

С другой стороны, то, что является плюсом, в некоторых ситуациях может являться и минусом. В связи с тем, что библиотека метаязыка применяет трансформации построчно, на большом количестве данных время их выполнения может значительно увеличиться. Пример приведен на рисунке 3.



Рисунок 3 – Время, затраченное на применение трансформаций для выборки объемом в 1000000 строк

Спроектированный метаязык способен решить несколько проблем в ставшем уже классическом подходе Apache Spark, а именно вынести операции над данными «наружу» из кода, что позволяет им быть более прозрачными для тех, кто отвечает за корректность данных трансформаций, а также за анализ их результата. Используя метаязык, можно повысить продуктивность работы за счет уменьшения времени ознакомления с синтаксисом, а также уменьшения времени на поиск ошибок. Однако, как и зачастую, универсального решения не существует, и библиотека метаязыка не избавлена от определенных минусов, но здесь уже все зависит от множества индивидуальных параметров и особенностей конкретных проектов.

Итак, модули разработанного метаязыка связаны друг с другом, формируя каркас библиотеки, определяя ее функционал. Применение разработанного метаязыка увеличивает читаемость запроса, уменьшает время на ознакомление с трансформациями и время на поиск ошибок.

### Литература

1 Explicit Programming Strategies [Electronic resource] / T. D. Latoza [et al.]. – 2019. – Mode of access : <https://arxiv.org/pdf/1911.00046.pdf>. – Date of access : 24.12.2019.

УДК 004.7

*В. В. Василевский*

### ОСОБЕННОСТИ РАЗРАБОТКИ ИГРОВЫХ ПРИЛОЖЕНИЙ НА ПЛАТФОРМЕ ANDROID

*В статье рассматриваются особенности разработки игровых приложений на платформе Android, обоснован выбор платформы Android для разработки игровых приложений. Дается описание программных средств разработки приложений, приводится их сравнительный анализ, указываются основные этапы создания Android-приложений. Приводится описание примера игрового приложения «Кто хочет стать миллионером», реализованного на языке Kotlin.*

**Введение.** Одной из самых популярных операционных систем, на которой работает большинство устройств, является Android [1]. Благодаря открытости и доступности платформы Android рост её популярности непрерывно увеличивается как среди разработчиков приложений, так и среди пользователей. Каждый месяц загружается около миллиона приложений на Android в Google Play, большую часть из которых составляют игровые приложения. Разработка игр сравнивается по популярности с живописью, музыкой, киноиндустрией и литературой.

Разработка приложений на Android является прибыльным и востребованным занятием. Сегодня почти у каждого есть смартфоны. А такие сервисы, как App Store и Google Play, позволяют пользователям скачивать и покупать различные игры и приложения, а разработчикам выставлять их для покупки и скачивания.

В статье описываются особенности разработки игровых приложений на платформе Android, рассказывается о создании игрового приложения «Кто хочет стать миллионером», реализованного на языке Kotlin [2].

**Средства реализации игровых Android-приложений.** Разработка приложений на платформе Android имеет ряд особенностей. Основной средой разработки является Android Studio – интегрированная среда разработки (IDE) для работы с платформой Android, анонсированная 16 мая 2013 года на конференции Google I/O. Она очень удобна, потому что основные пакеты и компоненты среда скачивает и создает сама.

Приложение в Android состоит из 4 основных компонентов: Activity, Service, широковещательный приемник, поставщик содержимого. Каждый компонент – это точка входа, через которую система или пользователь может получить доступ [1].

Activity – это визуальный компонент приложения, любая страница, которую мы видим, открывая приложение. Одна активность может взаимодействовать с другими и наоборот, проще говоря, это некое подобие веб-страницы.

Services – это то, что выполняется, пока приложение находится в фоновом режиме. Этот компонент может выполнять длительные операции или работу для удалённых процессов без визуального интерфейса.

Широковещательный приемник – это компонент Android, позволяющий приложению реагировать на сообщения, которые рассылаются операционной системой Android или приложением.

Поставщик содержимого управляет данными приложения из файлов, баз данных, интернета или других хранилищ.

Игровые приложения должны быть написаны с учетом следующих ограничений, характерных для мобильных устройств, а именно: на вычислительную мощность; на объем оперативной памяти; на объем дисковой памяти. Кроме этого необходимо учитывать, что приложение ориентировано на небольшой экран с невысоким разрешением.

Создание игр для мобильных устройств предполагает детальное описание следующих элементов игры: объектных моделей, функциональных спецификаций, игрового контента и интерфейса.

При разработке мобильных приложений используются различные программные средства. Одним из них является язык Kotlin [2]. Kotlin – язык программирования, работающий поверх JVM (Java Virtual Machine), также компилирующийся в JavaScript через инфраструктуру LLVM. Был разработан компанией JetBrains. Назван он в честь острова Kotlin, расположенного в Финском заливе. Наряду с Java является одним из самых популярных языков для разработки приложений под Android.

