

*Сарычева Юлия Юрьевна, студент-магистр, Калужский филиал ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»*

*Белов Юрий Сергеевич, научный руководитель, к.ф.-м.н., доцент, Калужский филиал ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»*

## **ТЕСТИРОВАНИЕ GUI МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПРИ ПОМОЩИ MONKEY**

**Аннотация:** Приложение можно рассматривать как комбинацию многих состояний графического интерфейса и переходов между ними. Каждое состояние GUI выполняет разные функции или отображает различный контент. Пользователи приложения перемещаются между состояниями пользовательского интерфейса, взаимодействуя с элементами пользовательского интерфейса. В частности, взаимодействие может включать щелчок, прокрутку или ввод текста в элемент GUI, например кнопку, изображение или текстовый блок. Работа генератора входных данных заключается в создании последовательности взаимодействий для анализируемого приложения, которые можно использовать для обнаружения проблем, связанных с программным обеспечением, таких как ошибки, уязвимости и проблемы безопасности. Monkey является официальным инструментом для автоматизированного тестирования приложений в Android. Monkey отправляет случайные типы входных событий в случайные места на экране без учета структуры своего графического интерфейса.

**Ключевые слова:** графический интерфейс пользователя, тестирование, мобильное приложение, автоматизированное тестирование.

**Abstract:** An application can be viewed as a combination of many GUI states and transitions between them. Each GUI state performs a different function or displays different content. Application users move between user interface states by interacting with user interface elements. In particular, the interaction may include clicking, scrolling, or entering text on a GUI element, such as a button, image, or text box. The job of the input generator is to generate a sequence of interactions for the application being analyzed that can be used to discover software-related issues such as bugs, vulnerabilities, and security issues. Monkey is the official automated testing tool for Android applications. Monkey sends random types of input events to random places on the screen, regardless of the structure of its GUI.

**Keywords:** graphical user interface, testing, mobile application, automated testing.

**Введение.** В Android генераторы ввода взаимодействуют с приложениями так же, как и люди: отправляют смоделированные жесты в графический интерфейс приложения. Поскольку допустимые жесты в состоянии пользовательского интерфейса ограничены, основное различие между различными генераторами тестов заключается в их стратегиях, используемых для определения приоритетов этих действий.

В основном существует три типа стратегий: целевые, основанные на моделях и случайные.

Случайная стратегия используется в Monkey [6], официальном инструменте для автоматизированного тестирования приложений в Android.

Monkey отправляет случайные типы входных событий в случайные места на экране без учета структуры своего графического интерфейса.

Рассмотрим процесс тестирования приложения Блокнот с помощью инструмента Monkey.

В первую очередь необходимо прописать в терминале путь до тестируемого приложения с помощью команды: `adb install «путь до файла с`

приложением», например, C:\notepad.apk. Результат работы команды представлен на рисунке 1.

```
C:\Users\Arseniy>adb install C:\notepad.apk
[100%] /data/local/tmp/notepad.apk
      pkg: /data/local/tmp/notepad.apk
Success
```

Рис. 1. Путь до тестируемого приложения

Чтобы запустить Monkey, необходимо знать имя пакета, к которому инструмент будет обращаться. Для этого необходимо сначала перейти в режим работы с Shell с помощью команды `adb shell`, а после прописать команду `pm list packages -f`. Она выводит список всех пакетов для всех приложений, установленных на устройстве, чтобы ограничить список пакетов по имени приложения, необходимо выполнить команду `pm list packages -f | grep «name»`, где `name` – это имя приложения, в данном случае `notepad`. Результат работы команды представлен на рисунке 2.

```
root@generic_x86_64:/ # pm list packages -f | grep notepad
package:/data/app/com.example.simplenotepad-1/base.apk=com.example.simplenotepad
```

