PROGRAMLAMA DİLLERİ

Hafta 3

Dil Değerlendirme Kriterleri

- **Okunabilirlik: programların okunabilme ve anlaşılabilme kolaylığı**

Değerlendirme Kriterleri: Okunabilirlik

- Genel basitlik

Değerlendirme Kriterleri: Okunabilirlik – Genel basitlik

- Large number of basic components - difficult to learn
- User learns only a subset
  - but this subset may differ from one user to another
- feature multiplicity:having more than one way to accomplish an operation
  - e.g. In Java

```
count = count + 1
count += 1
count ++
++count
```

- operator overloading
  - if users are allowed to create their own and not use this sensibly it is a problem, e.g. to use + for integer and floating point addition is acceptable, but to sum up all the elements of two singe dimensional arrays is not – different from vector addition

- On the other hand, the simplest does not mean the best. e.g.Assembly languages

Evaluation Criteria: Readability – Orthogonality

- It means that a relatively small set of primitive constructs can be combined in a relatively small number of ways to build the control and data structures of the language
- Furthermore, every possible combination of primitives is legal and meaningful

Example:

- Four primitive data types : integer, float, double and character
- Two type operators : array and pointer
- If the two type operators can be applied to themselves and the four primitive data types, a large number of data structures can be defined
- However, if pointers were not allowed to point to arrays, many of those possibilities would be eliminated

Evaluation Criteria: Readability – Orthogonality

- Orthogonality is closely related to simplicity
- The more orthogonal the design of a language, the fewer exceptions the language rules require

- **Pascal is not an orthogonal** language, because

  A function cannot return a record (only unstructured types allowed),

  A file must be passed as a **`var`** parameter,

  Formal parameter types must be named (cannot be type descriptions)

  Compound statements are formed by **`begin-end`** pair, except repeat-until

- **C is not an orthogonal language, because**

  -records(structs) can be returned from functions but arrays cannot

  -a member of a structure can be any type but not void or structure of the same type

  -a member of an array can be any type but not void or function

Evaluation Criteria: Readability – Data types and structures

Facilities for defining data types and data structures are helpful for readability

- If there is no boolean type available then a flag may be defined as integer:

  `found = 1` (instead of `found = true`)

  May mean something is found as boolean or what is found is 1

An array of record type is more readable than a set of independent arrays

- In Fortran

```
Character(Len=30) Name(100)
IntegerEmployeeNumber(100)
Real Salary(100)
```

Evaluation Criteria: Readability – Syntax considerations

- **Identifier Forms:** restricting identifier length is bad for readability.
- Example:
  - FORTRAN77 identifiers can have at most **6** characters.
  - The extreme case is the ANSI BASIC, where an identifier is either a single character or a single character followed by a single digit.

- Availability of word concatenating characters (e.g., _) is good for readability.

Evaluation Criteria: Readability – Syntax considerations

Special Words:

Readability is increased by special words (e.g., **`begin`**, **`end`**, **`for`**).

In PASCAL and C, **`end`** or **`}`** is used to end a compound statement. It is difficult tell what an **`end`** or **`}`** terminates.

However, ADA uses **`end if`** and **`end loop`** to terminate a selection and a loop, respectively.

Another issue is the use of special words as names of variables. For example, in FORTRAN77, special words, e.g., **`DO`** and **`END`** can be used as variable names.

Evaluation Criteria: Readability – Syntax considerations

Forms and Meaning:

Forms should relate to their meanings. Semantics should directly follow from syntax.

For example,

sin(x)

should be the sine of **x**,

not the sign of **x** or cosign of **x**.

- Grep is hard to understand for the people not familiar with using regular expressions
- grep : g/regular_expression/p /reg_exp/ : search for that reg_exp g: scope is whole file p:print

Değerlendirme Kriterleri: Yazilebilirlik

- Basitlik ve ortogonalite

```
count++;        count = count + 1;
```

Değerlendirme Kriterleri: Güvenilirlik

- Tür denetimi

Değerlendirme Kriterleri: Güvenilirlik – Tip Denetimi

- Testing for type errors in a given program either by the compiler or during program execution
- The compatibility between two variables or a variable and a constant that are somehow related (e.g., assignment, argument of an operation, formal and actual parameters of a method).
- **Run-time** (Execution-time) checking is expensive.
- **Compile-time** checking is more desirable.
- The earlier errors in programs are detected, the less expensive it is to make the required repairs

Değerlendirme Kriterleri: Güvenilirlik – Tip Denetimi

For example, the following program compiles an runs!

```
foo (float a) {
  printf ("a: %g and square(a): %g\n", a,
  a*a);
}
main () {
  char z = 'b';
  foo(z);
}
```

Output is : `a: 98 and square(a): 9604`

Değerlendirme Kriterleri: Güvenilirlik – Özel Durum İşleme

- The ability of a program
- to intercept run-time errors, as well as other unusual conditions
- to take corrective measures and continue

- Ada, C++, and Java include extensive capabilities for exception handling, but in C and Fortran it is practically non-exsistent

Değerlendirme Kriterleri: Güvenilirlik – Diğer Ad

- Having two distinct referencing methods (or names) for the same memory cell.
- It is a dangerous feature in a programming language.
- E.g., pointers in PASCAL and C
- two different variables can refer to the same memory cell

Değerlendirme için diğer kriterler

- Taşınabilirlik: program bir ortamdan diğerine taşınabilir