

Ministry of education of the Republic of Belarus
Higher education establishment
BELARUSSIAN STATE UNIVERSITY OF INFORMATICS
AND RADIOELECTRONICS
Computer systems and networks faculty
Foreign languages department #2

Precis
«Mobile application of displaying and forecasting financial data»

Master course student

Anashkevich P. S. _____

Scientific advisor

Subbotkina I. G. _____

Minsk 2016

TABLE OF CONTENTS

Abstract	3
Introduction	4
1 The growth of mobile	5
2 iOS app development technologies	7
2.1 Programming languages	7
2.2 Databases	9
2.3 Addition tools	9
3 Forecasting methods	10
3.1 Classification of forecasting methods	10
Conclusion	11
References	12
Glossary	13

ABSTRACT

INTRODUCTION

1 THE GROWTH OF MOBILE

For many employees, desktop computing is a thing of the past. Or, at the very least, advances in mobile technology over the past few years have made it easier than ever to work remotely. What once required a computer can now often be done on a smartphone.

The number of hours that people spend online today continues to grow every year. These hours have shifted from desktop computers to mobile phones to a range of other devices. In fact, the amount of time spent on the mobile web increased 575% in around 3 years.

It is not just the amount of time that people spend online that matters. People now have more devices to choose from when they want to go online. As the adoption of mobile devices grows, the amount of time that people spend on them has grown with it. In fact, the amount of time spent on mobile internet has now overtaken the amount of time spent online on a desktop.

These trends do not happen in a vacuum. Mobile internet did not shift from the development BBM to free Netflix streaming overnight. A series of events had to take place in order for mobile internet to be feasible at this level.

First, mobile devices needed to be more accessible. American carriers took care of that by leasing these phones alongside service contracts. This had benefits for everyone because consumers no longer had to shell out hundreds of dollars up front and carriers could make a bundle by locking those consumers into contracts.

Second, the availability of mobile internet had to improve. Even the youngest adopters can remember a time when mobile internet was good for chatting and email but still couldn't beat wifi. Carriers knew that the cost of selling more data would be investment in better data services including better technology and more towers.

Third, mobile devices and data had to be useful in a way that was not currently available on the PC. Thus, the mobile app was born. Today, mobile apps are changing the game completely and users are abandoning their mobile browsers for apps. Google reported that its Android market received 50 billion app downloads in 2013 alone. Some reports suggest that this will grow to 268 billion downloads by 2017.

Apps are not just a new trend. Nearly 80% of mobile hours are spent using apps and total app usage grew 76% in 2014 alone. This is because apps offer a unique value proposition that merges convenience and sleek design. For many customers, this value proposition is worth enough to drag even the most dedicated PC user onto their

phone or tablet. [<https://www.cleverism.com/mobile-usage-implications-for-growing-your-business/>]

Worldwide combined shipments of devices (PCs, tablets, ultramobiles and mobile phones) are expected to reach 2.4 billion units in 2016, a 1.9 percent increase from 2015, according to Gartner, Inc. End-user spending in constant U.S. dollars is expected to decline 0.5 percent for the first time.

Таблица 1.1 – Worldwide devices shipments by device type, 2015-2018
(millions of units)

Device Type	2015	2016	2017	2018
Traditional PCs (Desk-Based and Notebook)	246	232	226	219
Ultramobiles (Premium)	45	55	74	92
PC Market	290	287	299	312
Ultramobiles (Basic and Utility)	196	195	196	198
Computing Devices Market	486	482	495	510
Mobile Phones	1 910	1 959	1 983	2 034
Total	2396	2441	2 478	2 545

The device market in 2016 will continue to be impacted by country-level economic conditions. «It's clear that vendors can no longer market their products with the mind of only targeting the mature and emerging markets,» said Ranjit Atwal, research director at Gartner. «Driven by economic variations the market is splitting into four categories: economically challenged mature markets, economically stable mature markets and the same for emerging markets. Russia and Brazil will fall into the category of economically challenged emerging markets while India will be stable, and Japan will belong to the economically challenged mature market.» [<http://www.gartner.com/newsroom/id/3187134>]

2 IOS APP DEVELOPMENT TECHNOLOGIES

iOS is the operating system that runs on iPad, iPhone, and iPod touch devices. The operating system manages the device hardware and provides the technologies required to implement native apps. The operating system also ships with various system apps, such as Phone, Mail, and Safari, that provide standard system services to the user.

2.1 Programming languages

There are two languages which can be used to create iOS applications. Consider each of them.

Objective-C is a general-purpose, object-oriented programming language that adds Smalltalk-style messaging to the C programming language. This is the main programming language used by Apple for the OS X and iOS operating systems and their respective APIs, Cocoa and Cocoa Touch.

Let's take a look at the advantages of Objective-C.

