

Matlab 2012b

Narzędzia programistyczne

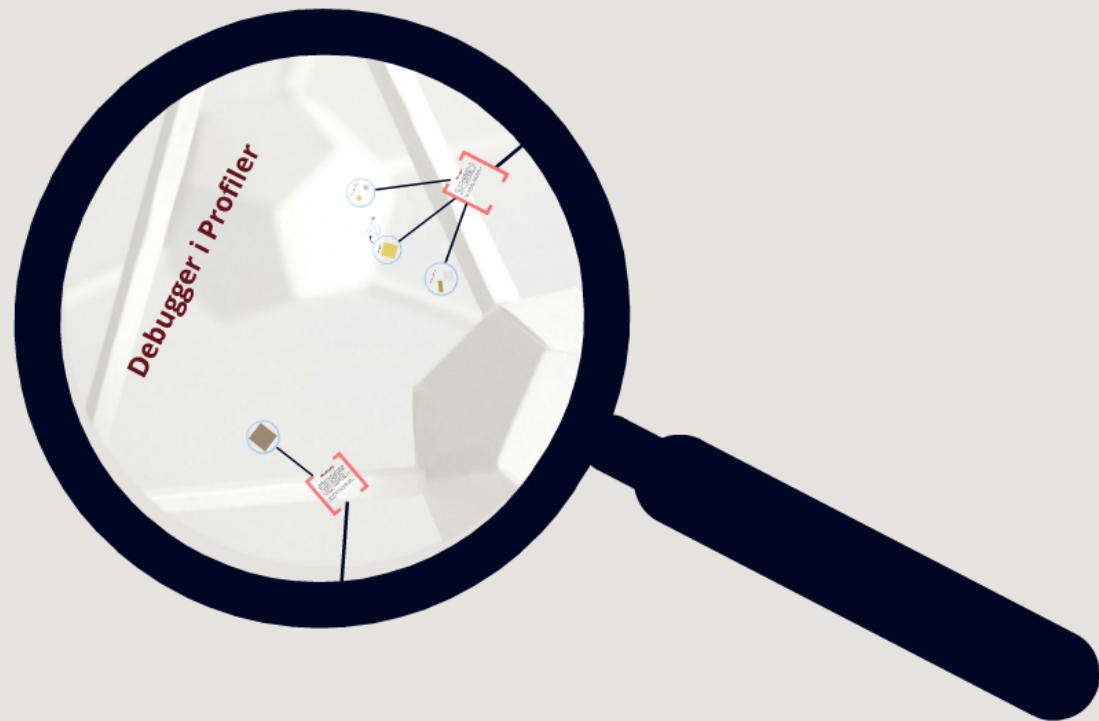
Dawid Mitoń



Matlab 2012b

Narzędzia programistyczne

Dawid Mitoń



Debugger i Profiler



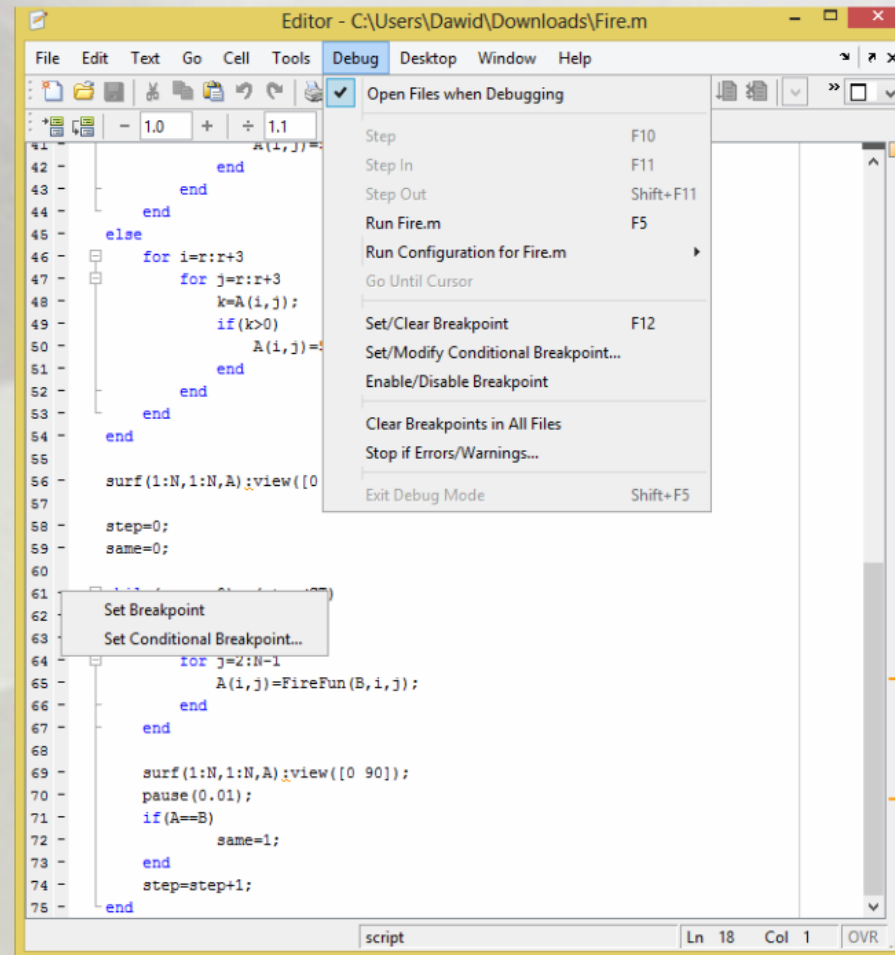
Debugger

Debug tool, Debugger (czytaj debager - z ang. odpluskwiacz) – program komputerowy służy do dynamicznej analizy innych programów, w celu odnalezienia i identyfikacji zawartych w nich błędów, zwanych z angielskiego bugami (robakami).

Proces nadzorowania wykonania programu za pomocą debuggera określa się mianem debugowania.

