



ИССЛЕДОВАНИЕ КАЧЕСТВА СБОРКИ ПРИЛОЖЕНИЙ ПОД ОПЕРАЦИОННУЮ СИСТЕМУ ANDROID РАЗЛИЧНЫМИ КОМПИЛЯТОРАМИ

НАРОЖНЫЙ В.В., НАРОЖНАЯ Е.В.

Оценивается качество сборки приложения тремя компиляторами: Android Development Tools, Android Studio 2 и Visual Studio 2015. В качестве критериев выбираются быстрдействие и размер полученных приложений. Для чистоты эксперимента описываются три приложения в каждой среде. Приложения реализуют простейший алгоритм поиска простых чисел.

Введение

Android – операционная система (OS Android) для смартфонов, интернет-планшетов, электронных книг, цифровых проигрывателей, наручных часов, игровых приставок, нетбуков, смартбуков, очков Google, телевизоров и других устройств [1].

Классическим средством разработки приложений под OS Android является язык программирования Java [2], принадлежащий компании Oracle.

Рекомендованная Google программная среда разработки приложений на OS Android до декабря 2014 года была Eclipse [3] Android Development Tools (ADT). На сегодняшний день компания Google активно продвигает новую среду Android Studio 2 (AS2) [4], которая вышла на смену Android Studio с 07.04.2016.

В последнее время появились сообщения о покупке компанией Microsoft компании Xamarin [5]. Xamarin – американская компания в области разработки ПО. Занимается разработкой и поддержкой Mono и инструментов для разработки приложений на языке C# для iOS, Android, Windows, Mac [6].

Компания Xamarin уже несколько лет создавала инструменты разработки под Android (в отличие от рекомендуемого разработчиками Android языка программирования Java, использовался язык программирования C# [7]), которые можно было самостоятельно дополнительно встроить в более ранние версии Visual Studio. В Visual Studio 2015 (VS15) Xamarin включен как один из основных инструментов, что представляет обоснованный интерес в свете покупки компании. Можно предположить, что Microsoft готов свернуть попытки занять рынок мобильных операционных систем для смартфонов и планшетов своей операционной

системой Windows Phone, которая явно не в состоянии отвоевать позиции у Android и iOS. Полное описание исследования приведено на электронном ресурсе [8].

1. Цель исследования

Одной из наиболее сложных задач является оценка качества работы вычислительных устройств. Сложно оценить, какое вычислительное устройство лучше. Классически необходимо выбрать несколько критериев и создать условия, при которых другими критериями можно пренебречь. Тогда проведение исследований устройств по этим критериям можно считать адекватным. Только на основании такого исследования можно делать выводы о качестве. Наличие различных компьютерных архитектур, сред программирования, операционных систем заметно усложняет такую оценку.

Цель и задачи исследования. Цель данного исследования – оценить качество сборки приложения тремя различными компиляторами: ADT, AS2 и VS15.

Для достижения поставленной цели необходимо решить следующие задачи:

- выбрать критерии оценки качества (быстрдействие и размер полученных приложений);
- для чистоты эксперимента написать три приложения на ADT, AS2 и VS15, где реализован простейший алгоритм поиска простых чисел (рис.1);

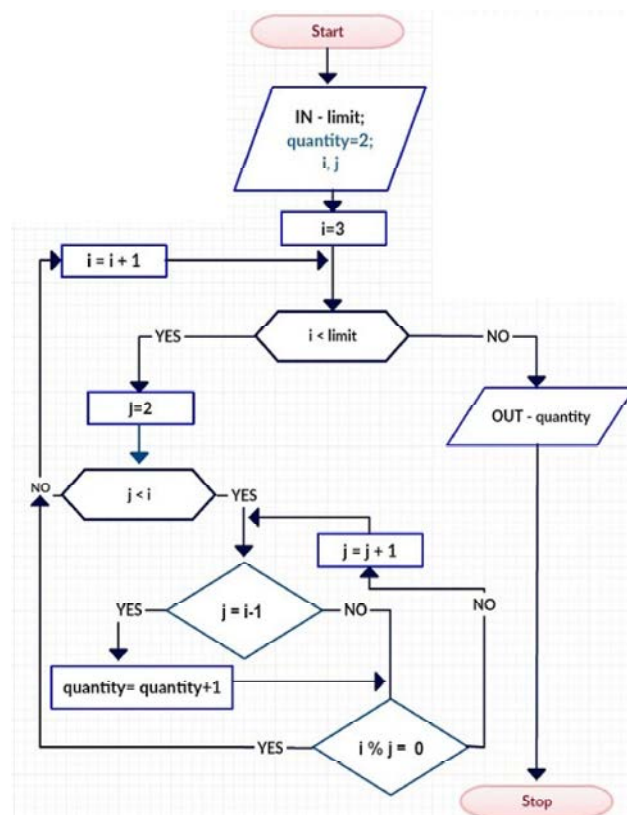


Рис. 1. Алгоритм поиска простых чисел

– в качестве вывода данных использованы время начала работы, время окончания работы и количество (quantity) найденных простых чисел для контроля правильности работы каждого приложения;

