DrJava's debugger och variabelbevakning

Detta dokument visar hur man kan köra ett program sats för sats med DrJava's debugger och samtidigt bevaka variabelvärden.

Man kommer då kunna se vilka satser som programmet genomlöper och samtidigt få se variabelvärden

Nedanstående program beräknar summan 1+2+3+4+5

```
public class Summera_12345 {
  public static void main (String[] arg) {
    int sum=0;
    for (int i=1; i<=5; i++) {
      sum = sum + i;
    } // Slut loopen
    System.out.println("Summan = " + sum);
  } // Slut mainmetoden
  // Slut klassen
```

Vi skall nu köra programmet så att vi ser vilka satser som genomförs och samtidigt få veta värdet på variabeln sum.

Så här ser det ut DrJava

J Z:\prog1\public_html	\ht14\miniLessons\Autotrace_Debugg\Summera_12345.java						
<u>File E</u> dit <u>T</u> ools <u>P</u> roje	ect <u>D</u> ebugger Language Level <u>H</u> elp						
👫 New 🖾 Open	🗐 Save 🖹 Close 🔏 Cut 🖪 Copy 📋 Paste 💭 Undo						
Summers 123 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 -	<pre>// Ett program som summerar 1+2+3+4+5 mha en loop // Ett litet program som vi testar DrJava's debugger på public class Summera_12345 { public static void main (String[] arg) { int sum=0; for (int i=1; i<=5; i++) { sum = sum + i; } // Slut loopen System.out.println("Summan = " + sum); } // Slut mainmetoden } // Slut klassen</pre>						
Compiler readure TD							
Compiler ready: JDK 6.0_45 from C:\Program Files\Java\jdk1.6.0_45\lib\tools.jar.							
Editing Z:\prog1\public_html\ht14\miniLessons\Autotrace_Debugg\Summer 1:0							

Välj Debug Mode

J Z:\prog1	publi	c_html	\ht:	14\miniLessons\Autotrace_Debugg\Summe	era_12345.java			×
<u>File</u> Edit	<u>T</u> ools	<u>P</u> roje	ect	<u>D</u> ebugger <u>L</u> anguage Level <u>H</u> elp		_		
🕞 New		Open		Debug Mode	Ctrl+Skift+D	Paste	🔵 Undo	
Summera_12	3	1 2 3	1.	Toggle Breakpoint on Current Line Clear All Breakpoints Breakpoints	Ctrl+B Ctrl+Skift+B	op ugger	på	
		4 5 7 8 9	р	Resume Debugger Step Into Step Over Step Out Automatic Trace	F7 F12 F11 Skift+F12 F9			
		22 23						
Interactio	ns	Cons	ole	Compiler Output				
Compiler Files\Jav	read va\jd	y: JD k1.6.	0	5.0_45 from C:\Program 45\lib\tools.jar.	Compiler	; t source	•	×
Editing Z:\pi	rog1\p	ublic_	htn	nl\ht14\miniLessons\Autotrace_Debugg\S	ummer			1:0

Resultatet blir så här

J Z:\prog1\public_htr	nl\ht14\miniLessons\Autotrace_Debugg\Summera_12345.java 📃 💻 💌								
File Edit Tools Pro	ject <u>D</u> ebugger <u>L</u> anguage Level <u>H</u> elp								
🤹 New 🖾 Oper	n 📑 Save 🔀 Close 🐰 Cut 🖾 Copy 📋 Paste 🔊 Undo 🧃								
Summera_122 1 // Ett program som summerar 1+2+3+4+5 mha en loop 2 // Ett litet program som vi testar DrJava's debugger på 3 4 public class Summera_12345 { 5 6 public static void main (String[] arg) { 7 8 int sum=0; 9 for (int i=1; i<=5; i++) { 10 sum = sum + i; 11 } // Slut loopen 12 13 System.out.println("Summan = " + sum); 14									
Stack	Threads Resume X								
Watche (Method									
	Step Into								
	Step Over								
	Step Out								
Current document is o	ut of sync with the debugger and should be recompiled!								
Interactions Cor	sole Compiler Output								
Compiler ready: Files\Java\jdk1.	Compiler ready: JDK 6.0_45 from C:\Program Files\Java\jdk1.6.0_45\lib\tools.jar.								
Toggling Debugger M	ode 1:0								

Väljs Detach Debugger

J Z:\prog1\public_html\ht]	L4\miniLessons\Autotrace_Debugg\Summer	a_12345.java		x
<u>File Edit Tools Project</u>	Debugger Language Level Help		_	
🕄 New 🔂 Open	◆ Debug Mode	Ctrl+Skift+D	Paste 🔉 Undo	
Summera_123 1 /. 2 /. 3	Toggle Breakpoint on Current Line Clear All Breakpoints Breakpoints	Ctrl+B Ctrl+Skift+B	op ugger på	1
4 p 5 6 7 8 9 10 11 12 13 14	Resume Debugger Step Into Step Over Step Out Automatic Trace Detach Debugger , , , State Toopen System.out.println("Summan = "	F7 F12 F11 Skift+F12 F9 + sum);		
Stack T			Resume	×
Watches Method		Line	Automatic Trace	
			Step Into	
			Step Over	
			Step Out	
Current document is out of	sync with the debugger and should be reco	ompiled!		
Interactions Console	Compiler Output			
Compiler ready: JDK 6 Files\Java\jdk1.6.0_4	5.0_45 from C:\Program 45\lib\tools.jar.	Compiler JDK 6.0_4	15 T	×
Toggling Debugger Mode				1:0

