Programming Paradigms

Literate Programming

Introduction

- What is Literate Programming?
- WEB, Weave and Tangle
- Coding Example
- Evolution of WEB
- Modern Uses
- Influences
- Further Reading
- Questions

What is literate programming?

- Conceived in 1980 by Donald Knuth.
- Knuth developed TeX .
- Developed as a result of:
  - Structured programming movement.
  - Programs as literature.
  - Proper documentation problems.
  - No language existed that is capable of both proper documentation and fast execution times.

What is literate programming?

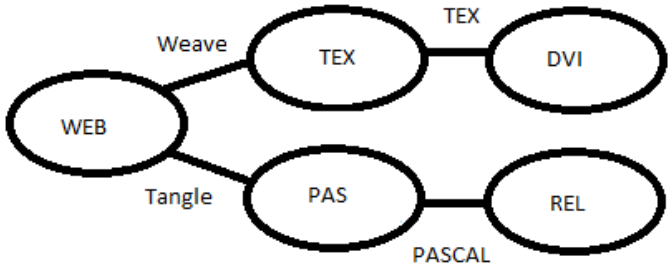
- Literate programming is the concept of writing programs as prose.
- A single document can generate both Proper and meaningful documentation and a Quickly executable code.
- Code is written to be logical to reader not computer.
- Knuth developed WEB to be able to do this.

WEB, Weave and Tangle

- Web was developed in 1981.
- Uses TEX as a documentation language and Pascal as a Programming language.
- Uses macros to link the two languages.
- Once a .web program is written two commands are used on it:
  - Weave – To generate Documentation.
  - Tangle – To generate Compiler Code.

WEB, Weave and Tangle

- The weave command interprets the web code in such a way to remove most code and produce formatted documentation.
- The Tangle command removes all comments from the program and reduces it to only compiler code with section numbers for comments.



Coding Example

```
@ To run the program with, say, a {\mc UNIX} shell, just type `\.{advent}'
and follow instructions. (Many {\mc UNIX} systems come with an
almost identical program called `\.{adventure}' already built in;
you might want to try it too, for comparison.)

@p
#include <stdio.h> /* basic input/output routines: |fgets|, |printf| */
#include <ctype.h> /* |isspace| and |tolower| routines */
#include <string.h> /* |strncmp| and |strcpy| to compare and copy strings */
#include <time.h> /* current |time|, used as random number seed */
#include <stdlib.h> /* |exit| */
@<Macros for subroutine prototypes@>@;
@#
typedef enum{@!false,@!true}boolean;
@<Type definitions@>@;
@<Global variables@>@;
@<Subroutines@>@;
@#
main()
{
  register int j,k;
  register char *p;
  @<Additional local registers@>;
  @<Initialize all tables@>;
  @<Simulate an adventure, going to |quit| when finished@>;
  @<Deal with death and resurrection@>;
quit: @<Print the score and say adieu@>;
  exit(0);
}
```

- @ used as a escape character for special commands for the weave commands.
- @ used in conjunction with <,>,!,^,P,I, etc. format the TEX document with appropriate style, font and size when weave is used on the document
- @ also used by Tangle to know what text to exclude and which sections to compile. Also allows the proper section comments to be added to the Pascal code.

Coding Example

```
2. To run the program with, say, a UNIX shell, just type 'advent' and follow instructions.
        (Many UNIX systems come with an almost identical program called 'adventure'
        already built in; you might want to try
it too, for comparison.)
#include <stdio.h> =¤ basic input/output routines: fgets , printf ¤=
#include <ctype.h> =¤ isspace and tolower routines ¤=
#include <string.h> =¤ strncmp and strcpy to compare and copy strings ¤=
#include <time.h> =¤ current time , used as random number seed ¤=
#include <stdlib.h> =¤ exit ¤=
< Macros for subroutine prototypes 3 >
typedef enum {
   False ; true
} boolean;
<Type definitions 5 >
< Global variables 7 >
< Subroutines 6 >
main( )
{
        register int j; k;
        register char ¤p;
       < Additional local registers 22 >;
       < Initialize all tables 200 >;
       < Simulate an adventure, going to quit when finished 75 >;
       < Deal with death and resurrection 188 >;
       quit : > Print the score and say adieu 198 >;
       exit (0);
}
```

- Sample of woven .web file.
- All text is now formatted properly with appropriate font and style.
- Macros now have references to other relevant sections of code.
- At the end of the woven file is an index that keeps of the references between macros and sections of code.
- This style allow program flow to be easily followed and well documented at the same time.

Coding Example

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <time.h>
#include <stdlib.h>
typedef enum{@!false,@!true}boolean;

main()
{
    int j,k;
    char *p;
    InitializeTable();
    Adventure();
    PrintResult();
     Return 0;
}
```

- Sample of tangled .web code.
- All comments are remove from the document.
- Macros are now replaced with function calls.
- Many variables and function have been move to be logical to the compiler.

Evolution of WEB

- Web83 was the first expansion on the web language, which increased scalability
- Pascal is a poor compiler language.
- Pascal was replaced by c as well as integrated into other Programs such as Matlab and Maple.
- Produced cWeb, MatWeb, MapleWeb, etc. in the late 80's and early 90's.

Modern Uses

- None
- Use of true literate programming has gone extinct.
- The most recent articles on the subject where in late 90's and early 2000.
- Knuth has continued some work in the subject since however it was only for educational purposes.

Influences

- While literate programming went extinct certain concepts remain.
- Literate programming solution to program flow influenced smart IDEs.
- As well systems for in program documentation such as Java doc where inspired by literate programming.
- Eclipse with java Doc is the modern successor of WEB.

Further Reading

- Knuth, Donald E. (1992). *Literate Programming.*, California: Stanford University Center for the Study of Language and Information
- http://www.literateprogramming.com/
- http://www.faqs.org/faqs/literate-programming-faq/index.html

Questions

- Questions?