

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ  
Кафедра веб-технологий и компьютерного моделирования**

**ПОЛЕВИКОВ**  
Алексей Юрьевич  
**РАЗРАБОТКА JAVA-ПРИЛОЖЕНИЙ ДЛЯ СРЕДСТВ МОБИЛЬНОЙ  
СВЯЗИ**  
Дипломная работа

Научный руководитель:  
канд. физ.-мат. наук,  
доцент В.С. Романчик

Допущен к защите

«\_\_\_» \_\_\_\_\_ 2014 г.

Зав. кафедрой веб-технологий и компьютерного моделирования  
кандидат физико-математических наук, доцент В.С.Романчик

Минск, 2014

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	6
ГЛАВА 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ .....	7
1.1    Общее понятие разработки приложений для мобильных устройств .....	7
1.2    Среды выполнения .....	8
1.3    Тестирование приложений .....	9
1.4    Эмуляторы мобильных устройств .....	10
1.5    Магазины приложений .....	11
1.6    Платформы для разработки .....	12
1.7    Android .....	13
1.8    iOS .....	16
1.9    Windows Phone .....	18
1.10    Java ME .....	20
1.11    Codename One SDK .....	23
ГЛАВА 2 РАЗРАБОТКА ПРИЛОЖЕНИЯ .....	24
2.1    Обзор языка Java .....	24
2.2    Пользовательский интерфейс .....	26
ГЛАВА 3 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ .....	31
3.1    Класс NewGridAdapter .....	31
3.2    Класс MemoryActivity .....	32
3.3    Класс MemoryStart .....	33
3.4    GUI .....	34
ЗАКЛЮЧЕНИЕ .....	35
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ .....	36

## РЕФЕРАТ

Дипломная работа содержит: 36 страниц, 8 иллюстраций (рисунков), 7 использованных литературных источников.

Ключевые слова: JAVA, МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ, СРЕДА РАЗРАБОТКИ, ANDROID.

В ходе данной дипломной работы была рассмотрена специфика использования языка Java при разработке мобильных приложений. Было проведено сравнение актуальных мобильных платформ а также инструментария для разработки под эти платформы. Было разработано приложение на Java под платформу Android, в качестве приложения была выбрана игра для развития и укрепления памяти. Были поставлены цели и задачи, исходя из полученной информации и проведенного анализа.

Приложение было написано в среде Eclipse с использованием необходимого набора для разработки под выбранную платформу. В процессе создания приложения были применены полученные на практике знание в сфере информационных технологий.

В первой главе рассматривается краткий обзор существующих технологий разработки приложения, проводится сравнительный анализ существующих средств.

Во втором разделе описывается язык разработки и в общих чертах рассматривается приложение.

В третьей главе приводится описание программной реализации разрабатываемой системы.

## РЭФЕРАТ

Дыпломная праца змяшчае: 36 старонак, 8 ілюстрацый (малюнкаў), 7 выкарыстаных літаратурных крыніц.

Ключавыя словы: JAVA, МАБІЛЬНЫЯ ДАДАТКІ, асяроддзе распрацоўкі, ANDROID.

У ходзе дадзенай дыпломнай працы была разгледжана спецыфіка выкарыстання мовы Java пры распрацоўцы мабільных прыкладанняў. Было праведзена параўнанне актуальных мабільных платформаў а таксама прыстасаванняў для распрацоўкі пад гэтыя платформы. Было распрацавана прыкладанне на Java пад платформу Android, у якасці прыкладання была абраная гульня для развіцця і ўмацавання памяці. Былі пастаўлены мэты і задачы, зыходзячы з атрыманай інфармацыі і праведзенага аналізу.

Прыкладанне было напісана ў асяроддзі Eclipse з выкарыстаннем неабходнага набору для распрацоўкі пад выбраную платформу. У працэсе стварэння прыкладання былі ўжытыя атрыманыя на практыцы веды ў сферы інфармацыйных тэхналогій.

У першай чале разглядаецца кароткі агляд існуючых тэхналогій распрацоўкі прыкладання, праводзіцца параўнальны аналіз існуючых сродкаў.

У другім раздзеле апісваецца мова распрацоўкі і ў агульных рысах разглядаецца праграма.

У трэцяй чале прыводзіцца апісанне праграмнай рэалізацыі распрацаванай сістэмы.

## **ABSTRACT**

Thesis project consists of: 36 pages, 8 figures (drawings), 7 used literature sources.

Keywords: JAVA, mobile applications development environment, ANDROID.

In the course of this thesis were discussed specifics of the Java language usage in mobile applications development. A comparison of current mobile platforms as well as developing tools for these platforms was made. Java application was developed under the Android platform, the game for development and strengthening of memory was chosen as application. Were set goals and objectives, based on the received information and analysis.

Application was written in Eclipse with the necessary set to develop for the selected platform. In the process of creating applications received in practice knowledge in the field of information technology have been applied.

The first chapter represents short review of existing technology application development, a comparative analysis of existing tools.

The second section describes the development language and broadly considered application.

The third chapter describes the software implementation of the developed system.