**Разработка Android-приложения «Кто хочет стать миллионером» на языке KOTLIN.** Схема работы приложения представлена на рисунке 1. Основные функции приложения: развлекательная; познавательная; обучающая.

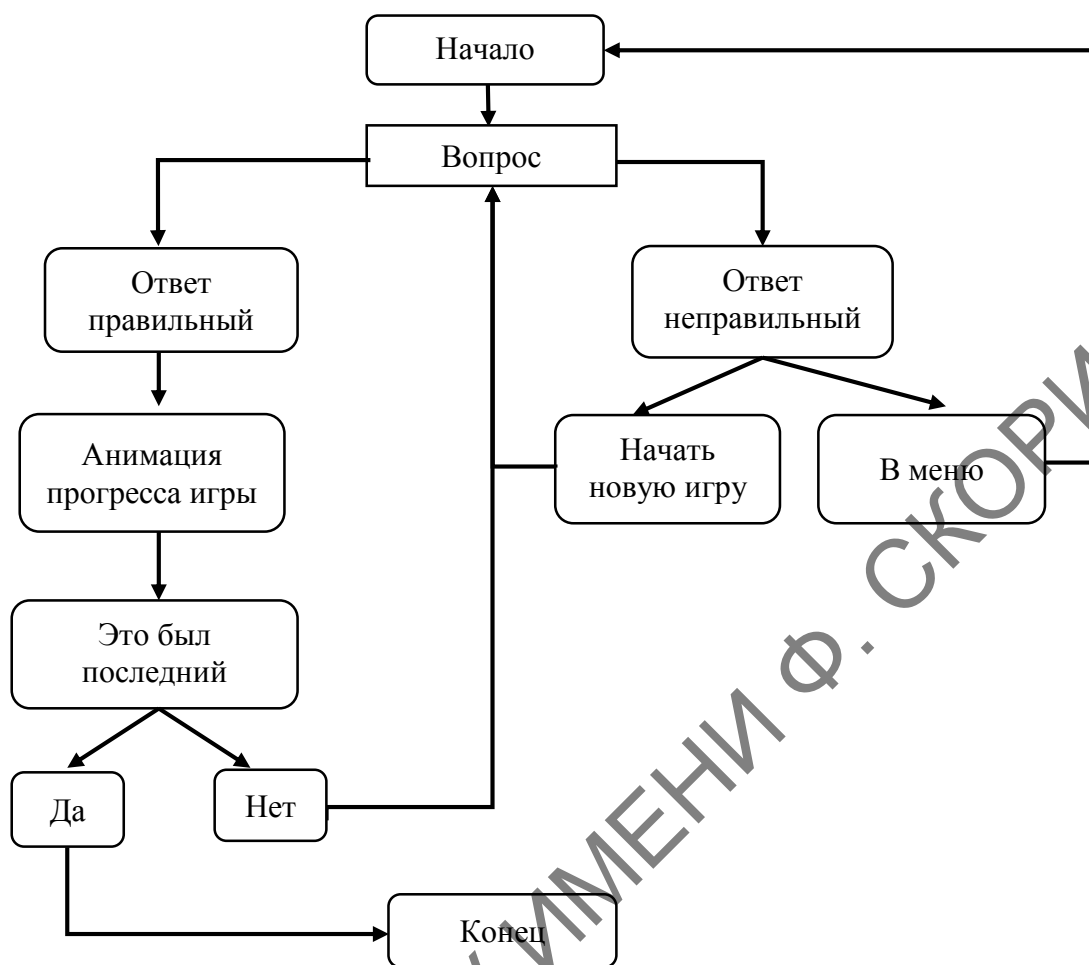


Рисунок 1 – Схема работы приложения

При запуске приложения пользователь попадает на главную форму, представленную на рисунке 2, при нажатии на кнопку “Играть” запускается сама игра (рисунок 3).



Рисунок 2 – Главная форма приложения

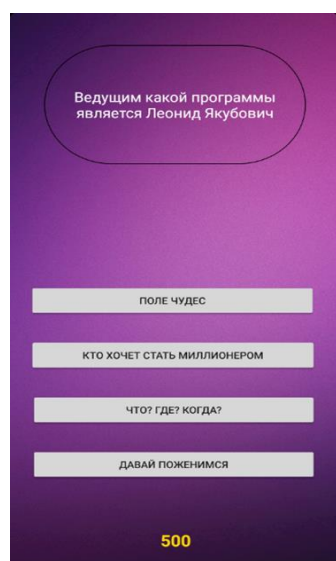


Рисунок 3 – Форма с вопросом и вариантами ответа

Всего в игре 10 уровней сложности, которые увеличиваются с каждым последующим правильным ответом. На каждый уровень есть 5 вопросов, выбор вопроса осуществляется случайным образом. Если пользователь отвечает правильно, запускается анимация выделения правильного ответа, такая, как показана на рисунке 4, если нет – неправильного (выбранного пользователем), такая, как показана на рисунке 5.

