Nature of Programming Languages

Topic 2. Language design criteria

Han, Nguyen Dinh (han.nguyendinh@hust.edu.vn)

Faculty of Mathematics and Informatics
Hanoi University of Science and Technology

Outline

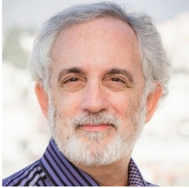
1. Historical overview
2. Efficiency
3. Regularity
4. Security
5. Simplicity and extensibility
6. Case studies: C++ and Python

Language design criteria

1. Historical overview

What is programming language design?

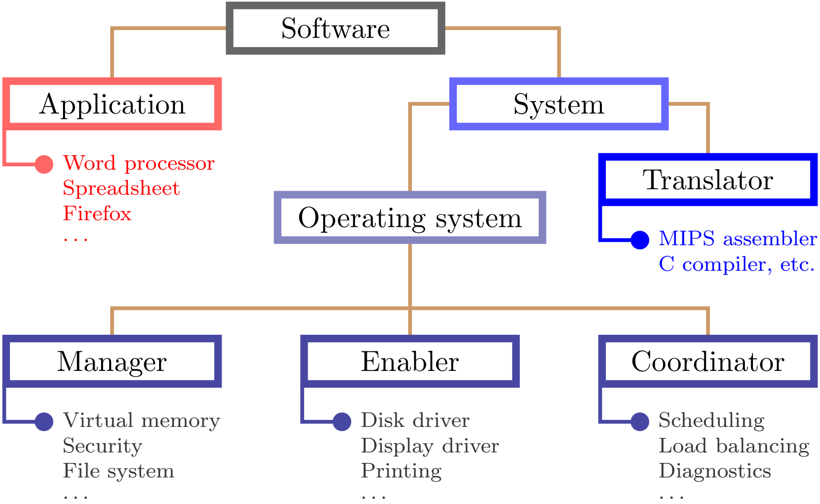
- Bjarne Stroustrup (Creator of C++): There are only two kinds of programming languages: those people always complain about and those nobody uses



- Mitch Kapor (Creator of Lotus, Firefox): What is design? It's where you stand with a foot in two worlds - the world of technology and the world of people and human purposes - and you try to bring the two together

- Larry Wall (Creator of Perl): Computer language design is just like a stroll in the park. Jurassic Park, that is

What is the primary purpose of a programming language?



Software

Application

- Word processor
- Spreadsheet
- Firefox
- . . .

System

Translator

- MIPS assembler
- C compiler, etc.

Operating system

Manager

- Virtual memory
- Security
- File system
- . . .

Enabler

- Disk driver
- Display driver
- Printing
- . . .

Coordinator

- Scheduling
- Load balancing
- Diagnostics
- . . .

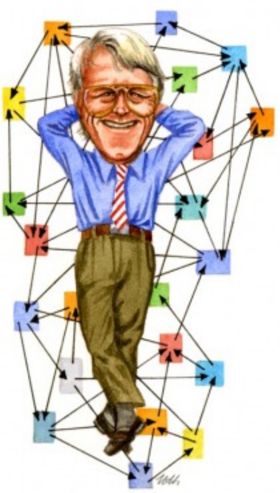
What is good programming language design?

A language is successful if it satisfies any or all of the following criteria:

1. Achieves the goals of its designers
2. Attains widespread use in an application area
3. Serves as a model for other languages that are themselves successful

NOTE ⇝ It is still extremely difficult to describe good programming language design. Fred Brooks, a computer science pioneer, maintains that language design is similar to any other design problem, such as designing a building. Niklaus Wirth, the designer of Pascal, advises that simplicity is paramount!

What is good programming language design?



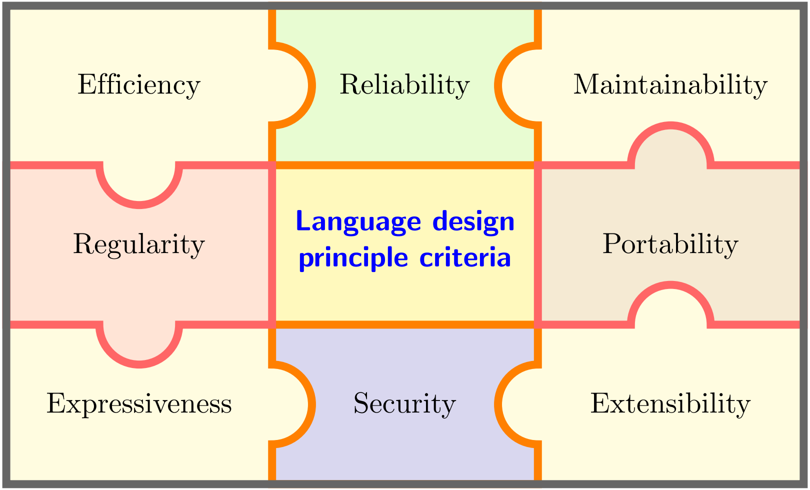
shfpucsp.wordpress.com

Designing an object to be simple and clear takes at least twice as long as the usual way. It requires concentration at the outset on how a clear and simple system would work, followed by the steps required to make it come out that way - steps which are often much harder and more complex than the ordinary ones. It also requires relentless pursuit of that simplicity even when obstacles appear which would seem to stand in the way of that simplicity

Theodor H. Nelson

Creator of Hypertext and Hypermedia

What are language design principle criteria?