## ВВЕДЕНИЕ

Нужда человека всегда оставаться в курсе событий создает большое подспорье для создания все новых мобильных устройств и гаджетов. Ввиду неудобства использования стационарных компьютеров и ноутбуков на первый план все чаще выходят мобильные телефоны и планшеты, которые также находятся под управлением операционной системы.

С учетом развития информационных технологий, из года в год появляются все более мощные мобильные устройства. Следовательно, существенно повышаются требования к приложениям, предназначенным для этих устройств. Существуют различные мобильные платформы, и для каждой имеется свой инструментарий разработки. Могут быть разработаны приложения, которые будут работать на нескольких платформах, то есть кроссплатформенные. Однако нужно понимать, что возможности программного обеспечения, написанного для определенной платформы, шире, поэтому логичнее будет устанавливать соответствующее ПО на соответствующую платформу.

С момента появления первых мобильных приложений Java всегда был одним из самых популярных языков, используемых для их разработки. Таким он остается и по сей день, несмотря на появление новых платформ.

Целью дипломной работы является написание приложения для мобильных устройств используя язык программирования Java.

В качестве платформы была выбрана Android, разработка велась в среде Eclipse с использованием плагина ADT и комплекта Android SDK.

Дипломная работа состоит из введения, 3 глав, заключения и списка использованной литературы.

# **ГЛАВА 1**

## **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

### **1.1 Общее понятие разработки приложений для мобильных устройств**

Разработка приложений для мобильных устройств — это процесс, при котором приложения разрабатываются для небольших портативных устройств таких как КПК, смартфоны или сотовые телефоны. Эти приложения могут быть предустановлены на устройство в процессе производства, загружены пользователем с помощью различных платформ для распространения ПО или являться веб-приложениями, которые обрабатываются на стороне клиента (JavaScript) или сервера.

## 1.2 Среда выполнения

Android, iOS, BlackBerry, Open webOS, Symbian OS, Bada от Samsung, и Windows Mobile поддерживают стандартные бинарные файлы приложений как на персональных компьютерах с кодом выполняющимся на процессоре определенного формата (в основном используется архитектура ARM). Windows Mobile может быть скомпилирована для архитектуры x86 для отладки на ПК без эмуляции процессора, а также поддерживает формат Portable Executable (PE) связанный с .NET Framework. Windows Mobile, Android, HP webOS и iOS предоставляют бесплатные SDK и интегрированные среды разработки для разработчиков.



### **1.3 Тестирование приложений**

Список инструментов тестирования приложений для мобильных устройств:

- Эмуляторы
- Облачные платформы устройств
- Автоматизированное воспроизведение скриптовых тестов
- Нагрузочное тестирование
- Манкитестинг
- Сборщики статистики

## 1.4 Эмуляторы мобильных устройств

Сперва приложение тестируется в среде разработки с использованием эмулятора. После этого приложение тестируется на устройстве. Эмуляторы являются простым способом тестировать приложение на мобильном телефоне, не используя его физически. Ниже представлен список доступных инструментов для тестирования приложений среди самых популярных мобильных операционных систем:

- Google Android Emulator

Android Эмулятор запускается на Windows как отдельное приложение без необходимости полностью загружать и устанавливать Android SDK.

- Официальный Android SDK Emulator

Включает в себя эмулятор мобильного устройства, который реализует все аппаратные и программные особенности типичного устройства.

- MobiOne

MobiOne Developer это mobile Web IDE для Windows помогающее разработчику программировать, тестировать, отлаживать упаковывать и внедрять мобильные веб-приложения на устройства, такие как iPhone, BlackBerry, устройства на Android и Palm Pre.

- TestiPhone

Основанный на веб-браузере симулятор для быстрого тестирования веб-приложений для iPhone. Работает с использованием Internet Explorer 7, Firefox 2 и Safari 3.

- iPhoney

Предоставляет точную среду веб-браузера, разработана Safari. Может быть использована для разработки веб-сайтов для iPhone. Не является эмулятором iPhone. iPhoney запускается только на Mac OS X 10.4.7 и выше.

- BlackBerry Simulator

Существует множество официальных эмуляторов BlackBerry. С любым из них возможна проверка того, как ПО, экран, клавиатура устройства будут работать с приложением.

## 1.5 Магазины приложений

Различные инициативы существуют от мобильных операторов и от производителей. Разработчики приложений могут предлагать и публиковать свои программы в магазинах приложений, с возможностью зарабатывать от распределения доходов по продажам. Самым известными являются App Store Apple, где только одобренные приложения могут распространяться и запускаться на iOS устройствах (также известно как walled garden), и Android Market Google, приложения в котором работают на устройствах с Android OS. HP / Palm также имеют Palm App Catalog где пользователи устройств на HP / Palm webOS могут загружать приложения непосредственно с устройства или отправить ссылку на приложение с помощью уникального метода распространения. Мобильные операторы Telefonica Group и Telecom Italia запустили межплатформенный магазин приложений для своих абонентов. Производитель мобильных устройств Nokia запустил Ovi app store для смартфонов Nokia.[1]

## 1.6 Платформы для разработки

Каждая из платформ для мобильных приложений имеет интегрированную среду разработки, предоставляющую инструменты, позволяющие разработчику программировать, тестировать и внедрять приложения на целевую платформу. Для сравнения мобильных приложений на Java с приложениями, написанными на других языках, сравним платформу Android с iOS и Windows Phone (Рисунок 1.1).

