20CYS312 - Principles of Programming Languages
Exploring Programming Paradigms

Assignment-01

Presented by Ashwin Anand
CB.EN.U4CYS21012
TIFAC-CORE in Cyber Security
Amrita Vishwa Vidyapeetham, Coimbatore Campus

Feb 2024

Outline

1. Paradigm - Declarative
2. Language - R
3. Paradigm - Scripting
4. Language - Perl
5. Comparison and Conclusion
6. Bibliography

Paradigm - Declarative

Declarative Programming focuses only on the goal to be achieved, rather than the path to be taken to achieve said goal. It depends on its components to carry out the steps.

A simple example of a declarative language is SQL. A simple select statement says nothing more than "SELECT something FROM something WHERE some condition". Here, we do not mention how to select, or how the check happens, but rather just depend on the statement to do its work.

Due to its heavy dependence on the preconfigured capabilities of the language, there are no 'purely' declarative programming languages. Most languages have declarative features, along with other paradigms like functional or procedural.

Some of the key concepts and features associated with declarative programming are:

- Abstraction of functions
- Declaration of immutable objects
- Correspondence and Dependence on mathematical logic
- Domain specific languages

Language - R

Figure: Logo of the language R.

What is R?

R is a language developed to facilitate statistical computing and graphics. It was heavily inspired by the language S, which was developed for a similar purpose years earlier.

The language has a wide range of predefined mathematical and statistical functions, which when paired with basic arithmetic are sufficient for most calculations. It also offers the *plot* function, which can be used to obtain a graphical representation if necessary.

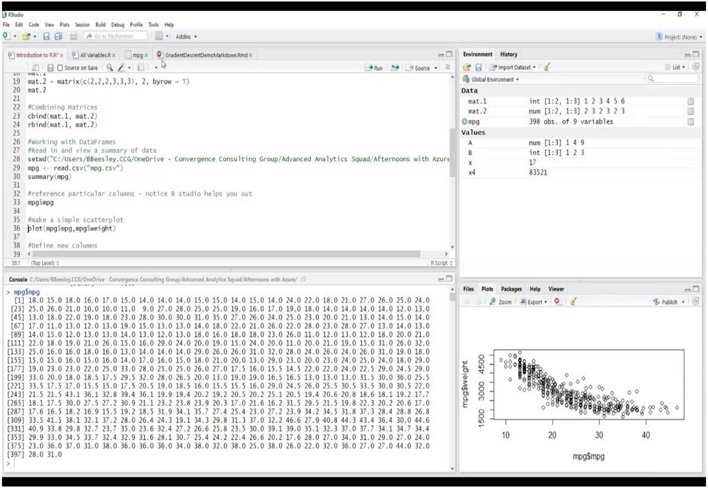
Some key features of R include:

- Open source, with an active community
- Several packages available to perform variety of functions
- Can be integrated with languages like Python and C++

-

Paradigms present in R include Functional Programming, Object-Oriented Programming (OOP), Vectorized Programming, Declarative Programming, Procedural Programming and Scripting.

R interface



```r
mat.1
mat.2 = matrix(c(2,2,2,3,3,3), 2, byrow = T)
mat.2

#Combining Matrices
cbind(mat.1, mat.2)
rbind(mat.1, mat.2)

#Working with DataFrames
#Read in and view a summary of data
setwd("C:/Users/BBeesley.CCG/OneDrive - Convergence Consulting Group/Advanced Analytics Squad/Afternoons with Azure
mpg <- read.csv("mpg.csv")
summary(mpg)

#reference particular columns - notice R studio helps you out
mpg$mpg

#make a simple scatterplot
plot(mpg$mpg,mpg$weight)

#Define new columns
```