Sposoby debugowania

Graficzny interfejs użytkownika



Editor - C:\Users\Dawid\Downloads\Fire.m

File Edit Text Go Cell Tools **Debug** Desktop Window Help

Open Files when Debugging

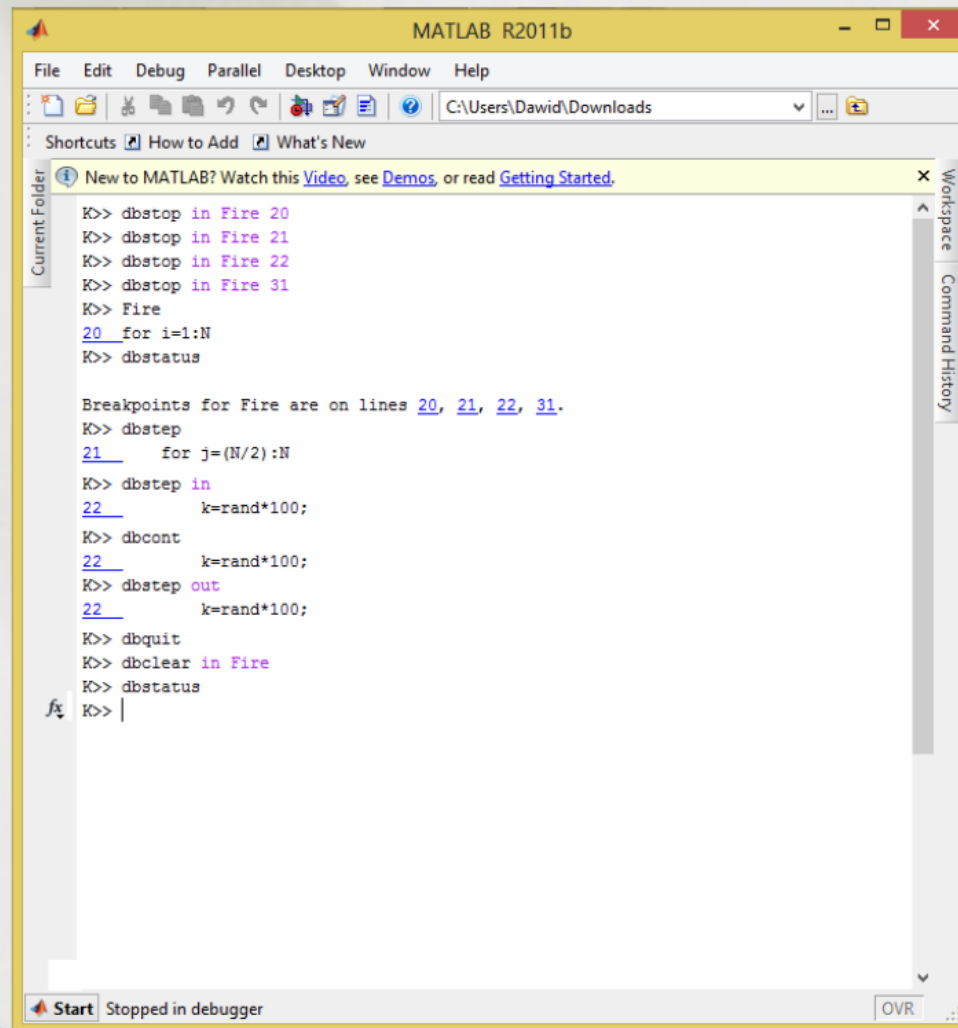
Step F10
Step In F11
Step Out Shift+F11
Run Fire.m F5
Run Configuration for Fire.m
Go Until Cursor
Set/Clear Breakpoint F12
Set/Modify Conditional Breakpoint...
Enable/Disable Breakpoint
Clear Breakpoints in All Files
Stop if Errors/Warnings...
Exit Debug Mode Shift+F5

```
41 - A(1,j)=  
42 -     end  
43 - end  
44 - end  
45 - else  
46 -     for i=r:r+3  
47 -         for j=r:r+3  
48 -             k=A(i,j);  
49 -             if(k>0)  
50 -                 A(i,j)=  
51 -             end  
52 -         end  
53 -     end  
54 - end  
55 -  
56 - surf(1:N,1:N,A):view([0  
57 -  
58 -     step=0;  
59 -     same=0;  
60 -  
61 -  
62 -  
63 -  
64 -     for j=2:N-1  
65 -         A(i,j)=FireFun(B,i,j);  
66 -     end  
67 - end  
68 -  
69 -     surf(1:N,1:N,A):view([0 90]);  
70 -     pause(0.01);  
71 -     if(A==B)  
72 -         same=1;  
73 -     end  
74 -     step=step+1;  
75 - end
```