**Top Five Smartphone Operating Systems, Shipments, and Market Share, 4Q 2013 (Units in Millions)**

Operating System	4Q13 Shipment Volumes	4Q13 Market Share	4Q12 Shipment Volumes	4Q12 Market Share	Year-Over-Year Change
Android	226.1	78.1%	161.1	70.3%	40.3%
iOS	51.0	17.6%	47.8	20.9%	6.7%
Windows Phone	8.8	3.0%	6.0	2.6%	46.7%
BlackBerry	1.7	0.6%	7.4	3.2%	-77.0%
Others	2.0	0.7%	6.7	2.9%	-70.1%
<b>Total</b>	<b>289.6</b>	<b>100.0%</b>	<b>229.0</b>	<b>100.0%</b>	<b>26.5%</b>

Рисунок 1.1 Рынок мобильных платформ по данным IDC

## 1.7 Android

Язык Java активно используется для создания мобильных приложений под операционную систему Android. При этом программы компилируются в нестандартный байт-код, для использования их виртуальной машиной Dalvik. Для такой компиляции используется дополнительный инструмент, а именно Software Development Kit, его разработали представители компании Google.

Разработку приложений можно вести в среде Eclipse, используя при этом плагин — Android Development Tools (ADT) или в IntelliJ IDEA. Версия JDK при этом должна быть 5.0 или выше.

С точки зрения программиста, Android — платформа, абстрагирующая разработчика от ядра и позволяющая ему создавать код на Java. Android обладает несколькими полезными возможностями.

Во-первых, это фреймворк, предлагающий большой набор API для создания различных типов приложений и, кроме того, обеспечивающий возможности повторного использования и замены компонентов, которые предлагаются платформой и сторонними приложениями.

Во-вторых, наличие виртуальной машины Dalvik, отвечающей за запуск приложений на Android.

Для работы над приложениями доступно множество библиотек: Bionic (библиотека стандартных функций, несовместимая с glibc); мультимедийные библиотеки на базе PacketVideo OpenCORE (поддерживают такие форматы, как MPEG-4, H.264, MP3, AAC, AMR, JPEG и PNG); SGL (движок двухмерной графики); OpenGL ES 1.0ES 2.0 (движок трёхмерной графики); Surface Manager (обеспечивает для приложений доступ к 2D/3D); WebKit (готовый движок для веб-браузера; обрабатывает HTML, JavaScript); FreeType (движок обработки шрифтов); SQLite (легковесная СУБД, доступная для всех приложений); SSL (протокол, обеспечивающий безопасную передачу данных по сети). По сравнению с обычными приложениями Linux приложения Android подчиняются дополнительным правилам: Content Providers — обмен данными между приложениями; Resource Manager — доступ к таким ресурсам, как файлы XML, PNG, JPEG; Notification Manager — доступ к строке состояния; Activity Manager — управление активными приложениями.

Google предлагает для свободного скачивания инструментарий для разработки (Software Development Kit), который предназначен для x86-машин под операционными системами Linux, Mac OS X (10.4.8 или выше), Windows

XP, Windows Vista и Windows 7. Для разработки требуется JDK 5 или более новый.

Разработку приложений для Android можно вести на языке Java (не ниже Java 1.5). Существует плагин для Eclipse — Android Development Tools (ADT), предназначенный для Eclipse версий 3.3—3.7. Также существует плагин для IntelliJ IDEA, облегчающий разработку Android-приложений, и для среды разработки NetBeans IDE, который, начиная с версии NetBeans 7.0, перестал быть экспериментальным, хоть пока и не является официальным. Кроме того, существует Motodev Studio for Android — комплексная среда разработки на базе Eclipse, позволяющая работать непосредственно с Google SDK.

В 2009 году в дополнение к ADT был опубликован Android Native Development Kit (NDK) — пакет инструментариев и библиотек, позволяющий реализовать часть приложения на языке C/C++. NDK рекомендуется использовать для разработки участков кода, критичных к скорости.

В 2013 году Google представила новую среду разработки Android Studio, основанная на IntelliJ IDEA от JetBrains.

В 2013-м году состоялся релиз Embarcadero RAD Studio — XE5. Возможность разработки нативных приложений для платформы Android.

Процесс создания Android приложения не требует дополнительных устройств, кроме, собственно, Android устройства (в принципе, можно обойтись и эмулятором).

По данным исследовательской компании IDC, операционная система от Google продолжает лидировать как на российском (77%), так и на мировом (78%) рынке смартфонов. Причина — в большом количестве производителей — здесь и сверхдешевые телефоны, и мощные смартфоны за несколько тысяч долларов.

Впрочем, столь большой охват порождает и основные недостатки разработки для Android. Если ориентироваться на широкий спектр устройств, придется учитывать разную производительность, бесконечное количество размеров экрана и память. Как следствие — увеличенные затраты на проектирование нескольких интерфейсов и дополнительное тестирование. Стоимость разработки в таком случае увеличивается пропорционально количеству поддерживаемых устройств.