– для минимизации возможного воздействия абстракции ООП [9] максимально использовать базовые команды языков Java и C#, а также минимизировано использовать средства ООП;

– для чистоты эксперимента не использованы параллельные потоки, что подвешивает интерфейс на все время поиска.

Таким образом, при поиске относительно большого количества простых чисел (в данных примерах в диапазоне от 3 до limit) можно считать оценку качества сборки приложений адекватной поставленной.

2. Коды программ и интерфейсы пользователей

С учетом всего сказанного были созданы три приложения на ADT, AS2 и VS15. Во всех трех приложениях реализованы одинаковые интерфейсы с использованием файлов данных XML (рис.2) и стилей по умолчанию, характерных для каждой среды.

```

1 <LinearLayout xmlns:android="http://schemas.android
2   android:layout_width="fill_parent"
3   android:layout_height="fill_parent"
4   android:orientation="vertical" >
5   <TextView
6     android:id="@+id/textView0"
7     android:layout_width="match_parent"
8     android:layout_height="wrap_content"
9     android:text="Search prime numbers to " />
10  <EditText
11    android:id="@+id/editText1"
12    android:layout_width="match_parent"
13    android:layout_height="wrap_content"
14    android:text="Input number" />
15  <TextView
16    android:id="@+id/textView1"
17    android:layout_width="match_parent"
18    android:layout_height="wrap_content"
19    android:text="Start time" />
20  <Button
21    android:id="@+id/button1"
22    android:layout_width="match_parent"
23    android:layout_height="wrap_content"
24    android:text="Start" />
25  <TextView
26    android:id="@+id/textView2"
27    android:layout_width="match_parent"
28    android:layout_height="wrap_content"
29    android:text="Stop time" />
30  <TextView
31    android:id="@+id/textView3"
32    android:layout_width="match_parent"
33    android:layout_height="wrap_content"
34    android:text="Quantity prime number" />
35 </LinearLayout>

```

Рис. 2. XML-файл данных ADT

После компиляции все установочные файлы имеют расширение apk (рис.3). Находятся на компьютере в соответствующих папках. Файлы необходимо перенести на устройство для проведения исследований.

Имя	Дата изменения	Тип
findprimeas2-debug.apk	25.04.2016 12:46	Файл "АРК"

Рис. 3. Инсталляционный файл APK

Приложения были написаны с учетом особенностей каждой среды и языков Java (рис.4) и C# (рис.5).

```

package com.example.findprimenumberseclipse;
import ...
public class MainActivity extends Activity
    implements OnClickListener{
    TextView textView1,textView2,textView3;
    Button button;
    EditText editText;
    Calendar c;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button=(Button)findViewById(R.id.button1);
        button.setOnClickListener(this);
        textView1=(TextView)findViewById(R.id.textView1);
        textView2=(TextView)findViewById(R.id.textView2);
        textView3=(TextView)findViewById(R.id.textView3);
        editText=(EditText)findViewById(R.id.editText1);
    }
    @Override
    public void onClick(View v) {
        int searchToTheNumber=Integer.parseInt(editText
            .getText().toString()), quantityPrimeNumber=1;
        c = Calendar.getInstance();
        textView1.setText(""+c.get(Calendar.HOUR)+"":"+
            c.get(Calendar.MINUTE)+"."+c.get(Calendar.SECOND));
        for(int i=3;i<searchToTheNumber;i++){
            for(int j=2;j<i;j++){
                if(j==i-1){quantityPrimeNumber++;}
                if(i%j==0){break;}
            }
        }
        c = Calendar.getInstance();
        textView2.setText(""+c.get(Calendar.HOUR)+"."+
            c.get(Calendar.MINUTE)+"."+c.get(Calendar.SECOND));
        textView3.setText("Quantity prime number: "
            +quantityPrimeNumber);
    }
}