Рис. 2. Найденный пакет тестируемого приложения

После того, как пакет будет найден, необходимо выйти из режима `shell` с помощью команды `exit`. Далее необходимо запустить Monkey с помощью команды `adb shell monkey -p «package» -v 300`, где «package» - это имя пакета, в данном случае `com.example.simplenotepad`, а 300 – это количество необходимых действий. Результат работы команды представлен на рисунке 3.

```

C:\Users\Arseniy>adb shell monkey -p com.example.simplenotepad -v 300
:Monkey: seed=1478753114664 count=300
:AllowPackage: com.example.simplenotepad
:IncludeCategory: android.intent.category.LAUNCHER
:IncludeCategory: android.intent.category.MONKEY
// Event percentages:
// 0: 15.0%
// 1: 10.0%
// 2: 2.0%
// 3: 15.0%
// 4: -0.0%
// 5: 25.0%
// 6: 15.0%
// 7: 2.0%
// 8: 2.0%
// 9: 1.0%
// 10: 13.0%
:Switch: #Intent;action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;launchFlags=0x10200000;component=com.example.simplenotepad/.MainActivity;end
// Allowing start of Intent ( act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.example.simplenotepad/.MainActivity ) in package com.example.simplenotepad
:Sending Trackball (ACTION_MOVE): 0:(-5.0,4.0)

```

Рис. 3. Запуск Monkey

После этого на эмуляторе открывается тестируемое приложение и в поле ввода вводится случайный текст. Внешний вид процесса тестирования представлен на рисунке 4.

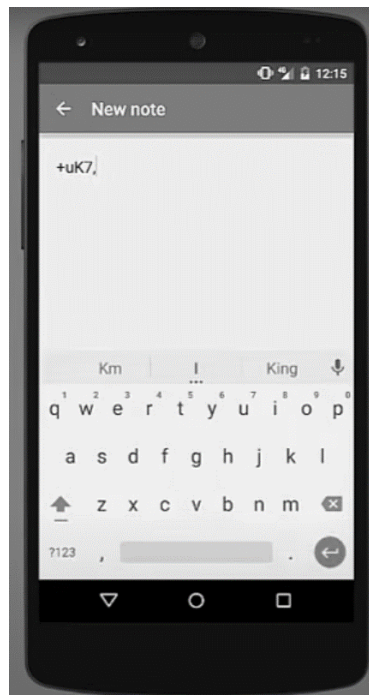


Рис. 4. Процесс тестирования с помощью Monkey

После выполнения 300 действий Monkey завершает свою работу, что представлено на рисунке 5.

```
:Sending Touch (ACTION_DOWN): 0:(1028.0,1390.0)
:Sending Touch (ACTION_UP): 0:(1080.0,1292.7303)
:Sending Touch (ACTION_DOWN): 0:(1030.0,585.0)
:Sending Touch (ACTION_UP): 0:(996.15967,609.3319)
:Sending Flip keyboardOpen=false
Got IOException performing flipjava.io.IOException: write failed: EINVAL (Invalid argument)
// Injection Failed
:Sending Touch (ACTION_DOWN): 0:(425.0,836.0)
:Sending Touch (ACTION_UP): 0:(424.42078,864.00507)
:Sending Trackball (ACTION_MOVE): 0:(0.0,-5.0)
:Sending Touch (ACTION_DOWN): 0:(868.0,874.0)
:Sending Touch (ACTION_UP): 0:(832.7742,823.341)
// Rejecting start of Intent ( act=android.intent.action.MAIN cat=[android.intent.category.HOME] cmp=com.google.android.googlequicksearchbox/com.google.android.launcher.GEL ) in package com.google.android.googlequicksearchbox
:Sending Trackball (ACTION_MOVE): 0:(2.0,2.0)
Events injected: 300
:Sending rotation degree=0, persist=false
:Dropped: keys=0 pointers=0 trackballs=0 flips=1 rotations=0
## Network stats: elapsed time=9696ms (0ms mobile, 0ms wifi, 9696ms not connected)
// Monkey finished
```

Рис. 5. Завершенный процесс тестирования с помощью Monkey

Однако в процессе тестирования использовались системные кнопки, чтобы их исключить необходимо для запуска тестирования ввести команду `adb shell monkey -pct-syskeys 0 -p «package» -v 300`, где после ключа `-pct-syskeys` указывается в процентном соотношении частота возникновения системных действий, в данном случае это 0, «package» - это имя пакета, в данном случае `com.example.simplenotepad`, а 300 – это количество необходимых действий. Если в команду выполнения после ключа `-pct-syskeys` добавить ключ `-throttle` и значение задержки после каждого действия в миллисекундах, например, 100, то процесс тестирования будет специально замедлен. Также в команду можно добавит ключ `-s` с числовым значением, который означает сколько раз будет проведен процесс тестирования. Результат тестирования приложения представлен на рисунке 6.

