L#2

Basics of Programming. Introduction

Course Basics of Programming Semester 1, FIIT

Mayer Svetlana Fyodorovna

Algorithm comparison criteria

- Memory space
- Time of executing

- Memory space

The fewer the number of variables, the less memory is spent

Types & memory space:
Integer => 2 bytes
Real => 6 bytes
Array[2-by-3] of integer => 6 * 2 = 12 bytes

Time of executing the program

- Replacing some parts of expressions with variables can significantly increase the speed of programs
- Example 1:

```
var x := 3;
var y := 4;
var z := 2*(x+1)*(x+1) + 3*(x+1);
```

- Three times of calculation x + 1
- How can we improve our program?
- Let's introduce a temporary variable t to store repeated computations:

```
var x := 3;
var y := 4;
var t := x+1;
var z := 2*t*t + 3*t;
```

expression x + 1 is calculated only once

Time of executing the program

- Replacing some parts of expressions with variables can significantly increase the speed of programs
- Example 2:

```
var x := 1.1;
var y := x*x*x*x*x*x*x*x*x*x*x*x*x*x*x*x; // x^16
```

- 15 * operations
- How can we improve our program?
- Let's introduce a temporary variable z to store some part of calculations:

```
var x := 1.1;
var z := x*x*x*x*x*x*x*x; // x^8
var y := z * z; // x^8 * x^8 = x^16
```

- 8 * operations

```
var x := 1.1;
var y := x * x; // x^2
y := y * y; // x^4
y := y * y; // x^8
y := y * y; // x^16
```

4 * operations (!!!)

4 vs 15 (!!!) – speed is better!

Time of executing the program

- Standard operation **
- Example 3:

```
var x := 1.1;
var y := x ** 16;
```

- Operation ** is very fast!

Standard functions in expressions

```
var x := 1.1;
var y := 2.5;
var m1 := Max(x,y);
var m2 := Min(x,y);
var xy := Abs(x-y); // x – y
var xy := Abs(x-y);
var p := Power(x,y); // x ** y
var len := Sqrt(x*x + y*y); // square root (x ² + y ²)
```

Errors in programs

Syntax error

```
var a := 1
var b := 2;
```

```
Program1.pas(3) : Found 'var' but expected ';'
```

Semantic error

```
var a: integer;
var r: real;
a := r;
```

```
Program1.pas(4) : Incompatible types: 'real' and 'integer'
```

Runtime error

```
var a := ReadInteger; // Enter 'hello'
```

```
Program1.pas(2) : Runtime exception: Input string was not
in a correct format.
```

Integer Division

- Standard operand **div** (integer division)
- Example: *How many Kilobytes are in x bytes?*

```
var x := 3586;
print ('answer = ', x div 1024);
```

- `answer =  3`

Remainder after Division

- **Standard operand mod** (the remainder after dividing)
- Example 1:

```
print (10 mod  2); // output 0
print (22 mod  7); // output 1
print (65 mod 10 ); // output 5
```

- Example 2:

```
print (10 mod  2); // output 0 ⟹ 10 is an even number
print (21 mod  2); // output 1 ⟹ 21 is an odd number
```

- Example 3:

```
var x:=352;
print ($'left digit of a number = {x div 100}'); // output 3
print ($'right digit of a number = {x mod 10}'); // output 2
print ($'middle digit of a number = {x div 10 mod 10}'); // output 5
```

Standard Form Of A Number

- Any number can be disassembled into digits using powers of **10** and specified operations.
- This rule is called the standard form of a number:

$$123 = 1*100 + 2*10 + 3$$

Q & A