Кроме того, обладатели Android не любят платить за приложения. Причина отчасти кроется в легкости установки взломанной программы — чаще всего достаточно найти нужный файл на одном из форумов. С другой стороны, покупатели дешевых моделей изначально не нацелены на большие траты в магазинах приложений.

Зарегистрироваться в Google Play может любой разработчик. Для этого нужно заплатить \$25 с помощью банковской карты, загрузить скриншоты приложения, его описание и код. У Google нет столь жестких требований к качеству приложения, как у Apple. Это снижает и риски затрат на доработку. Кроме того, есть альтернативные магазины приложений, такие как Amazon Apps, Samsung Apps, Opera Apps, SlideMe и Yandex.Store. У каждого своя политика модерации, однако присутствие в них дает ощутимый приток пользователей.[2]

## 1.8 iOS

iOS (до 24 июня 2010 года — iPhone OS) — мобильная операционная система, разрабатываемая и выпускаемая американской компанией Apple. Была выпущена в 2007 году; первоначально — для iPhone и iPod touch, позже — для таких устройств, как iPad и Apple TV. В отличие от Windows Phone и Google Android, выпускается только для устройств, производимых фирмой Apple.

Для разработки приложений используется iOS SDK, язык разработки — Objective-C.

iOS SDK (ранее iPhone SDK) — комплект средств разработки для iOS, выпущенный в марте 2009 года корпорацией Apple. iOS SDK выпускается только для Mac OS X.

Разработчики могут устанавливать любую цену, превышающую минимальную установленную, за их приложения, которые будут распространяться через App Store, из которой они будут получать 70%. Кроме того, они могут распространять своё приложение бесплатно, в этом случае они должны платить только членские взносы, которые составляют 99 долларов в год.

Apple iOS использует вариант того же ядра XNU, что и Mac OS X. SDK разбита на следующие наборы:

- Cocoa Touch
- Мультиязычное управление
- Поддержка акселерометра
- Иерархия видов
- Локализация
- Поддержка камеры
- Мультимедиа
- OpenAL
- Микширование и запись аудио
- Воспроизведение видео
- Форматы изображений
- Кварц
- Анимационное ядро
- OpenGL ES
- Сервисное ядро
- Сеть



- Встроенная база данных SQLite
- Локационное ядро
- Многопоточность
- CoreMotion
- Ядро OS X
- TCP/IP
- Сокеты
- Управление питанием
- Файловая система
- Безопасность

По популярности iOS сильно уступает Android. Между тем, главным преимуществом компании Apple является состоятельность ее аудитории. При столь маленьком объеме рынка (около 12%), на устройства iOS приходится половина всех доходов от продажи приложений — \$6,4 млрд в год.

В среднем за одно скачивание в App Store разработчик получает \$0,24. Если учесть еще и высокую стоимость самой дешевой модели iPhone, покупатели iOS-устройств становятся хорошей целевой аудиторией для создателей приложений. Кроме того, устройства Apple не так просто взломать. Это еще одна причина высокой покупательской способности на платформе.

Сами разработчики отдают предпочтение платформе Apple. Отчасти это связано с тем, что при проектировании нужно учитывать всего несколько форматов дисплеев. Возможно еще и потому, что компания смогла упаковать инструменты разработки в более качественные продукты, по сравнению с конкурентами.

С основными проблемами можно столкнуться при публикации созданного приложения. Аккаунт iOS-разработчика Apple стоит \$99 в год. За эти деньги вы получите возможность загрузить приложение для его последующей модерации.

С суровой политикой Apple может столкнуться любая компания. Строгость корпорации к внешним разработчикам записывают в главные риски проектирования. Лучше заранее быть готовым к дополнительным тратам на доработку.[3]

## 1.9 Windows Phone

Windows Phone — мобильная операционная система, разработанная Microsoft, вышла 11 октября 2010 года. 21 октября начались поставки первых устройств на базе новой платформы.

Все программы для Windows Phone 7 создаются с использованием управляемого кода .NET. В настоящее время C# — единственный поддерживаемый язык программирования. Свободно доступный для загрузки Microsoft Visual Studio 2010 Express для Windows Phone включает XNA Game Studio 4.0 и экранный эмулятор телефона, а также интегрируется с Visual Studio 2010.

Визуальные элементы и анимация для приложений Silverlight могут создаваться в Microsoft Expression Blend. Платформы Silverlight и XNA для Windows Phone 7 имеют ряд общих библиотек, то есть некоторые библиотеки XNA могут использоваться в программе Silverlight и наоборот. Но нельзя создавать программу, сочетающую в себе визуальные элементы обеих платформ.

Как правило, Silverlight используется для программ, которые можно классифицировать как приложения или утилиты. Описание компоновки элементов управления и панелей пользовательского интерфейса в этих программах выполняется с помощью Расширяемого языка разметки приложений (Extensible Application Markup Language, XAML). В файлах выделенного кода могут реализовываться операции по инициализации и некоторая логика, но основным их назначением является обработка событий элементов управления. Silverlight позволяет реализовывать в Windows Phone стиль насыщенных интернет-приложений (Rich Internet Applications, RIA), включая мультимедиа и Веб. Для Windows Phone создана версия Silverlight 3, в которую не вошли некоторые возможности, не подходящие для телефона, но компенсированы рядом дополнений.