Mittendelen av DrJava's fönster frigörs i ett eget fönster

🛃 Debugger	- Z:\prog1\public_html\ht	14\miniLesso	ons\Aut 😐 😐	x
	Stack Threads		Resume	X
Watches	Method Li		Automatic Trace	
<u></u>			Step Into	
			Step Over	
			Step Out	
Current docur	ment is out of sync with th	e debugg		

Modifiera det fönstret så att det ser ut så här

Name Value Type Stack	Automatic Trace
Slack	
11-1	Step Into
Met	Step Over
	Step Out

Markera en sats i mainmetoden. Välj att markera den första satsen

🛿 Z:\prog1\public_html\ht14\miniLessons\Autotrace_Debugg\Summera_12345.java									
<u>File Edit Tools Proje</u>	ect <u>D</u> ebugger <u>L</u> ar	nguage Level	<u>H</u> elp						
👫 New 🖾 Open	📳 Save 🛛	× Close	🔏 Cut	🚹 Сору	📋 Paste	Undo	۲		
<pre>3ummera_123 1 // Ett program som summerar 1+2+3+4+5 mha en loop 2 // Ett litet program som vi testar DrJava's debugger på 3 4 public class Summera_12345 { 5 6 public static void main (String[] arg) { 7 8 int sum=0; 9 for (int i=1; i<=5; i++) { 10 sum = sum + i; 11 } // Slut loopen 12 13 System.out.println("Summan = " + sum); 14 15 } // Slut mainmetoden 16 17 18 19 19 10 10 10 10 10 10 10 10 10 10 10 10 10</pre>									
Interactions Cons	sole Compiler O	utput Break	points						
					P	revious	X		
						Next			
						Disable			
						Go to			
	Remove								
					Re	emove All			
Editing Z:\prog1\public_	_html\ht14\miniLes	sons\Autotrac	e_Debugg\	Summer			8:14		

Högermusklicka och välj Toggle Breakpoint



Satsen markeras med rött

J Z:\prog1	public_html	\ht14\miniLesso	ons\Autotrace	_Debugg\Sum	imera_12345.ja	ava	- 0	x
<u>File Edit</u>	<u>T</u> ools <u>P</u> roje	ct <u>D</u> ebugger	<u>L</u> anguage Le	vel <u>H</u> elp				
👫 New	🔂 Open	📕 Save	💦 Close	🔏 Cut	🖪 Сору	📋 Paste	Undo	•
Summera_123 I // Ett program som summerar 1+2+3+4+5 mha en loop // Ett litet program som vi testar DrJava's debugger på public class Summera_12345 { public static void main (String[] arg) { int sum=0; for (int i=1; i<=5; i++) { sum = sum + i; l1 } // Slut loopen l2 System.out.println("Summan = " + sum); l4 l5 } // Slut mainmetoden l6 r r r								
Interactio	ns Cons	ole Compile	r Output Br	eakpoints				
Z:\pro 8: int s	ig1\public_h sum=0;	tml\ht14\miniL	essons\Autotr	ace_Debugg\	Summera_12	345.ja Pr	Next Next Go to emove Move All	×
Editing Z:\pr	rog1\public_	html\ht14\minil	_essons\Auto	trace_Debug	g\Summer			8:5

Vi har därmed bestämt en "stoppsats" i programmet, vilket innebär att om vi kör programmet, kommer programmet att stanna vid denna sats. Man kan köra programmet vidare från denna sats. Nu skall vi köra programmet, genom att välja Run. Vad som kommer att hända är att programmet kommer att starta, men stanna vid den sats vi markerat med rött



Den röda satsen blir blå, vilket indikerar att programmet har tagit en paus och nästa sats som skall utföras är den

blåa



I debuggerfönstret, skriv in variabelnamnet sum följt av ENTER, så bör det se ut så här.



Vi har därmed begärt att få bevakning på variabeln sum, dvs få dess värden utskrivna under Value. Men där står <not found> vilket beror på att satsen int sum=0; inte är utförd än.



Så här ser det ut i programkoden, nästa sats att utföra är blå.



För varje Step Over som görs kommer programmet att utföra nästa sats. Programmet innehåller ju en loop, så vi kommer att se att programmet snurrar i loopen.



DrJava's debugger - bevakning av variabelvärden

När programmet utför det andra varvet i loopen ser det ut så här.



Vi ser i debuggerfönstret (under Watches) att variabeln sum har värdet 1

🛓 Debugge	r - Z:\prog1	public_html	h	t14\miniLesso	ns\Aut	x
Watches					Resume	×
Name	Value	Туре	•		Automatic Trace	
sum	1	int or Int			Step Into	
				M	Step Over	
				s	Step Out	
				S		

För att köra färdigt programmet, välj Resume.

Alternativ körning – debugga med Automatic Trace

Låt nu programmet vara kvar i Debug Mode och ha kvar stoppsatsen enligt föregående exempel. Starta programmet med Run, ha kvar bevakning på sum, men välj nu Automatic Trace (istf tidigare Step Into). Programmet kommer nu köra hela programmet sats för sats i slowmotion och du kommer löpande få se värdet av sum.



För att ändra hastigheten på slowmotion, gå in under Edit>Preferences>Debugger Och ändra värdet på Auto-Step Rate