Console C:/Users/BBeesley.CCG/OneDrive - Convergence Consulting Group/Advanced Analytics Squad/Afternoons with Azure/

```
> mpg$mpg
  [1] 18.0 15.0 18.0 16.0 17.0 15.0 14.0 14.0 14.0 15.0 15.0 14.0 15.0 14.0 24.0 22.0 18.0 21.0 27.0 26.0 25.0 24.0
 [23] 25.0 26.0 21.0 10.0 10.0 11.0  9.0 27.0 28.0 25.0 25.0 19.0 16.0 17.0 19.0 18.0 14.0 14.0 14.0 14.0 12.0 13.0
 [45] 13.0 18.0 22.0 19.0 18.0 23.0 28.0 30.0 30.0 31.0 35.0 27.0 26.0 24.0 25.0 23.0 20.0 21.0 13.0 14.0 15.0 14.0
 [67] 17.0 11.0 13.0 12.0 13.0 19.0 15.0 13.0 13.0 14.0 18.0 22.0 21.0 26.0 22.0 28.0 23.0 28.0 27.0 13.0 14.0 13.0
 [89] 14.0 15.0 12.0 13.0 13.0 14.0 13.0 12.0 13.0 18.0 16.0 18.0 18.0 23.0 26.0 11.0 12.0 13.0 12.0 18.0 20.0 21.0
[111] 22.0 18.0 19.0 21.0 26.0 15.0 16.0 29.0 24.0 20.0 19.0 15.0 24.0 20.0 11.0 20.0 21.0 19.0 15.0 31.0 26.0 32.0
[133] 25.0 16.0 16.0 18.0 16.0 13.0 14.0 14.0 14.0 29.0 26.0 26.0 31.0 32.0 28.0 24.0 26.0 24.0 26.0 31.0 19.0 18.0
[155] 15.0 15.0 16.0 15.0 16.0 14.0 17.0 16.0 15.0 18.0 21.0 20.0 13.0 29.0 23.0 20.0 23.0 24.0 25.0 24.0 18.0 29.0
[177] 19.0 23.0 23.0 22.0 25.0 33.0 28.0 25.0 25.0 26.0 27.0 17.5 16.0 15.5 14.5 22.0 22.0 24.0 22.5 29.0 24.5 29.0
[199] 33.0 20.0 18.0 18.5 17.5 29.5 32.0 28.0 26.5 20.0 13.0 19.0 19.0 16.5 16.5 13.0 13.0 13.0 31.5 30.0 36.0 25.5
[221] 33.5 17.5 17.0 15.5 15.0 17.5 20.5 19.0 18.5 16.0 15.5 15.5 16.0 29.0 24.5 26.0 25.5 30.5 33.5 30.0 30.5 22.0
[243] 21.5 21.5 43.1 36.1 32.8 39.4 36.1 19.9 19.4 20.2 19.2 20.5 20.2 25.1 20.5 19.4 20.6 20.8 18.6 18.1 19.2 17.7
[265] 18.1 17.5 30.0 27.5 27.2 30.9 21.1 23.2 23.8 23.9 20.3 17.0 21.6 16.2 31.5 29.5 21.5 19.8 22.3 20.2 20.6 17.0
[287] 17.6 16.5 18.2 16.9 15.5 19.2 18.5 31.9 34.1 35.7 27.4 25.4 23.0 27.2 23.9 34.2 34.5 31.8 37.3 28.4 28.8 26.8
[309] 33.5 41.5 38.1 32.1 37.2 28.0 26.4 24.3 19.1 34.3 29.8 31.3 37.0 32.2 46.6 27.9 40.8 44.3 43.4 36.4 30.0 44.6
[331] 40.9 33.8 29.8 32.7 23.7 35.0 23.6 32.4 27.2 26.6 25.8 23.5 30.0 39.1 39.0 35.1 32.3 37.0 37.7 34.1 34.7 34.4
[353] 29.9 33.0 34.5 33.7 32.4 32.9 31.6 28.1 30.7 25.4 24.2 22.4 26.6 20.2 17.6 28.0 27.0 34.0 31.0 29.0 27.0 24.0
[375] 23.0 36.0 37.0 31.0 38.0 36.0 36.0 36.0 34.0 38.0 32.0 38.0 25.0 38.0 26.0 22.0 32.0 36.0 27.0 27.0 44.0 32.0
[397] 28.0 31.0
>
```

```
Data
  mat.1     int [1:2, 1:3] 1 2 3 4 5 6
  mat.2     num [1:2, 1:3] 2 3 2 3 2 3
  mpg       398 obs. of 9 variables
Values
  A         num [1:3] 1 4 9
  B         int [1:3] 1 2 3
  x         17
  x4        83521
```

Figure: Example of the R IDE interface

Paradigm - Scripting

Scripting is a type of programming often used by developers for their own personal use. Scripts are simply pieces of code written to perform a certain very particular function as part of a larger code. Scripts are ungeneralized, and work only under very specific circumstances.