Главное назначение XNA — создание высокопроизводительных игр. Для 2D-игр спрайты и подложки описываются с помощью растровых изображений; для 3D игр создаются трехмерные модели. Действие игры, включающее перемещение графических объектов по экрану и запрос пользовательского ввода, обрабатывается встроенным игровым циклом XNA.

Удобно провести границы и принять, что Silverlight используется для приложений, а XNA — для игр, но это не должно накладывать ограничения. Вне всяких сомнений, Silverlight может применяться для реализации игр, и

традиционные приложения могут создаваться на XNA, хотя это будет сопряжено со значительными трудностями.

Silverlight подходит для игр с небольшими требованиями по графике, либо использующих векторную, а не растровую графику, либо темп которых определяется реакцией пользователя, а не таймером.

Windows Phone SDK включает в себя следующие компоненты.

- Экспресс-выпуск Microsoft Visual Studio 2010 для Windows Phone
- Эмулятор Windows Phone
- Сборки пакета Windows Phone SDK 7.1
- Пакет SDK и DRT для Silverlight 4
- Расширения пакета Windows Phone SDK 7.1 для XNA Game Studio 4.0
- Пакет Microsoft Expression Blend SDK для Windows Phone 7
- Пакет Microsoft Expression Blend SDK для ОС Windows Phone 7.1
- Клиент служб данных WCF для Windows Phone
- Пакет Microsoft Advertising SDK для Windows Phone

Лидером по количеству загрузок среди всех платформ однозначно является магазин приложений Google с прибылью от закачек \$1,2 млрд в год. Число загрузок из Windows Phone Store в несколько раз меньше, при этом доход от них составляет \$0,95 миллиарда.

Получается, что в среднем разработчик зарабатывает около \$0,23 за одно скачивание из Windows Phone Store (прибыль Android-разработчика за одно скачивание его приложения составляет \$0,04).

В достоинства Windows Phone Store можно записать и слабую конкуренцию внутри магазина. У большинства компаний разработка приложения для Windows Phone стоит на третьем месте (сразу после iOS и Android), поэтому часто до нее просто не доходят. Например, Instagram для WP8 вышел несколько месяцев назад и только в бета-версии.[4]

## 1.10 Java ME

Нельзя не упомянуть про платформу JavaME. Несмотря на то что в настоящее время она считается устаревшей, из диаграммы видно, что она по-прежнему занимает значительную долю на рынке мобильных устройств (Рисунок 1.2).

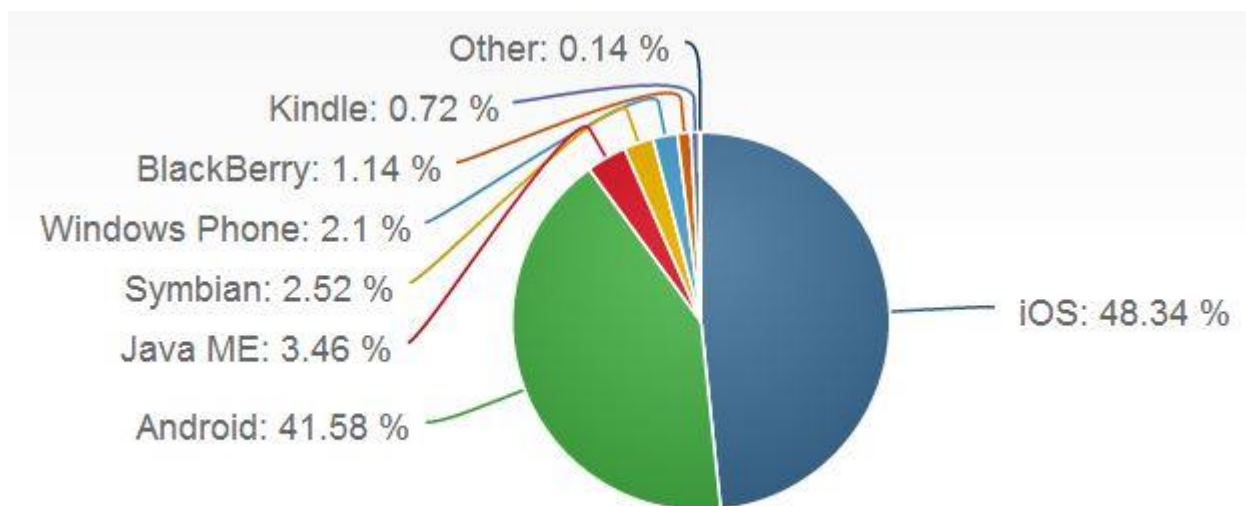


Рисунок 1.2 Распределение рынка мобильных ОС по данным netmarketshare

Главная проблема мобильных телефонов — все они работают под управлением прошивки. Если в смартфоне функциональность устройства можно расширять (на том смартфоны и стоят), то в обычных мобилниках это невозможно. Тут и вступает в дело Java.

Идея состоит в том, что команды отдаются не напрямую процессору, а виртуальной Java-машине (JVM — Java Virtual Machine). На Java ME ее еще называют KVM, Kilobyte Virtual Machine. Вместо команд процессора программа на Java представляет собой байт-код — команды, которые и должна выполнять Java-машина. Для того чтобы программа заработала, достаточно, чтобы на системе была установлена эта самая Java-машина. На компьютерах она ставится отдельной программой, а в большинстве телефонов является частью прошивки.