Set Breakpoint
Set Conditional Breakpoint...

script Ln 18 Col 1 OVR

Funkcje debugera z okna poleceń

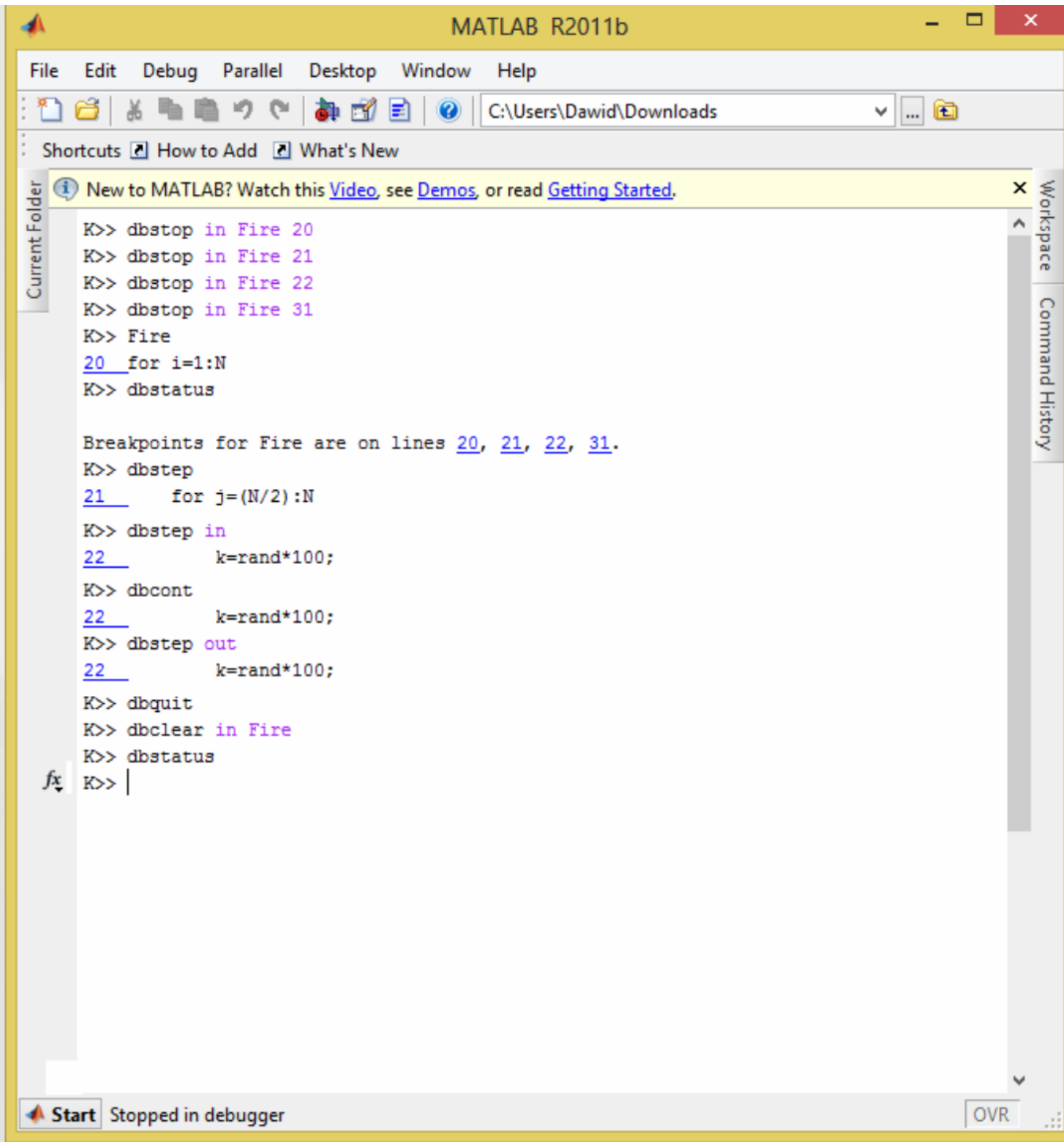


The image shows the MATLAB R2011b Command Window interface. The window title is "MATLAB R2011b". The menu bar includes File, Edit, Debug, Parallel, Desktop, Window, and Help. The address bar shows the current directory as "C:\Users\Dawid\Downloads". The Command Window displays the following commands and output:

```
K>> dbstop in Fire 20
K>> dbstop in Fire 21
K>> dbstop in Fire 22
K>> dbstop in Fire 31
K>> Fire
20 for i=1:N
K>> dbstatus

Breakpoints for Fire are on lines 20, 21, 22, 31.
K>> dbstep
21 for j=(N/2):N
K>> dbstep in
22 k=rand*100;
K>> dbcont
22 k=rand*100;
K>> dbstep out
22 k=rand*100;
K>> dbquit
K>> dbclear in Fire
K>> dbstatus
K>> |
```

The status bar at the bottom indicates "Start Stopped in debugger" and "OVR".



Breakpoints

Editor - C:\Users\Dawid\Downloads\Fire.m

—

Editor - C:\Users\Dawid\Downloads\Fire.m

File Edit Text Go Cell Tools Debug Desktop Window Help

1 - ST=1000;
2 - N=100;
3 - A=zeros(N,N);
4 -
5 - %PROPORCJE LIŚCIASTE:IGLASTE:PUSTE 60:30:10
6 - for i=1:N
7 - for j=1:(N/2);
8 - k=rand*100;
9 - if(k<=60)
10 - A(i,j)=10;
11 - elseif(k>60 && k<=90)
12 - A(i,j)=20;
13 - elseif(k>90 && k<=100)
14 - A(i,j)=0;
15 - end
16 - end
17 - end
18 -
19 - %PROPORCJE LIŚCIASTE:IGLASTE:PUSTE 30:60:10
20 - for i=1:N
21 - for j=(N/2):N
22 - k=rand*100;
23 - if(k<=60)
24 - A(i,j)=20;
25 - elseif(k>60 && k<=90)
26 - A(i,j)=10;
27 - elseif(k>90 && k<=100)
28 - A(i,j)=0;
29 - end
30 - end
31 - end
32 -
33 - %GENEROWANIE ZAPŁONU
34 - r=round(rand*100);
35 - if(r>(N/2))