```

Рис.4. Программа, написанная на Java

```

using ...
namespace App1
{[Activity(Label = "App1", MainLauncher = true,
    Icon = "@drawable/icon")]
    public class MainActivity : Activity
    { TextView textView1, textView2, textView3;
      Button button;
      EditText editText;
      protected override void OnCreate(Bundle bundle)
      {base.OnCreate(bundle);
        SetContentView(Resource.Layout.Main);
        button = FindViewById<Button>(Resource.Id.button1);
        textView1=FindViewById<TextView>(Resource.Id.textView1);
        textView2 = FindViewById<TextView>(Resource.Id.textView2);
        textView3 = FindViewById<TextView>(Resource.Id.textView3);
        editText = FindViewById<EditText>(Resource.Id.editText1);
        button.Click += delegate
        { int searchToTheNumber = Int32.Parse((
            (EditText)editText).Text), quantityPrimeNumber = 1;
          textView1.Text = string.Format(DateTime.
            Now.ToString("HH:mm:ss tt"));
          for (int i = 3; i < searchToTheNumber; i++)
          { for (int j = 2; j < i; j++)
              { if (j == i - 1) { quantityPrimeNumber++; }
                if (i % j == 0) { break; }
              }
            }
          textView2.Text = string.Format(DateTime.
            Now.ToString("HH:mm:ss tt"));
          textView3.Text = string.Format(
            "{0} Quantity prime number: ", quantityPrimeNumber);
        }
    }
}

```

Рис.5. Программа, написанная на C#

3. Установка и исследование

Все установочные файлы для OS Android имеют расширение APK. Для установки приложений необходимо зайти в настройки устройства и найти пункт «Разрешить установку из неизвестных источников» (на разных устройствах этот пункт может иметь разное местоположение и название, но, как правило, это в разделе «Безопасность»). Установить галочку на разрешение. Затем загрузить установочные файлы и поочередно сделать на них «тач». Приложения установятся на устройство (рис.6). В данном исследовании использовано устройство Galaxy Ace 3 GT-S7272.



Рис.6. Иконки приложений на устройстве

Теперь можно зайти в «Настройки» - «Диспетчер приложений» и посмотреть размеры, которые занимают приложения в памяти устройства ADT (рис.7), AS2 (рис.8), VS15 (рис.9).

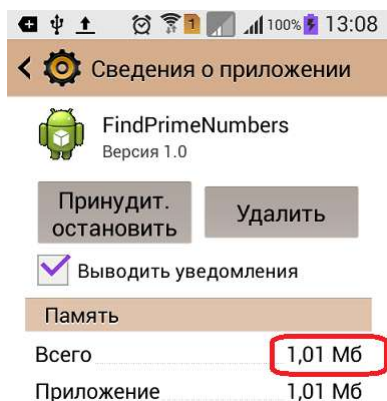


Рис. 7. Размер приложения от компилятора ADT

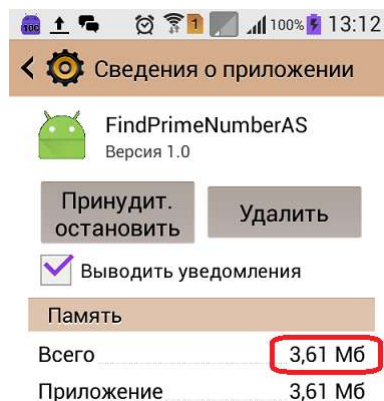


Рис. 8. Размер приложения от компилятора AS2

По требованию издательства цвета, на приведенных скринах, были обращены. Как можно видеть, приложение после установки занимает 1,01 Мб.

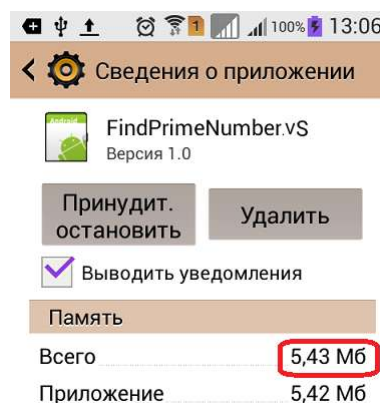


Рис. 9. Размер приложения от компилятора VS15

Итоговый результат можно увидеть на диаграмме (рис.10).

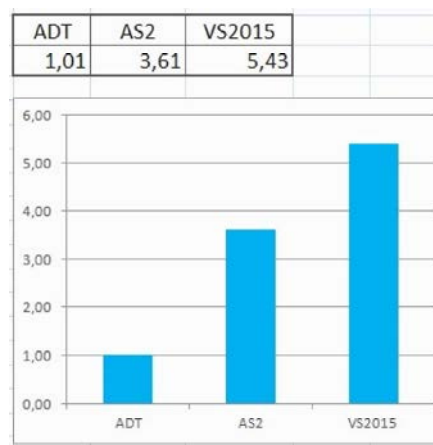


Рис.10. Диаграмма размера установочных файлов установленных приложений

Проверить качество работы компилятора по критерию занимаемого объема внутренней памяти достаточно быстро. Задача по проверке скорости работы приложений занимает значительно больше времени. Для чистоты эксперимента каждый запуск проводился после предварительной очистки памяти. Процесс работы приложений занимал несколько минут (в зависимости от вводимого значения limit). Итоговые результаты приведены на рис. 11.