Для программ, которые рассчитаны на Java ME, есть особое название — мидлет. Их очень часто путают с апплетами, но это совершенно разные понятия. Апплеты — это программы на Java, которые рассчитаны на запуск в рамках других программ, например в интернет-браузере, а мидлет — это

вполне самостоятельная программа. Игра, «читалка», ICQ-клиент — все что угодно.

Мобильные программы распространяются не в виде разрозненных файлов, а в виде специальных архивов и файлов описания. Это файлы JAR и JAD. JAR расшифровывается как Java Archive. На самом деле это самый обычный архив Zip, просто с другим расширением. В нем хранятся все файлы программы: .class (они содержат байт-код), файлы ресурсов (например, картинки или звуки) и файл-манифест. Последний описывает программу: название, производитель, версия и другие данные. JAD — это файл описания (расшифровывается как Java Application Descriptor). Он содержит все те же сведения, что и файл манифеста, плюс размер архива и путь к нему (URL-адрес). Для чего же он нужен, если вся информация уже содержится в файле манифеста? А для того, чтобы можно было посмотреть сведения о мидлете, не качая архив, который может быть достаточно велик.

Понятно, что для установки обязательно нужен файл JAR. JAD-файл на некоторых старых телефонах тоже требовался, но практически любой современный телефон без него спокойно обходится.

Одно из главных понятий, которые есть в программировании, — это API (Application Programming Interface), интерфейс прикладного программирования. Это набор команд, которые программа может отдать устройству. Например, большинство современных телефонов обладает камерой. Но одного ее наличия для съемки из Java-приложения недостаточно. Нужно, чтобы был API управления камерой — иначе телефон просто «не поймет», чего от него хотят. API и определяет функциональность устройства.

Самый базовый API, на котором строятся все остальные, — это либо CDC (Connected Device Configuration), либо CLDC (Connected Limited Device Configuration). Оба представляют собой сильно урезанные наборы команд из «большой джавы». CDC предназначается для смартфонов, коммуникаторов и КПК (как более мощный), а для мобильных остается CLDC, конфигурация попроще. Сейчас существуют две версии CLDC: 1.0, которая уже практически нигде и не используется, и 1.1, главное отличие которой от предыдущей версии — поддержка расчетов с плавающей точкой. Обе из них созданы уже достаточно давно, но замену им пока почему-то не готовят.

Поскольку мобильники сильно отличаются по устройству от компьютеров, понадобился API, который может дать программисту средства сделать удобные меню, хранить настройки приложений и другие специфические для мобильных возможности. Эту задачу берет на себя API под названием MIDP.

Это слово, наверное, видел каждый, кто брал в руки каталог телефонов. Расшифровывается оно как Mobile Information Device Profile.

На данный момент существует несколько версий. MIDP 1.0 создан очень и очень давно, в 2000 году. Он накладывал много ограничений на программы — его возможности были очень небольшими. Поэтому в 2002 была выпущена новая версия, MIDP 2.0. Эта версия используется и по сей день, причем практически во всех новых телефонах. Так что сейчас слова «Java ME» и «MIDP 2.0» — почти синонимы. По сравнению с предшественницей, эта версия дает куда больше возможностей: приемлемое звуковое сопровождение, расширенные сетевые возможности, богатые средства для создания интерфейса и игровой графики. Именно MIDP 2.0 дал толчок к развитию мобильного игрового устройства.

Стоит также упомянуть MIDP 2.1, который был разработан в 2006 году. Он не дает каких-либо новых возможностей, зато в этой версии уточнены некоторые особенности реализации Java на телефонах. Ее уже встраивают в конкретные телефоны, хоть это и не афишируется.[5]

## 1.11 Codename One SDK

Codename One позиционирует себя как уникальный SDK, позволяющий разработчикам создавать нативные мобильные приложения на Java используя единую базу кода.

Компания из Израиля говорит, что с помощью Codename One создано уже более 1 000 приложений, что делает платформу "одной из самых быстрорастущих".

Codename One основана на Java и может создавать нативные приложения для iOS, Android, BlackBerry и Windows Phone 7. Идея состоит в том, что вся логика программируется один раз и сохраняется в каждом приложении, а интерфейсы каждый раз создаются отдельно для разных систем. Приложение потом собирается из исходников в нативное приложения на облачном сервисе. Это, по словам создателей, поможет избежать фрагментации, связанной с различными платформами.[6]

## ГЛАВА 2 РАЗРАБОТКА ПРИЛОЖЕНИЯ

### 2.1 Обзор языка Java

Java — объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой виртуальной Java-машине вне зависимости от компьютерной архитектуры. Дата официального выпуска — 23 мая 1995 года.

Программы на Java транслируются в байт-код, выполняемый виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор.

Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности благодаря тому, что исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером) вызывают немедленное прерывание.