script Ln 20 Col 1 OVR

19

20

21

22

23



-

3. FOR

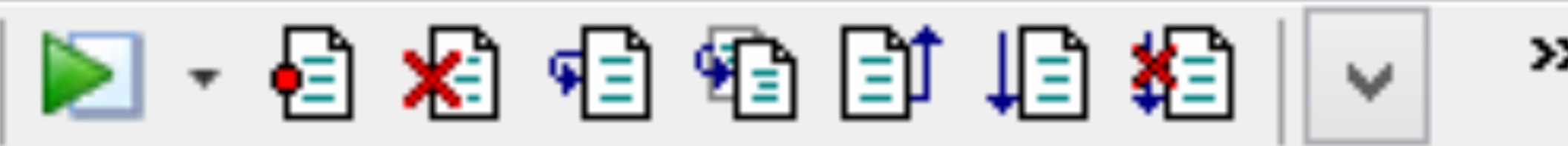


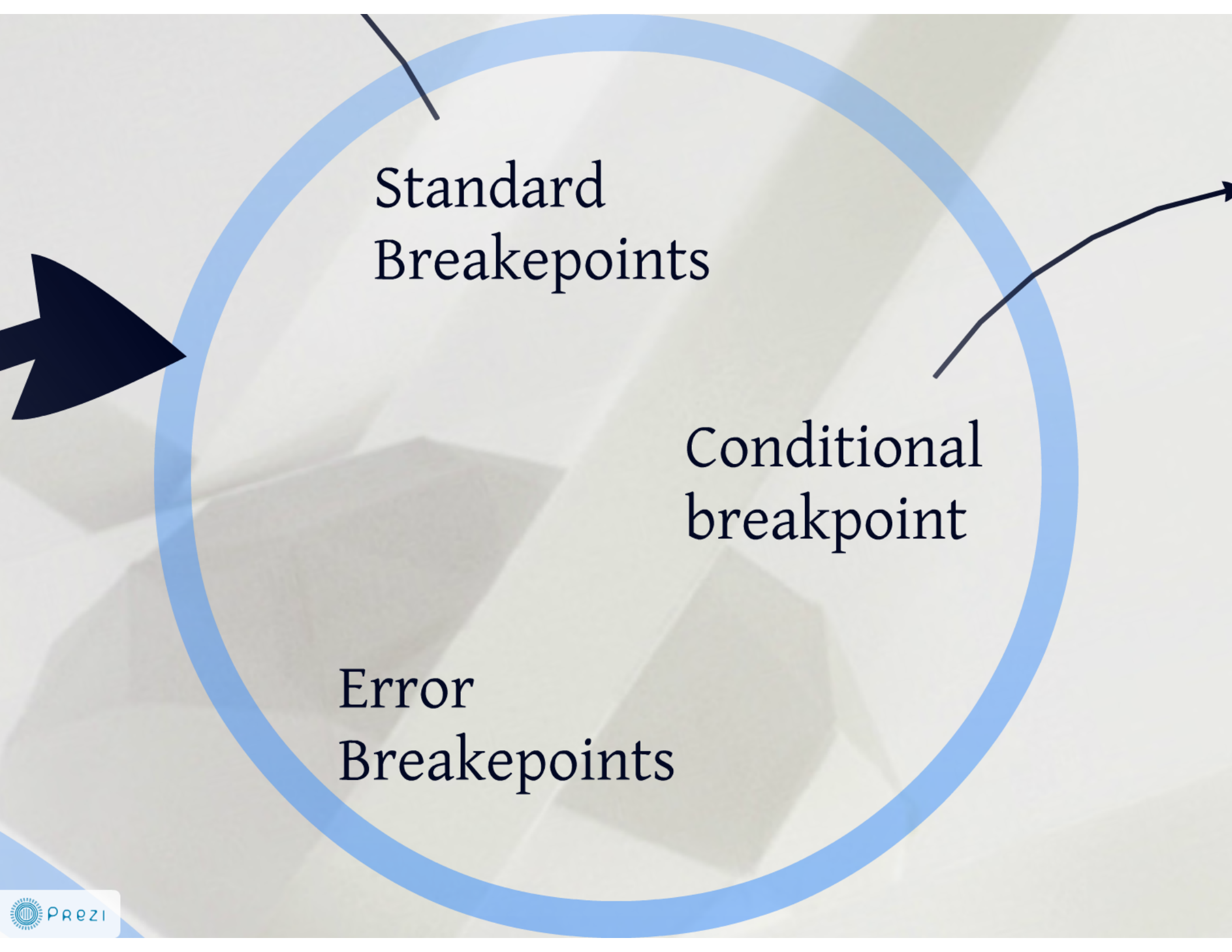
for i

f

downloads\Fire.m

File Edit View Help





