

NEW PROGRAMMING LANGUAGE GENERATION

Digital Age or Computer Age is characterized by a rapid development and fast changes in this field due to the world computerization. Programming languages play a great importance here as all programs and different apps are written using them. Lots languages are used for various purposes. But why choose MacBook? This question often comes on nowadays, but the definite answer is that the Mac is intuitive, moreover, all thanks to the MacOS. It was originally designed in such a way that anyone can figure it out, regardless of whether one has used Apple devices before or not. There is one simple rule that applies to the Mac — do it on a whim and you won't be wrong.

Swift is a new programming language for developing iOS and OS X applications that combines the best features of C and Objective-C, but without the restrictions imposed for the sake of compatibility with C. Swift uses secure programming patterns and adds modern features transforming the creation of applications in a simpler, more flexible process.

Let's consider the difference in the capabilities Swift and Objective-C provide to iOS developers. Of course, developers who are interested in the new language from Apple have already researched lots of concerned materials, but it's necessary to focus on those differences that really affect the development process and the architecture of the application. That is, you should know the differences in order to use the language as efficiently as possible.

First, there are some differences between native and cross platform development (especially for iOS and MacOS). Native development works faster and safer. It is easy to get access to the camera, microphone, accelerometer, geolocation, address book, etc. User has a possibility to work with a native interface. It is easy to update code without rewriting match code.

What about drawbacks? Frequent update force you to rewrite a code many times, but this problem is manageable. Many projects were made on Objective-C and it is easier to support them on this language rather than to rewrite them on Swift. It is a very young and rapidly gaining popularity language of programming and it is supposed to replace Objective-C.

Swift has many benefits as compared to Objective-C and other languages:

1. Swift is very fast. It is faster than Objective-C in 2.5 times, than Python in 8 times, and almost as fast as C++.

2. It is easier for programmers in use. It has simple syntax like in Python, it is easy to navigate in project.

3. Swift has all advantages of OOP (Object Oriented Programming), uses principals DRY (don't repeat yourself) which also helps to write simple and understandable code.

4. It has full iteration with the code written in Objective-C. It will allow you to use two languages at the same time, etc.

As for drawbacks, the work with two languages (Swift and Objective-C) at the same time requires much more time because compiler needs to “build a bridge” between them. And besides these regular updates of Swift will make us to update our code. This problem is partly solved, because old versions of code will be compiled in updated language. Comparing Swift with the most popular languages C++ and Python we can clearly find out that it has more details for the second rather than from the first one. Here is an example of counting the sum of squares of all even numbers from 1 to 101.

<i>C++</i>	<i>Python</i>	<i>Swift</i>
------------	---------------	--------------

<pre>#include <iostream> using namespace std; int main() { int s = 0; for (int i = 0; i < 101; i++) { if (i % 2 == 0) s += i * i; } cout << s; return 0; }</pre>	<pre>s=0 for i in range(101): if i%2==0: s+=i*i print(s)</pre>	<pre>var s = 0 for i in 1...100 { if i % 2 == 0 { s += i * i; } } print(s)</pre>
---	---	--

All programs work similar. They check all numbers from 1 to 100 and if it is divisible by 2 without remainder, squares it and add to the sum which was equal to 0 at the start of the program. But Swift has some interesting details. Operator= for classes works with references of objects and for structures it works with values if objects. Swift inherited some details from C++ like usage of curly braces, from Python — absence of semicolon after each operation. By syntax Swift looks like Python and Ruby more than C++, C# or Java. But syntax is not so important compare to parameters were mentioned before (speed, safety, mobility, etc.).

Swift programming language deserves a great attention, as it's gaining the popularity among programmers. Today Swift can be very useful as for beginners so for senior programmers who are already familiar with the code and have programmed in other languages such as Python, Java, C++, JavaScript and etc.