Выводы

Научная новизна. Предложен метод оценки качества сборки приложения тремя компиляторами: Android Development Tools, Android Studio 2 и Visual Studio 2015. Обоснован выбор критериев (быстродействие и размер полученных приложений) для чистоты эксперимента.

Практическая ценность. Несмотря на огромный опыт компании Microsoft по созданию средств программирования, следует отметить, что по обоим критериям средства разработки приложений на Java лучше. Оба параметра в пользу средств ADT и AS2. В свою

очередь, ADT значительно опережает по размерам установленного приложения. Возможно, AS2 и VS15 можно оптимизировать по этому параметру, но большинство разработчиков этим не занимается, а значит, создаются приложения с явно завышенными объемами включенных ресурсов в конечный продукт. Возможно, это не так заметно для больших приложений, но тут результат очевиден. Также не добавляют оптимизма процесс установки VS15 или AS2 (явно сложнее процесса установки ADT) и значительно более высокие требования к ресурсам компьютера.

limit	ADT,сек	AS2,сек	VS15,сек
60000	9	9	10
80000	16	16	18
100000	25	25	27
120000	34	34	37
140000	47	47	51
160000	61	61	66
180000	77	77	83
200000	94	94	102
220000	112	112	122

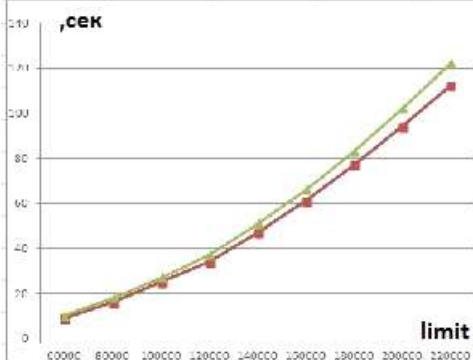


Рис. 11. Показатели скорости работы приложений

Учитывая все сказанное выше, можно сделать вывод, что явно прослеживается очередной виток от качества сборки приложения к увеличению числа различных фишек в средах разработки.

Литература: 1.*Android*: электронный ресурс. Точка доступа https://ru.wikipedia.org/wiki/Android_2. 2.*Java*: электронный ресурс. Точка доступа https://uk.wikipedia.org/wiki/Java_3. 3.*Eclipse*: электронный ресурс. Точка доступа https://uk.wikipedia.org/wiki/Eclipse_4. 4.*Android Studio*: электронный ресурс. Точка доступа https://ru.wikipedia.org/wiki/Android_Studio. 5.*Microsoft to acquire Xamarin and empower more developers to build apps on any device*: электронный ресурс. Точка доступа <http://blogs.microsoft.com/blog/2016/02/24/microsoft-to-acquire-xamarin-and-empower-more-developers-to-build-apps-on-any-device/>. 6.*Xamarin*: электронный ресурс. Точка доступа https://ru.wikipedia.org/wiki/Xamarin_7. 7.*C#*: электронный ресурс. Точка доступа https://ru.wikipedia.org/wiki/C_Sharp_8. 8.*Полное описание исследований*: электронный ресурс <http://androidresearchua.blogspot.com/>. 9.*Абстрактный тип данных*: электронный ресурс. Точка доступа https://ru.wikipedia.org/wiki/Абстрактный_тип_данных

Поступила в редколлегию 16.02.2016

Рецензент: д-р техн. наук, проф. Моисеенко В.И.

Нарожный Виталий Васильевич, канд. техн. наук, доцент кафедры Информационных технологий и мехатроники Харьковского национального автомобильно-дорожного университета. Научные интересы: исследование особенностей операционной системы Android. Адрес: Украина, 61000, Харьков, ул. Ярослава Мудрого, 25, тел.:+380508577597.

Нарожная Екатерина Витальевна, студентка кафедры Информационных технологий и мехатроники Харьковского национального автомобильно-дорожного университета. Научные интересы: исследование особенностей операционной системы Android. Адрес: Украина, 61000, Харьков, ул. Ярослава Мудрого, 25, тел.:+380667487704.