Използване на конструкции за управление – условни изрази и цикли

The most important to consider when organizing code is ordering dependencies

Wrong:

GetData();

GroupData();

Print();

Correct:

data = GetData();

groupedData = GroupData(data);

PrintGroupedData(groupedData);

Always put the normal (expected) condition first after the if clause

Wrong:

**var response = GetHttpWebResponse();**

**if (response.Code == Code.NotFound)**

**{**

**// ...**

**}**

**else**

**{**

**if (response.Code == Code.OK)**

**{**

**// ...**

**}**

**}**

Correct:

**var response = GetHttpWebResponse();**

**if (response.Code == Code.OK)**

**{**

**// ...**

**}**

**else**

**{**

**if (response.Code == Code.NotFound)**

**{**

**// ...**

**}**

**}**

Avoid comparing to true or false

Wrong:

**if (HasErrors == true)**

**{**

**...**

**}**

Correct:

**if (HasErrors)**

**{**

**...**

**}**

Start from the most common cases and then go to the more uncommon ones

Avoid double negation:

Wrong:

**if (!HasNoError)**

**{**

**DoSomething();**

**}**

Correct:

**if (HasErrors)**

**{**

**DoSometing();**

**}**

Do not use complex if conditions

* + You can always simplify them by introducing boolean variables or boolean methods

Wrong:

**if (x > 0 && y > 0 && x < Width-1 && y < Height-1 &&**

**matrix[x, y] == 0 && matrix[x-1, y] == 0 &&**

**matrix[x+1, y] == 0 && matrix[x, y-1] == 0 &&**

**matrix[x, y+1] == 0 && !visited[x, y])**

the code above can be refactored in this way:

**bool inRange = x > 0 && y > 0 && x < Width-1 && y < Height-1;**

**if (inRange)**

**{**

**bool emptyCellAndNeighbours =**

**matrix[x, y] == 0 && matrix[x-1, y] == 0 &&**

**matrix[x+1, y] == 0 && matrix[x, y-1] == 0 &&**

**matrix[x, y+1] == 0;**

**if (emptyCellAndNeighbours && !visited[x, y]) …**

**}**

Write numeric boolean expressions as they are presented on a number line:

-------|---------|------------|---------

a x b

correct:

if(a < x && x < b) { … }

for loops

* Use meaningful variable names

Correct:

for (int year =2000; year < 2015; year++) { … }

Avoid code that depends on the loop’s last index

Добре е циклите да са кратки. За да бъдат кратки е добре да отделяме по-големите парчета код в отделни методи