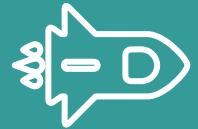
Opcja Evaluate Expression



The screenshot shows a Java code editor with a debugger interface. The code is a simple program that prints prime numbers from 2 to 30. A break point is set at the line where the variable `number` is assigned the value 2. A tooltip window titled "Evaluate" is open, showing the expression `number` with a value of 2. The tooltip also includes a dropdown menu with the variable `number` selected, and a status message at the bottom.

```
public static void main(String[] args) { args: {}  
    for (int number = 2; number <= 30; number++) { number: 2  
        boolean isPrime = PrimeTester.isPrime(number); number: 2  
  
        if (isPrime) {  
            System.out.println(number);  
        } else {  
            System.out.println("Not prime");  
        }  
    }  
}
```

Zadanie 4



- Otwórz projekt 04-json
- Program ma odczytać plik JSON z listą osób, a następnie znaleźć i wypisać adresy osób, które są w domenie .pl

Rekurencja

Rekurencja to odwoływanie się funkcji do samej siebie

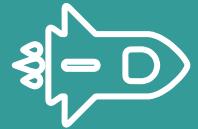
```
public static void main(String args[])
{
    .....
    .....
    recursiveMethod(); Normal Method Call
    .....
    .....
}

static void recursiveMethod() Recursive Method Call
{
    .....
    // recursive call
    recursiveMethod();
    .....
}
```



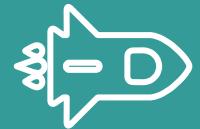
Żeby zrozumieć rekurencję
musisz najpierw zrozumieć rekurencję.

Zadanie 5



- Otwórz projekt 05-recursion
- Doprowadź program do działania.

Zadanie 6



Ustaw swój projekt, który tworzysz w ramach infoShare Academy w taki sposób, aby możliwe było jego debugowanie.



Dzięki

LinkedIn: <https://bit.ly/2Bzapg3>

- Trener [infoShare Academy](#)
- Full Stack Developer w [Bright Inventions](#)