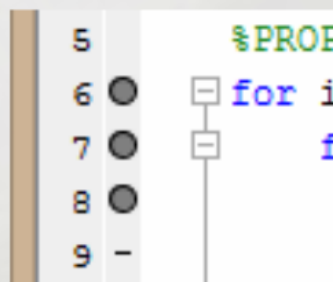
Standard
Breakpoints

Conditional
breakpoint

Error
Breakpoints

Standard Breakpoints

Zwykły, standardowy
punkt przerwania.



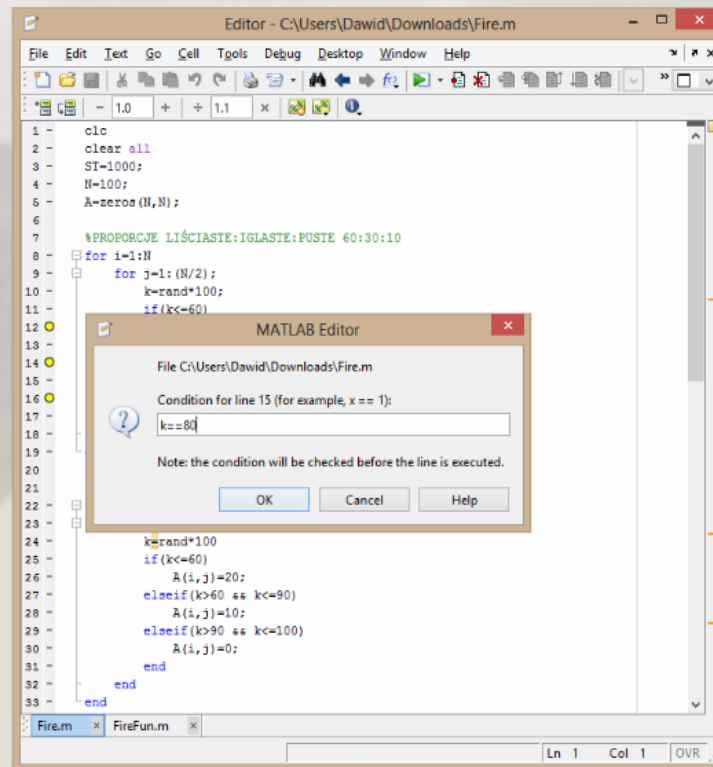
Komendy konsoli:

- dbstop
- dbstep
- dbcont
- dbstatus
- dbquit
- dbc clear



Conditional breakpoint

Warunkowy punkt przerwania



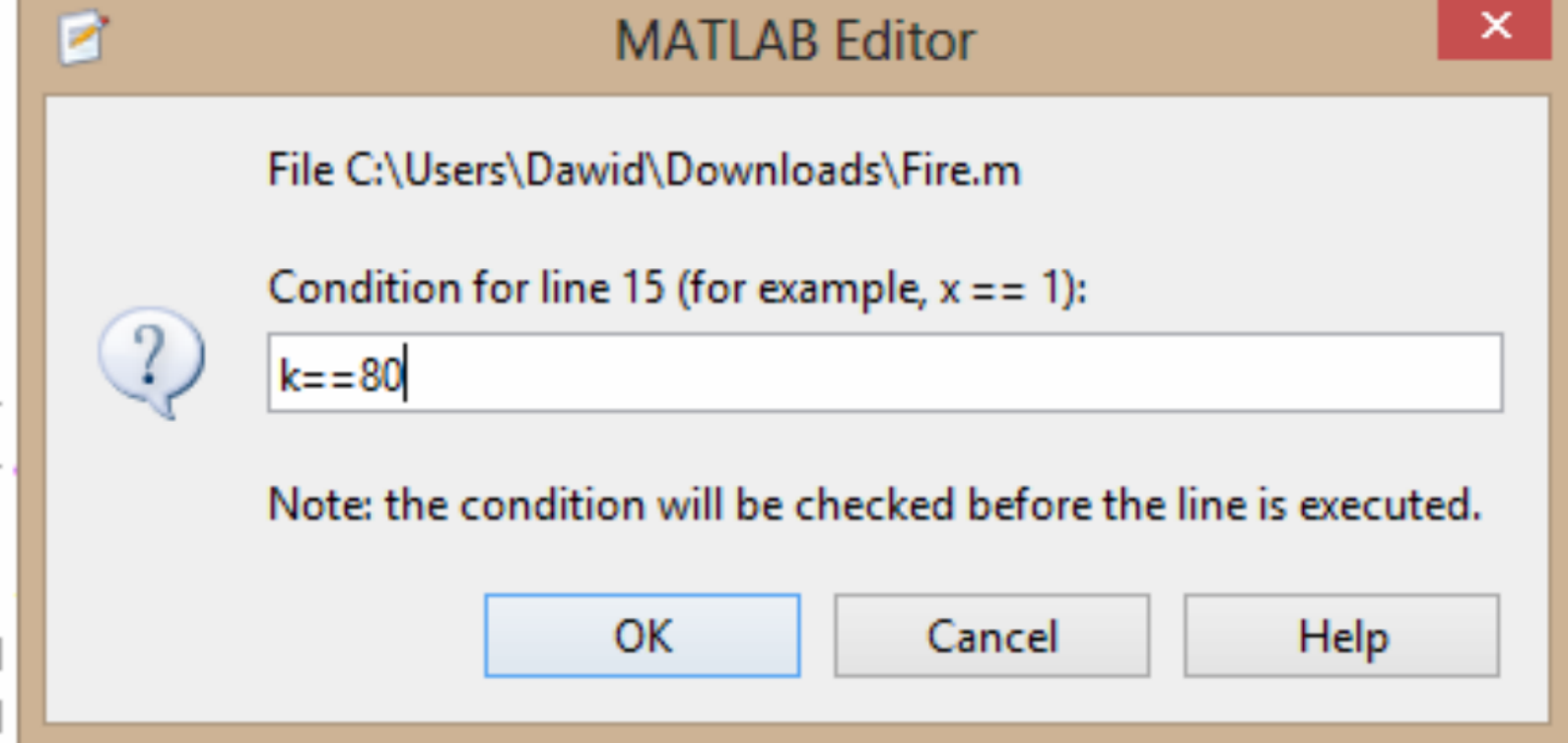
```
A=zeros(N,N);  
%PROPORCJE LIŚCIASTE:IGLASTE:PUSTE 60:30:10
```

```
for i=1:N
```

```
    for j=1:(N/2);
```

```
        k=rand*100;
```

```
        if(k<=60)
```



```
        k=rand*100
```

```
        if(k<=60)
```

```
            A(i,j)=20;
```

```
        elseif(k>60 && k<=90)
```

```
            A(i,j)=10;
```

```
        elseif(k>90 && k<=100)
```

```
            A(i,j)=0;
```

Error Breakpoints

Podgląd zmiennych

```

18
19     %PROPORCJE LIŚCIASTE:IGLASTE:PUSTE 30:60:10
20     for i=1:N
21         for j=(N/2):N
22             j: 1x1 double =
23             50
24             elseif(k>60 && k<=90)
25                 A(i,j)=10;
26             elseif(k>90 && k<=100)
27                 A(i,j)=0;
28             end
29         end
30     end
31 end

```

[mos](#), or read [Gettin](#)

Workspace

 Select data to plot

Name ▲	Value	M
 A	<100x100 double>	0
 B	<100x100 double>	0
 N	100	10
 ST	1000	10
 ans	[]	
 i	1	1
 j	50	50
 k	14.1861	14
 r	20	20
 same	1	1
 step	290	29

Workspace

Command History

PROFILER

W inżynierii oprogramowania, to forma dynamicznej analizy programu (w przeciwieństwie do statycznej analizy kodu). Jest to badanie zachowania programu używając informacji zdobytych podczas jego wykonywania. Zwykle przeprowadza się je by dowiedzieć się, które części programu zoptymalizować, by zwiększyć jego ogólną prędkość lub zmniejszyć wymagania pamięci.

Przedwczesna optymalizacja może niepotrzebnie zwiększyć złożoność kodu bez prawdziwego przyrostu wydajności.