Efficiency

Reliability

Maintainability

Regularity

Language design principle criteria

Portability

Expressiveness

Security

Extensibility

Historical overview

Can anyone share with us
your opinion about
programming language design criteria?

Language design criteria

2. Efficiency

Efficiency

- Executable code

Language	Speed	Space
C (gcc)	1.0	1.1
C++(g++)	1.0	1.6
Java (JDK)	1.7	9.1
Lisp	1.7	11
C# (mono)	2.4	5.6
Python	6.5	3.9
Ruby	16	5.0

- Translation

Java

```
int i = 10;
String s = "My information";
// Do something with i and s
```

Python

```
i = 10
s = "My information";
# Do something with i and s
```

- Programming: re-usability, security, maintainability, etc.

Language design criteria

3. Regularity

Regularity

- Generality: operations/constructs available for all closely related cases
  - In C: compare two integers with == but not two structures/arrays
  - In Java: can make collections of objects (e.g. Integer) but not primitive values (e.g. int)
- Orthogonality: constructs can be combined in a meaningful way
  - In C: all parameters are passed by value except arrays
  - In Java: a class can have static members but an abstract class cannot
- Uniformity: constructs appear and behave consistently (e.g. in C, Java: = means "assignment" while == is a comparison)

Language design criteria

4. Security

Security

- Constructs to build programs that cannot be subverted
  - Array bounds checks
  - Safety in types
- Catch programming errors at compile time
  - Strong type system
  - Memory safety
- Constructs to handle usage errors
  - Exception handling mechanism
- Mechanisms to test and uncover errors
  - Recycle dynamic storage prevention

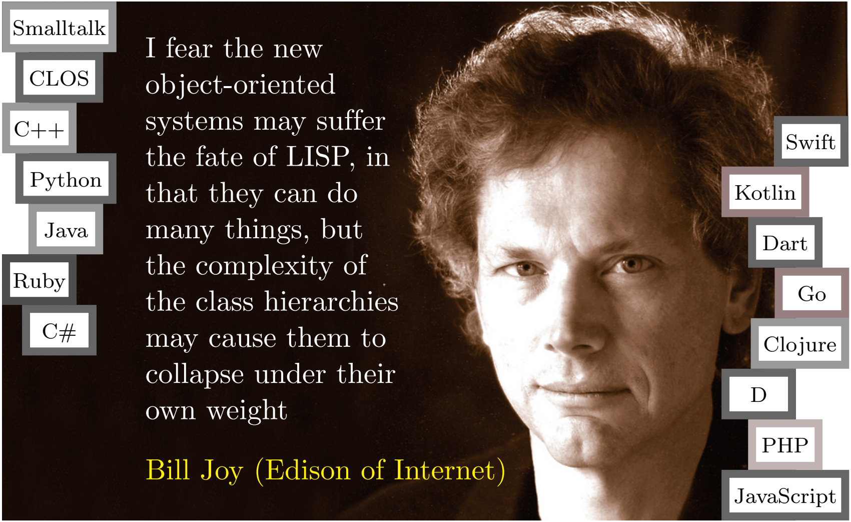
Language design criteria

5. Simplicity and extensibility

Simplicity and extensibility

- Simplicity: ability to express programs concisely, in a manner that is easy to write, read, and understand
  - Simplicity of learning vs. simplicity of programming vs. simplicity of understanding
  - Small set of basic concepts
  - Constructs can be expressed and used only in one way
- Extensibility: ability to add new features to the language
  - Data type definition in Pascal, Modula, Ada, etc.
  - Definition of new operators (or reuse existing operators such as '+') in SML, Prolog, Haskell, etc.

Simplicity and extensibility



Smalltalk
CLOS
C++
Python
Java
Ruby
C#

> I fear the new object-oriented systems may suffer the fate of LISP, in that they can do many things, but the complexity of the class hierarchies may cause them to collapse under their own weight

Bill Joy (Edison of Internet)

Swift
Kotlin
Dart
Go
Clojure
D
PHP
JavaScript

Language design criteria

6. Case studies: C++ and Python

Your second assignment

Students work individually to carry out the following tasks:

1. Investigate insightfully the design of C++ and Python
2. Find and summarize all advanced design principles and features that make the success of these two languages
3. Point to any "mistakes" in the design of C++ and Python
4. Provide evidence (i.e. code snippets, literature facts and examples, etc.) to support your findings
5. Report (in Word or Latex) and share your methods and results

NOTE ⇝ The report should have two separate parts (i.e. for C++ and Python, respectively). Please convert your report to PDF when you submit it

Language design criteria

THANK YOU VERY MUCH FOR YOUR ATTENTION!