Използване на променливи, данни, изрази и константи

Има данни, които се инициализират още при тяхното деклариране, в compile time(напр. string- ове, int- ове и др.) и такива, които се инициализират по- късно, в runtime(напр. обекти)

Повечето конвенции, които са споменати надолу се отнасят за езици като C# и Java

Хубаво е когато декларираме дадена променлива да й задаваме някаква начална стойност дори тази стойност да е default-натата

Wrong:

int sum;

….

Console.WriteLine(sum);

Correct:

int sum = 0;

….

Console.WriteLine(sum);

Вместо да си създаваме собствена логика за променливите е по-добре да използваме енумарации, т.е. :

Грешно:

int mode = 1;

…

if (mode == 1) {…} // Read

if (mode == 2) {…} // Write

if (mode == 3) {…} // Read and Write

Правилно:

enum ResourceAccessMode { Read, Write, ReadAndWrite }

!ако даден език няма енумерации в синтаксиса си, то може да използваме константи вместо енумерации

Когато връщаме някаква стойност в някакъв метод, трябва да присвояваме тази стойност на дадена променлива и чак след това да връщаме резултата(така може да дебъгваме кода по-лесно)

Scope – къде е видима дадена променлива

**Always try to reduce maximally the scope of a variable**

This way the coupling is reduced

Avoid public fields (exception: readonly, const in C#)

Span – времето, между което се използва дадена променлива. Той се дефинира като брой редове код. Колкото по-малко е span- а на една променлива, т.е. колкото по-малко редове код има между използването на една променлива, толкова по-добре

Трябва да дефинираме променливите точно преди тяхното използване

Live time – цялото време, през което използваме дадена променлива – от първоначалната й декларация до последното използване на тази променлива

Една променлива трябва да прави едно единствено нещо!

Never use complex expressions in the code!

Example: matrix[i][j] = matrix[xCoords[findMax(i)+1]][ycoords[findMind(j)-1]]\*matrix[yCoords[findMax(j)+1]][xCoords[findMind(i)-1]];

Magic numbers (exceptions: 0, -1, 1, null, “”) – their meaning is not obvious. All magic numbers should be presented as constants with describable names.

JS does not support constants so we have to simulate constants (written in ALL\_CAPS):

var PI = 3.14;

var CONFIG =

{

COLOR: “#000000”,

DEFAULT\_WIDTH: 200,

DEFAULT-HEIGHT: 300

};

When to use constants?

File names: public static readonly string SettingsFileName =

“ApplicationSettings.xml”;

Mathematical constants: public const double E = 2.7182;

Bounds and ranges: public const int ReadBufferSize = 5 \* 1024 \* 1024;