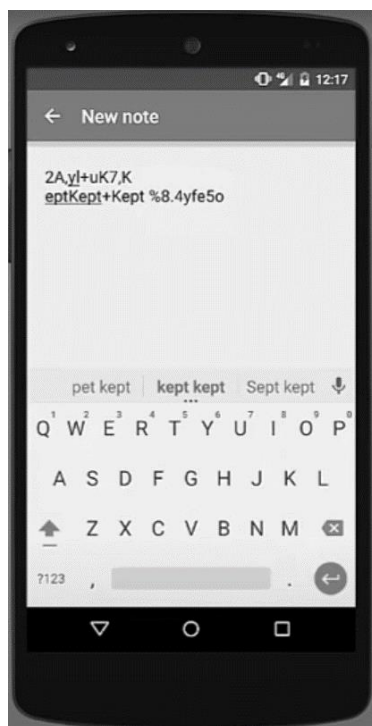


Рис. 6. Результат тестирования с помощью Monkey

В случае, если приложение вылетит в процессе тестирования, ключ `-kill-process-after-error` остановит выполнение тестирования, а не продолжит его.

**Заключение.** Обеспечение качества — это путь, которым тщательно занимаются команда QA и инженеры-тестировщики. На этом пути возникает множество проблем, особенно когда есть ручное тестирование, которое QA должно поддерживать на протяжении всего жизненного цикла тестирования.

По мере роста спроса на рынке программного обеспечения организациям необходимо решать свои задачи и опережать конкурентов. Автоматизированное тестирование может сократить эти усилия, не теряя качества. Автоматизированная генерация ввода GUI стала активной областью исследований с момента распространения мобильных приложений. Примером генератора может служить Monkey, официальный инструмент для автоматизированного тестирования приложений в Android.

#### **Библиографический список:**

1. Naja Fa., Mansur Sy., Wibawanto Ad. Automated Software Testing on Mobile Applications: A Review with Special Focus on Android Platform // 20th

International Conference on Advances in ICT for Emerging Regions. 2020. pp. 4-6.

2. Василенко, Р. И. Автоматизированное тестирование мобильных приложений / Р. И. Василенко, С. А. Белоусова // Инновационные технологии и дидактика в обучении: сборник статей III Международной научно-практической конференции, Краснодар, 29–30 июня 2015 года / Борисова Е.А. – Краснодар: Издательство Южного Федерального Университета, 2015. – С. 31-34. – EDN UTWWDN.

3. Михалевская, К. А. Сравнение инструментов для автоматизации тестирования мобильных приложений на ОС Android / К. А. Михалевская, М. А. Сергачева, И. Н. Мерзляков // КОГРАФ - 2020: сборник материалов 30-й Всероссийской научно-практической конференции по графическим информационным технологиям и системам, Нижний Новгород, 13–16 апреля 2020 года. – Нижний Новгород: Нижегородский государственный технический университет им. Р.Е. Алексеева, 2020. – С. 250-255. – DOI 10.46960/43791586\_2020\_250. – EDN XZGGPU.

4. Ананьев, В. Ю. Автоматизированное тестирование мобильных приложений при помощи nunit и Appium / В. Ю. Ананьев // Инновационные технологии, экономика и менеджмент в промышленности: Сборник научных статей X международной научной конференции, Волгоград, 21–22 октября 2021 года. – Волгоград: Общество с ограниченной ответственностью "КОНВЕРТ", 2021. – С. 136-138. – EDN BBDGXU.

5. Pan M., Xu To., Pei Yu. GUI-Guided Test Script Repair for Mobile Apps // IEEE Transactions on Software Engineering. 2022. Vol. 48. №3. pp. 3-5.