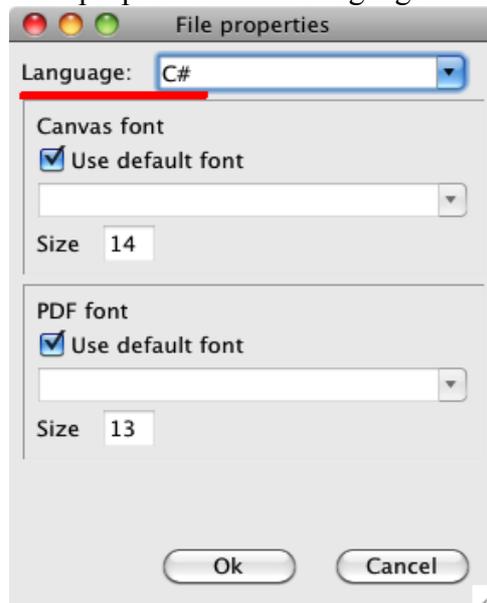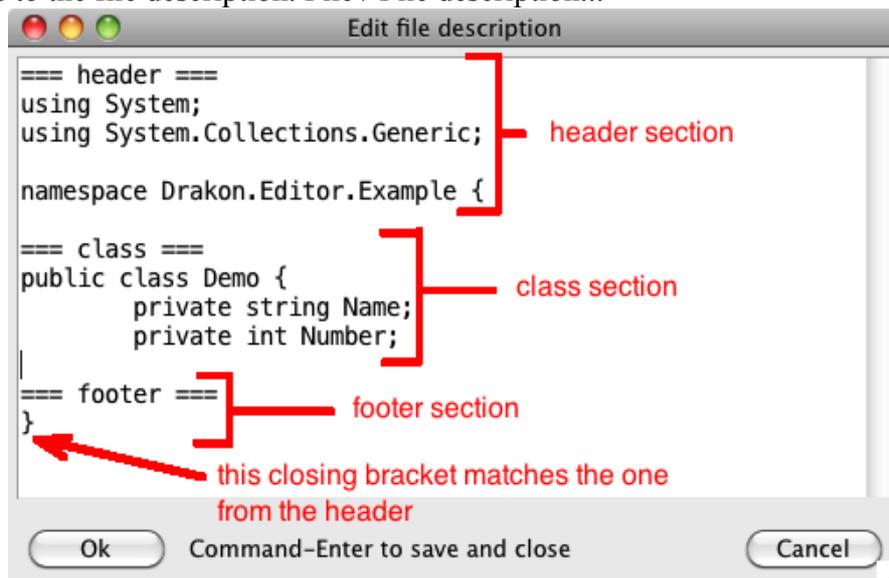# C# programming with DRAKON Editor

1. Set the language to C#. File / File properties... → Language.

2. Add sections to the file description. File / File description...

The **header** goes to the top of the output file.
The **footer** goes to the bottom of the output file.
The **header** and **footer** sections are optional.

The **class** section must contain the start of the class declaration.
- It must contain the class (or struct) name.
- The class section may or may not contain fields and methods.
- There can be only one class per file.

Please do not put the closing bracket to the **class** section }. DRAKON Editor will do it for you.

3. Add keywords, arguments and the return type to the *parameters* item of the diagram.
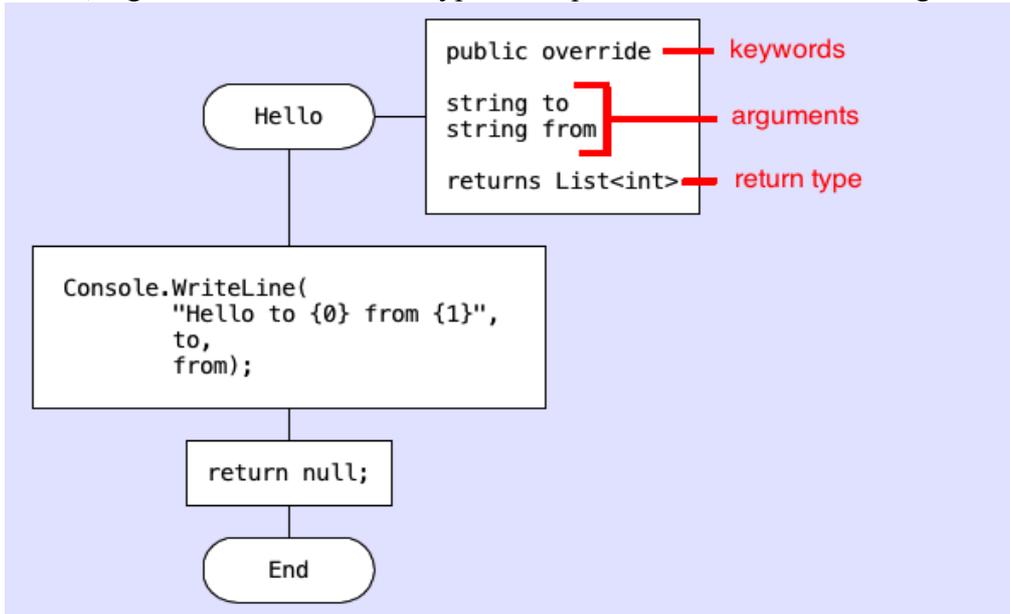
Keywords are optional.

Keywords

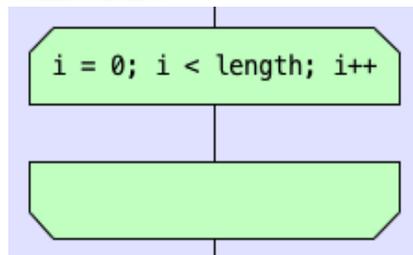Access: **public, protected, private, internal.**
Dispatch: **virtual, abstract, static, override**
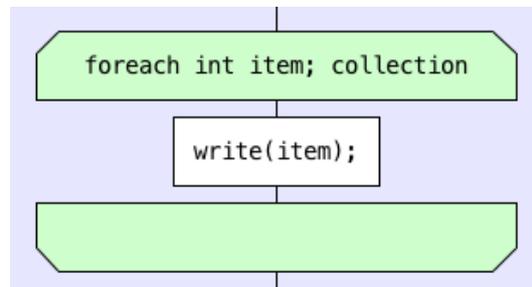**ctr** turns the method into a constructor.

Loop syntax

The *Loop start* icon is similar to the standard C# *for* construct. There are two ways to use *Loop start:*
1. Three expressions separated by semicolons:

The loop variable should be declared at the start of the diagram.

2. "for each"-style loop:

The *Loop start* item should have: **foreach Type loopVar; collection**
There is a limitation: the type of the loop variable cannot be **var**.
The collection must implement the **IEnumerable<T>** interface.
The loop variable name must be unique within the diagram.

Examples

Parameters item	Generated method header
public	public void MethodName() {
protected ctr	protected ClassName() {

public static
int left
int right
returns String
	public static String MethodName(
	    int left,
	    int right) {

protected override String foo	protected override void MethodName(String foo) {
protected abstract returns int	protected abstract int MethodName();

Problems

Problem: the C# compiler may complain that a variable is not initialized before use.
How to solve: declare and initialize the variable at the beginning of the diagram.