The reason for this paradigm existing is because it provides an easier solution than writing reusable functions. It is a tradeoff between generalization and beauty for quick use and quality.

Some features of scripting languages:

- Automation of menial tasks
- Integration with large code
- Ease of use

-

Some common scripting languages include Perl, Bash, and TCL.

Language - Perl

Figure: Logo of the language Perl.

What is Perl?

Perl is a high level general purpose scripting language. Originally released in 1987 as a unix scripting language, it has since gone through many changes, even resulting in a sister language known as Raku.

Perl offers several powerful regex manipulation tools, and text processing facilities. This makes it suitable for a lot of scenarios where task scheduling and automation are necessary.

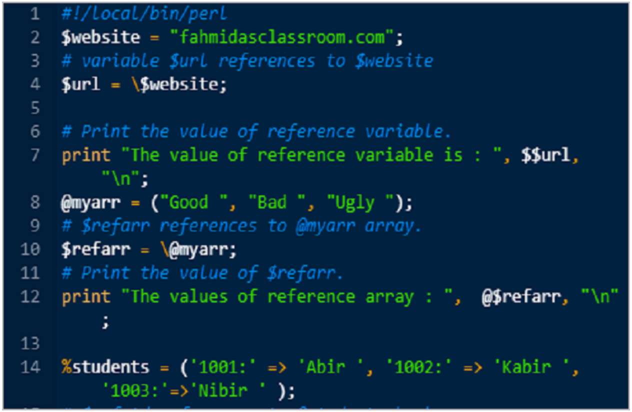
Some key features of Perl include:

- High text processing capabilities
- Very suitable to automate small tasks
- CPAN provides extensive modules and libraries
- Can be used for both hardware and software purposes

-

The paradigms of Perl include Procedural, Object-Oriented, Functional and scripting

Example of Perl



```perl
#!/local/bin/perl
$website = "fahmidasclassroom.com";
# variable $url references to $website
$url = \$website;

# Print the value of reference variable.
print "The value of reference variable is : ", $$url,
    "\n";
@myarr = ("Good ", "Bad ", "Ugly ");
# $refarr references to @myarr array.
$refarr = \@myarr;
# Print the value of $refarr.
print "The values of reference array : ",  @$refarr, "\n"
    ;

%students = ('1001:' => 'Abir ', '1002:' => 'Kabir ',
    '1003:'=>'Nibir ' );
```

Figure: A code snippet in Perl.

Comparison and Conclusion

The declarative paradigm is often used in query languages. It is also used in situations where the system is rule based, and depends on mathematical logic.

In contrary, the scripting paradigm is employed in situations where automation of menial tasks is required. Text processing, or image operations are often written as scripts.

To compare:

- Declarative programming emphasizes on the goal, while scripting provides a form to reach said goal.
- Declarative programming mainly utilizes the approach of abstraction, while scripting depends more on functionality given a scenario.

-

In conclusion, both declarative and scripting paradigms are rather important in the languages today. Most languages that exist right now are a combination of several of these paradigms. For example, Python is a functional programming language (a subset of declarative languages) and also a scripting languages. This makes the language very versatile.

Bibliography

- https://www.partech.nl/nl/publicaties/2021/02/declarative-vs-imperative-programming
- https://stackoverflow.com/questions/129628/what-is-declarative-programming
- https://dev.to/ruizb/declarative-vs-imperative-4a7l
- https://www.r-project.org
- https://en.wikipedia.org/wiki/R_(programming_language)
- https://blog.machinezoo.com/Scripting_as_a_programming_paradigm
- https://en.wikipedia.org/wiki/Scripting_language
- https://www.perl.org
- https://www.geeksforgeeks.org/perl-programming-language/
- https://www.cs.unc.edu/~jbs/resources/perl/perl-basics/