Часто к недостаткам концепции виртуальной машины относят то, что исполнение байт-кода виртуальной машиной может снижать производительность программ и алгоритмов, реализованных на языке Java. В последнее время был внесен ряд усовершенствований, которые несколько увеличили скорость выполнения программ на Java:

- применение технологии трансляции байт-кода в машинный код непосредственно во время работы программы (JIT-технология) с возможностью сохранения версий класса в машинном коде,
- широкое использование платформенно-ориентированного кода (native-код) в стандартных библиотеках,
- аппаратные средства, обеспечивающие ускоренную обработку байт-кода (например, технология Jazelle, поддерживаемая некоторыми процессорами фирмы ARM).



По данным сайта [shootout.alioth.debian.org](http://shootout.alioth.debian.org), для семи разных задач время выполнения на Java составляет в среднем в полтора-два раза больше, чем для C/C++, в некоторых случаях Java быстрее, а в отдельных случаях в 7 раз медленнее. С другой стороны, для большинства из них потребление памяти Java-машиной было в 10-30 раз больше, чем программой на C/C++. Также примечательно исследование, проведённое компанией Google, согласно которому отмечается существенно более низкая производительность и большее потребление памяти в тестовых примерах на Java в сравнении с аналогичными программами на C++.[7]

## 2.2 Пользовательский интерфейс

При запуске игры появляется меню с двумя пунктами: Старт и Выход (Рисунок 2.1).



Рисунок 2.1 Главное меню

При нажатии на пункт меню Старт запускается само приложение (Рисунок 2.2). На экране появляется массив изображений размера 6 на 6, пользователь видит только их обратную сторону. Также в верхнем левом углу экрана располагается счетчик ходов, а в правом находится таймер, который запускается одновременно с запуском приложения.



Рисунок 2.2 Интерфейс приложения

При нажатии на изображение оно открывается и остается открытым, после этого нужно открыть еще одно изображение (Рисунок 2.3). При совпадении изображений они убираются, в обратном случае вновь закрываются. (Рисунок 2.4, Рисунок 2.5)



Рисунок 2.3



Рисунок 2.4



Рисунок 2.5

Целью игры является найти все пары изображений за минимальное количество ходов и времени.

По завершении игры появляется диалоговое окно с уведомлением о прохождении игры и количеством ходов и времени, затраченных на прохождение. (Рисунок 2.6)

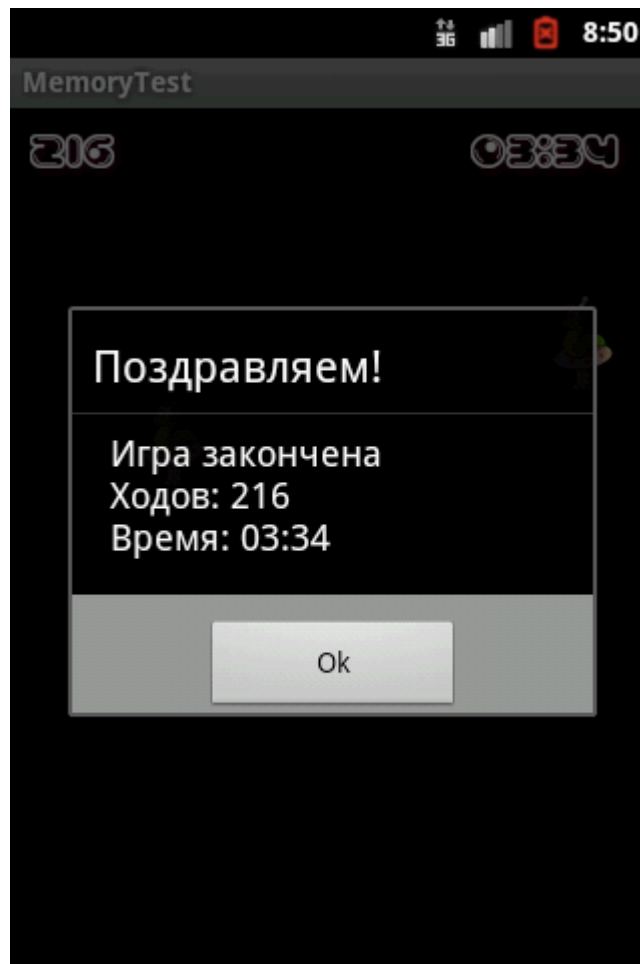


Рисунок 2.6

## ГЛАВА 3

### ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ

#### 3.1 Класс NewGridAdapter

Для отображения картинок на экране будет использоваться инструмент GridView. Однако этот элемент не предназначен для непосредственно хранения элементов. Хранить и передавать данные мы будем с помощью встроенного адаптера (BaseAdapter) в View, составляющего основу для GridView. Функционала базового адаптера будет недостаточно, следовательно, нужно написать другой. Напишем класс NewGridAdapter, который будет наследоваться от встроенного BaseAdapter. Методы, реализуемые в реализованном классе NewGridAdapter:

- void MakePictArray () – помещаем изображения в вектор и располагаем их в рандомном порядке
- View getViewByPos(int positionNumber, View convertView, ViewGroup parent) – возвращает изображение из ячейки с номером position\_number, при этом учитывая его текущее состояние (CELL\_OPENED, CELL\_CLOSED, CELL\_DELETED)
- void checkCellsOpened() – проверка на наличие открытых ячеек. Если открыты два изображения, то делаем их сравнение и в случае совпадения удаляем и закрываем если они не совпадают.
- void cellOpen (int positionNumber) – присваивает изображению с указанным номером статус CELL\_OPENED
- int getElementNum() – возвращает число изображений в GridView
- Object getObj(int positionNumber) – возвращается объект с номером positionNumber
- long getElementId(int positionNumber) – возвращается идентификатор объекта с номером positionNumber
- boolean checkGameOver() – проверка условия завершения игры (нету закрытых картинок)

## 3.2 Класс MemoryActivity

В классе MemoryActivity мы создаем объект GridView и связываем его с адаптером, реализованным в классе GridAdapter. Кроме того, мы создаем счетчик ходов и сразу присваиваем ему нулевое значение, а также запускаем таймер.

Далее пишем обработчик события setOnItemClickListener(), который будет вызываться при нажатии на элемент. При его вызове должно происходить следующее:

- проверка наличия открытых ячеек. Если открыты два изображения, то делаем их сравнение и в случае совпадения удаляем и закрываем если они не совпадают.
- открывается выбранное изображение
- если открытых изображений больше нет, то игра завершается

Для этого мы используем методы checkCellsOpened(), cellOpen() и checkGameOver(), реализованные в GridAdapter.

Стоит также отметить, что инкрементация счетчика ходов происходит при каждом открытии ячейки.

Для уведомления об окончании игры используется метод ShowGameOver(). В нем появляется диалоговое окно, в котором пишется уведомление о завершении игры, количество ходов и количество времени, потраченное на прохождение. Также в диалоговом окне есть кнопка завершения, при нажатии на которую текущая Activity закрывается.



### 3.3 Класс MemoryStart

В этом классе мы работаем с элементами, которые видим в главном меню при запуске игры.

Сперва мы создаем два объекта типа Button, затем пишем для каждого из них свой обработчик. При нажатии на кнопку “Старт” вызывается метод StartGame(), в котором запускается Activity с игрой. При нажатии на кнопку “Выход” мы вызываем метод finish(), который закрывает текущую Activity.

### 3.4 GUI

Графический интерфейс приложения реализуется в двух документах: start.xml и main.xml.

В start.xml создается интерфейс главного меню приложения. Сначала мы задаем фон, изображение для которого берется из папки res/drawable. Далее создаем стиль для элемента Button. Мы определяем положение элемента на дисплее, а также его размеры. Для придания элементу необходимой формы и цвета подключаем файл style.xml из папки res/drawable, в котором это все реализовано. Также через атрибут style подключаем стиль для текста, который будет расположен на кнопке. Для кнопок есть отдельный стиль при нажатом и ненажатом состоянии.

В main.xml сперва описываем элемент GridView. Позиционируем его на дисплее, задаем размеры, а также указываем число столбцов.

Далее описываем графическое представление элементов TextView и Chronometer. В TextView у нас будет выводиться счетчик ходов. Для обоих элементов мы указываем их позицию на дисплее, размеры, а также задаем стиль MyText, описанный в файле style.xml. Добавление шрифта текста, используемого в этих элементах, реализуется программно в классе MemoriaActivity подключением шрифта, расположенного в папке assets.

## ЗАКЛЮЧЕНИЕ

В ходе данной дипломной работы была изучена специфика разработки мобильных приложений на языке Java. Были сравнены различные мобильные платформы, основное внимание уделялось платформам, для разработки под которые используется язык Java.

Было разработано приложение для улучшения и укрепления памяти, в качестве платформы была выбрана Android.

При разработке использовалась среда Eclipse с установленным плагином Android Developer Tools и комплектом Android SDK. Приложение тестировалось на эмуляторе, поддерживающем самые последние версии мобильных устройств.

Приложение предназначено для широкой аудитории, следовательно, может быть монетизировано. Также предполагается расширение функционала приложения с целью улучшения его эффективности.

Ввиду востребованности и роста популярности мобильных приложений, в частности приложений подобной направленности, можно сделать вывод, что проделанная работа является актуальной.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- [1] Разработка мобильных приложений [Электронный ресурс] – Электронные данные. – Режим доступа: [http://en.wikipedia.org/wiki/Mobile\\_application\\_development](http://en.wikipedia.org/wiki/Mobile_application_development)
- [2] Мобильная платформа Android [Электронный ресурс] – Электронные данные. – Режим доступа: <http://androidbest.jimdo.com/>
- [3] Сравнение платформ [Электронный ресурс] – Электронные данные. – Режим доступа: <http://digitalid.ru/>
- [4] Windows Phone SDK [Электронный ресурс] – Электронные данные. – Режим доступа: [http://fakty-o.ru/windows\\_phone\\_sdk](http://fakty-o.ru/windows_phone_sdk)
- [5] J2ME [Электронный ресурс] – Электронные данные. – Режим доступа: <http://www.pilowar.ru/>
- [6] CodenameOne [Электронный ресурс] – Электронные данные. – Режим доступа: <http://apps4all.ru/>
- [7] Java [Электронный ресурс] – Электронные данные. – Режим доступа: <http://ru.wikipedia.org/wiki/Java>