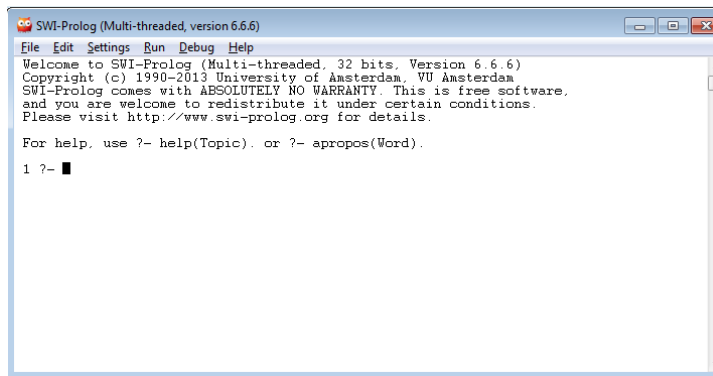
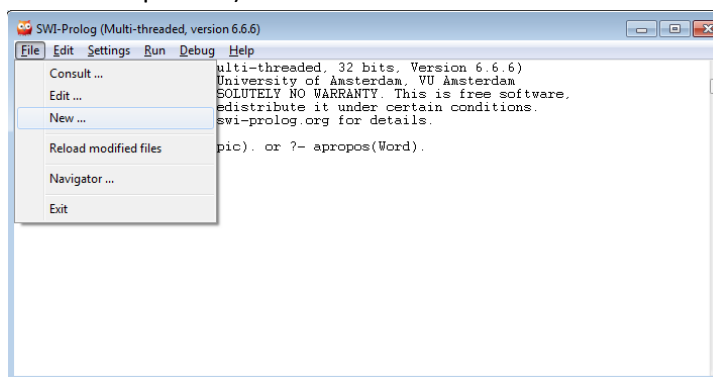


Using SWI-Prolog

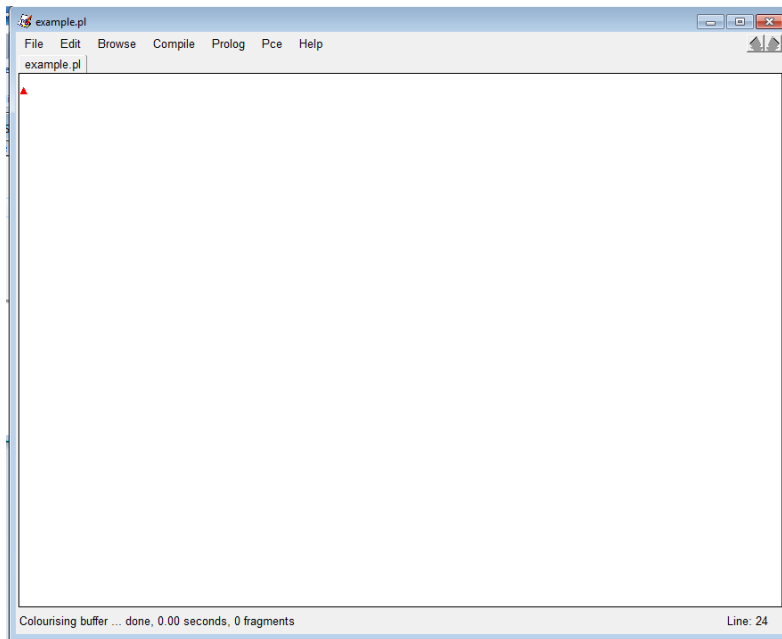
1. Install SWI-Prolog (<http://www.swi-prolog.org/Download.html>)
2. Start the SWI-Prolog Console. It should look like this:



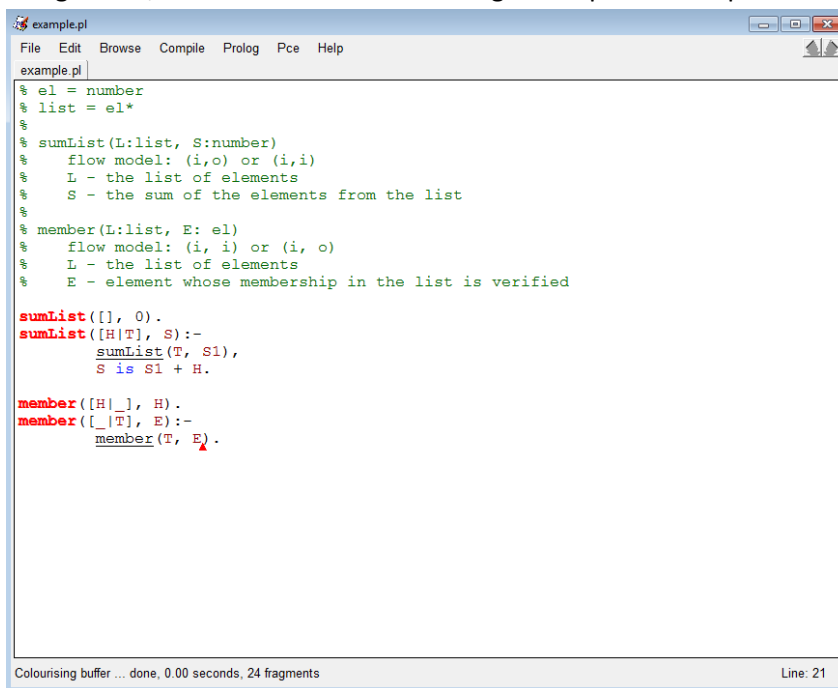
3. For creating a new file, choose File-> New. Give a name to the new file and add the .pl extension (SWI-Prolog files have .pl as the default extension, unless you selected .pro during the installation process).



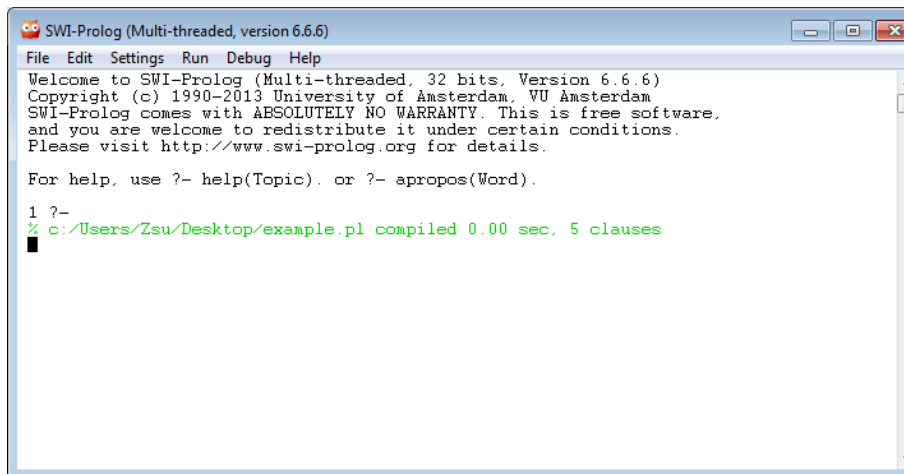
4. The editor window will open.



5. In the editor you can write the clauses for the needed predicates. Don't forget to add the necessary comments at the beginning of the file! (the domain definitions, and for each predicate its signature, flow model and the meaning of the predicates' parameters).



6. When finished with the predicates, save the file and return to the SWI-Prolog console. Choose File->Consult and select the file where your code is. You should see a message that the file was loaded successfully.

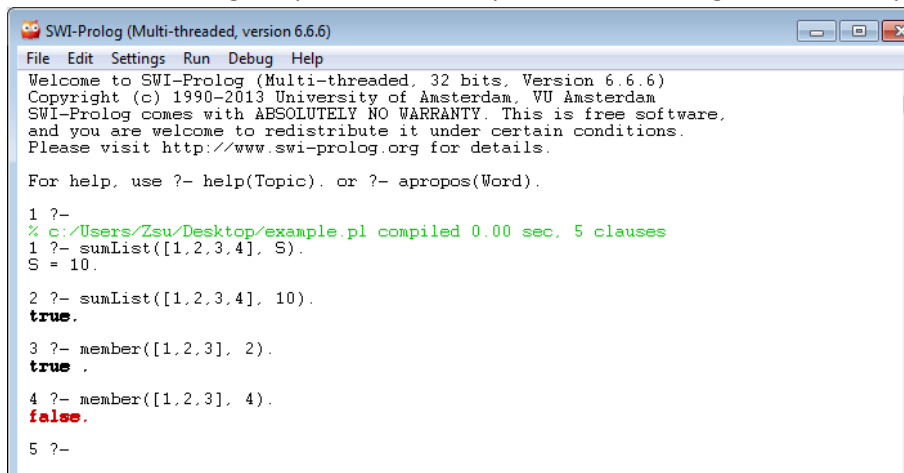


```
SWI-Prolog (Multi-threaded, version 6.6.6)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (Multi-threaded, 32 bits, Version 6.6.6)
Copyright (c) 1990-2013 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?-
% c:/Users/Zsu/Desktop/example.pl compiled 0.00 sec, 5 clauses
█
```