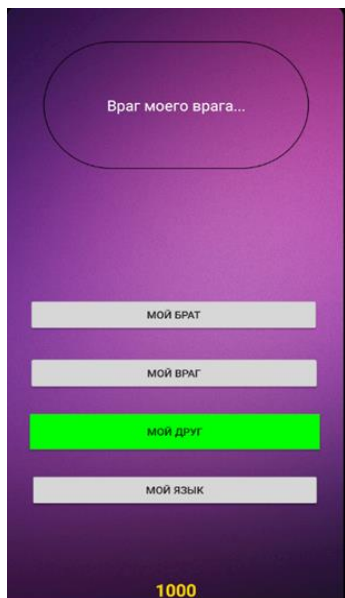


Рисунок 4 – Анимация правильного ответа



Рисунок 5 – Анимация неправильного ответа

Далее, если ответ был верен, запускается анимация, показывающая процесс игры. Для продолжения игры достаточно нажать в любую точку экрана, как показано на рисунке 6. Если ответ был неправильным – запускается окно, предлагающее начать игру заново или выйти в меню, рисунок 7.



Рисунок 6 – Анимация процесса игры

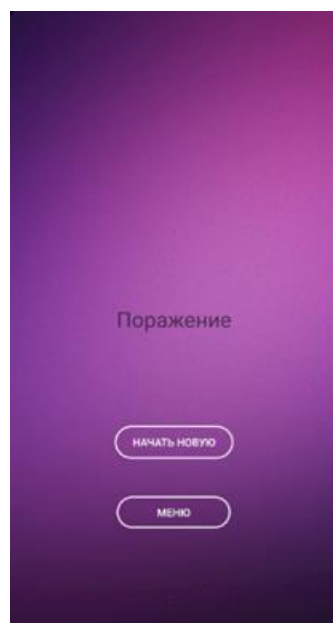


Рисунок 7 – Экран поражения

**Заключение.** В статье представлено игровое приложение «Кто хочет стать миллионером», разработанное для операционной системы Android. Написанное приложение имеет удобный пользовательский интерфейс, легко и быстро в освоении. При реализации была учтена ограниченность ресурсов мобильных устройств и особенности разработки игровых мобильных приложений.

### Литература

- 1 Дейтел, П. Android для разработчиков / П. Дейтел, Х. Дейтел, Э. Дейтел. – 2-е изд. – Санкт-Петербург : Питер, 2015. – 384 с.
- 2 Скин, Дж. Kotlin. Программирование для профессионалов / Дж. Скин, Д. Гринхол. – Санкт-Петербург : Питер, 2020. – 464 с.

УДК 333.71

*А. А. Волкова*

### ИССЛЕДОВАНИЕ ДИНАМИКИ ВАЛЮТНЫХ ПАР

*Проводился анализ валютных пар: российский рубль по отношению к доллару США (RUB/USD), швейцарский франк по отношению к доллару США (CHF/USD), фунт стерлингов по отношению к доллару США (GBP/USD), евро по отношению к доллару США (EUR/USD) и японская иена по отношению к доллару США (JPY/USD). Исследована структура временных рядов, стационарность. Исследована динамика ежеквартальных корреляций. Построены динамические модели курсов валют в классе моделей ARIMA (p, d, q).*

Международная торговля осуществляется в денежных единицах. Чтобы купить товары, услуги или финансовые активы за рубежом, нужно купить или обменять валюту своей страны на нужную валюту на валютном рынке. Основные массы валюты для международной торговли обмениваются в крупнейших финансовых центрах мира. Поэтому определение динамики роста и поведения обменных курсов, выяснение взаимосвязи между валютами является актуальной задачей при исследовании поведения курсов валют RUB (российский рубль), USD (доллар США), CHF (швейцарский франк), GBP (фунт стерлингов), EUR (евро), JPY (японская иена). Информационно-статистической базой изучения динамики валютных пар послужили ежедневные данные закрытия курсов валют на Московской бирже [1] за период с 03.01.2005 по 24.04.2020.

Временные ряды проверены на стационарность с помощью теста Дикки-Фуллера на наличие единичных корней. В этом тесте предполагается, что валютная пара, например, российский рубль к доллару (RUB/USD) является процессом случайного блуждания. Все ряды оказались нестационарными, поэтому для дальнейших исследований исходные данные были преобразованы в логарифмические темпы роста по формуле  $\ln(k_t / k_{t-1})$ , что позволило избавиться от нестационарности (подтвердилось тестом Дикки-Фуллера и анализом автокорреляционных функций (АКФ) и частных автокорреляционных функций (ЧАКФ)). Здесь  $k_t$  – валютная пара в момент времени  $t$ .

Для оценки взаимосвязи курсов  $X$ ,  $Y$  использовался коэффициент корреляции Пирсона, позволяющий измерить степень тесноты статистической связи:

$$r(Y, X) = \frac{\sum_{t=1}^T (x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum_{t=1}^T (x_t - \bar{x})^2 \sum_{t=1}^T (y_t - \bar{y})^2}}, \quad (1)$$