Cheban O.G., Maevskiy M.O.
Students,
Kralina G.S.
Teacher;
Myhol V.M.
Teacher,
College of Information Technology and Land Management
of National Aviation University

COMMON LANGUAGE INFRASTRUCTURE

Common Language Infrastructure (CLI) as solution to development and support program applications, written in various languages of high-level programming.

There are many programming languages of high level in modern informational technology world. It sphere also doesn't stand still, There are persistent creating of new programming language, every year their count is growing, and choose of starting language for young professional, also growing. Except of separating for program language, also has the division of platform which program will develop [2].

Constant and dynamic development of all areas of IT technologies leads to understand all come in information that retrieve at current stage, is impossible. Every professional has choice what learn and what put away. So often come on situation when young developer cannot take part in the project if he doesn't know some program language that take him in less profitable state that his older colleges. The same problem in program support stage. Especial that touch on big company. One more problem is cooperation with various program groups in big projects when develop goes in different program language. Limiting a project to a small group of programming languages goes to problem of searching the right number specialists [3].

Another big problem is developing cross platform programs. In this case important to not only, choose right program language, critical to consider platform flavor that further decrease variant to choose specialists. This problem is well illustrated by the development of the Chess Planet program for Android operational system in 2011. Main problem was in interface. Firstly, he was created by hypertext markup language (HTML), but speed of response was not acceptable. Developer's command was force to throw away the biggest part of their work of interface realization. In addition, there was a problem of dynamic implementation of the interface, with the addition new large size elements, visual artifacts arose.

Microsoft faced these problems and created the technological standard Component Object Model (COM), and although this technology had many flaws – it was very difficult, the necessary cross-platform and a large number of supported programming languages could not be reached, this allowed Microsoft to take into account many errors and create another technology – dotnet (.Net Framework). The goal of creating .Net was to provide a powerful, flexible, and simple programming model compared to COM. The .Net Framework is a software platform

for building programs based on the Windows operating system family, as well as numerous non-Microsoft operating systems such as Mac OS X and various Linux and Unix distributions. The whole .Net stands on three pillars – Common Language Runtime (CLR), Common Type System (CTS), Common Language Specification (CLS). CLR implements automatic management and loading of .Net objects. The CTS describes all possible types and software constructions supported by the CLR. But some languages do not support all CTS features, therefore there is a CLS which lists all the necessary supported features to comply with the .Net technology. There is also a Common Language Infrastructure (CLI), the requirements of which are written in the international standard ECMA-335. CLI, like CLS, defines the requirements for different high-level programming languages for running in different system environments, without the need to rewrite code to match the unique characteristics of system environments.

Thanks to the .Net Framework technology and the standards released on it, it is possible not only to combine various high-level programming languages into one program, but also the performance of this program on different platforms, be it Mac OS X or popular Linux distributions [1]. Also .Net technology is constantly maintained, which means it will not quickly become obsolete. Also, of course, the advantage is that .Net allows you to use the advantages of individual programming languages, although in this case performance is guaranteed on various platforms in all cases.

This article proposes a solution for the development and maintenance of cross-platform programs, as well as programs requiring different and/or specific programming languages using the .Net Framework technology [4]. Also, a young specialist who is well versed in this technology can quickly integrate into a project that goes according to the ECMA-335 standard, where development is being done in a poorly known programming language, he will need to learn only some specific features of this language (if these features are needed him when developing).

References:

1. Ендрю Троелсен. Язык программирования C# 5.0 и платформа .NET 4.5. 6-е издание. – Вильямс, 2013. – 1311 с. : ил.

2. Джеффрі Ріхтер CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / пер. Е. Матвеева. – 4-е изд. – Санкт-Петербург : Питер, 2017. – 895 с. : ил., табл.

3. Захаров В.Б., Мальковський М.Г., Мостяєв А.И. Проблемы выбора языков программирования при разработке кроссплатформенных приложений // International Journal of Open Information Technologies. – 2017. – Т. 5, № 7. – С. 29-37.

4. Ecma-international [Електронний ресурс]. – Режим доступу: https:// www.ecma-international.org/publications/standards/Ecma-335.htm.