7. You can start calling the predicates from your file. Don't forget to end every call with a period.



```
SWI-Prolog (Multi-threaded, version 6.6.6)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (Multi-threaded, 32 bits, Version 6.6.6)
Copyright (c) 1990-2013 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?-
% c:/Users/Zsu/Desktop/example.pl compiled 0.00 sec, 5 clauses
1 ?- sumList([1,2,3,4], S).
S = 10.

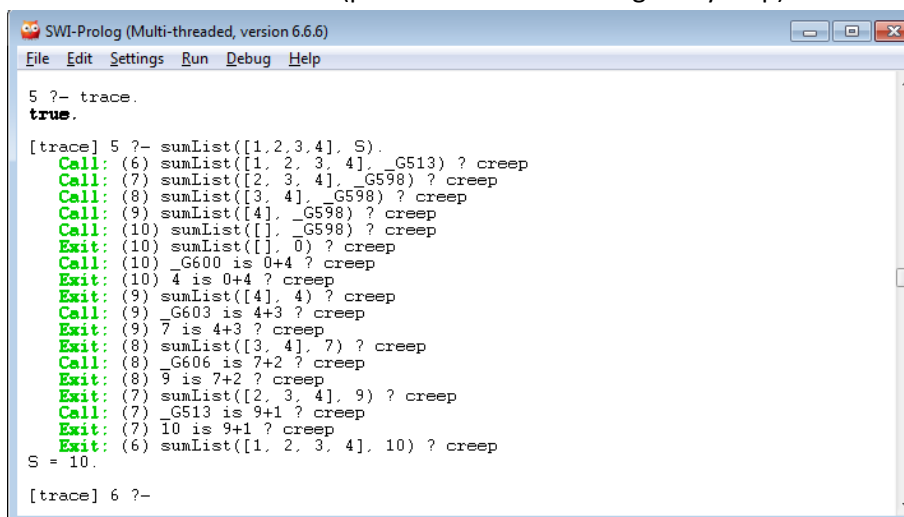
2 ?- sumList([1,2,3,4], 10).
true.

3 ?- member([1,2,3], 2).
true.

4 ?- member([1,2,3], 4).
false.

5 ?-
```

8. If you want to see how the predicates are called or to debug your program, you can use the *trace*. command. Once entering trace mode, for every predicate call you will see the recursive calls and returns for the call (press *Enter* for executing every step).



```
SWI-Prolog (Multi-threaded, version 6.6.6)
File Edit Settings Run Debug Help

5 ?- trace.
true.

[trace] 5 ?- sumList([1,2,3,4], S).
Call: (6) sumList([1, 2, 3, 4], _G513) ? creep
Call: (7) sumList([2, 3, 4], _G598) ? creep
Call: (8) sumList([3, 4], _G598) ? creep
Call: (9) sumList([4], _G598) ? creep
Call: (10) sumList([], _G598) ? creep
Exit: (10) sumList([], 0) ? creep
Call: (10) _G600 is 0+4 ? creep
Exit: (10) 4 is 0+4 ? creep
Exit: (9) sumList([4], 4) ? creep
Call: (9) _G603 is 4+3 ? creep
Exit: (9) 7 is 4+3 ? creep
Exit: (8) sumList([3, 4], 7) ? creep
Call: (8) _G606 is 7+2 ? creep
Exit: (8) 9 is 7+2 ? creep
Exit: (7) sumList([2, 3, 4], 9) ? creep
Call: (7) _G513 is 9+1 ? creep
Exit: (7) 10 is 9+1 ? creep
Exit: (6) sumList([1, 2, 3, 4], 10) ? creep
S = 10.

[trace] 6 ?-
```

9. To exit trace mode, enter the *notrace*. command.