First of all, code, written in Objective-C and C, can be used in Swift, but not vice versa. For example, the existing solutions and libraries were all created in Objective-C but are used in Swift without any issues. You have probably seen Apple's Top 25 Favourite iOS Apps 2015. If you dig deeper you will be surprised to learn how many of them were built in Objective-C despite of the company's dedication to Swift. Still it is not the reason to refuse using this modern programming language from Apple. On the contrary, they encourage developers to switch to Swift. Maybe 2016 will end up with a much more different rank.

Secondly, C++ code cannot be used in Swift. It is, however, possible to use Objective-C++.

Thirdly, Objective-C can be compiled into static libraries and dynamic frameworks, while Swift can be compiled only into dynamic frameworks.

The syntax of Objective-C is stable while Swift syntax is still improving. Nevertheless, Swift 3.0 promises to bear breaking changes that will not be followed by other great ones. This means that developers will have to convert code from Swift 2.1 to Swift 2.2 and only then to Swift 3.0. Experience shows that sometimes converting runs unsmoothly and it is necessary to rewrite pieces of code. Not the most convenient thing, of course, but we have to understand that Swift is a new programming language and updates are unavoidable.

Apps written in Swift before Version 3.0 will be 10-20 Mb bigger in size than the ones built in Objective-C. Everything depends on architecture here.

The reason is that all Swift runtime libraries have to be included in an app if it contains at least 1 line of Swift code. The creators promise to stabilize API in Swift 3.0 so there will be no need doing this anymore.

Swift 2.1 compiler is unstable. It crashes sometimes which never happens in Objective-C. Swift code is compiled slower but this is because Swift compiler knows more about variables, than Objective-C compiler. It analyzes code deeper, in particular type inference, generics, and protocols with associated types. This issue will be possibly and hopefully solved in Swift 2.2. By the way, 97% of all existing crashes are fixed already.

Xcode does not support refactoring of Swift code. It should be done manually. This feature is available only for Objective-C.

Now let us consider the Swift language.

Swift is a modern programming language that is fast, safe and interactive. Swift is a powered programming tool that is used to develop applications for iOS, OS X, and watchOS. Using swift one can create codes easily as it has many feature that have fun and it is interactive, the syntax is short and expressive. With the use of Swift one can run apps lightning fast speed as it works side-by-side with Objective-C.

Following are the main features of Swift language.

Error handling model. Swift provide an advance error handling model that is clear, expressive syntax of exceptional handling. Developers can create their own custom error names so that it can describe clear error cases.

Syntax Improvements. Now using Swift you can create more expressive codes as it has improved a lot across the language. The SDK has employed new Objective-C features such as generics and nullability annotation to keep the codes clean and safe.

Open Source. Later this year Swift will be termed as open source. It is a unique combination of power, elegance and safety that has ability to move entire software industry forward.

Modern. Consider it to be the latest research of programming language, combined with a vast experience of developing Apple platforms. Named parameters brought forward from Objective-C are expressed in a clean syntax that makes APIs in Swift even easier to read and maintain.

Interactive playgrounds. Using playgrounds one can write swift codes with incredibly simple and fun. Results are appeared immediately as you type the codes. The result view can display graphics, lists of results, or graphs of a value over time. You can open the Timeline Assistant to watch a complex view evolve and animate, great for experimenting with new UI code, or to play an animated SpriteKit scene as you code it.

Designed for Safety. It eliminates the entire class of unsafe code. Variables are always initialized before use, arrays and integers are checked for overflow, and memory is managed automatically.

Fast and powerful. Swift is a fast compiling and executable programming language. It use high performance LLVM compiler, Swift code is transformed into optimized native code that gets the most out of modern hardware.

2.2 Databases

Core Data

Realm

2.3 Addition tools

3 Forecasting methods

3.1 Classification of forecasting methods

CONCLUSION

REFERENCES

- [1] Михайловский, В. С. Неомарксизм вчера, сегодня, завтра [Текст] / В. С. Михайловский // Социология. — 2015. — № 4. — С. 37–43.
- [2] Степин, В. С. Эпоха перемен и сценарии будущего [Текст] / В. С. Степин. — [Б. м. : б. и.], 1996.
- [3] Букова, И. А. Социализм в XXI веке: миф или реальность? [Текст] / И. А. Букова // Социально-гуманитарные знания. — 2010. — № 3. — С. 335–341.
- [4] Неомарксизм и проблемы социологии культуры [Текст] / Ю. Н. Давыдов, О. С. Ли, С. М. Митина, А. И. Чупрынин. — [Б. м.] : Наука, 1980.
- [5] Степин, В. С. Философия науки и техники [Текст] / В. С. Степин, В. Г. Горохов, М. А. Розов. — [Б. м.] : Гардарики, 1999.
- [6] Tormey, S. Key Thinkers: From Critical Theory to Post-Marxism [Текст] / S. Tormey, J. Townshend. — [Б. м.] : SAGE Publications Ltd, 2006.

GLOSSARY

Test Glossary section.