

## Лабораторная работа №12

### Разработка apk, iOS, exe-приложений

Цель работы: сформировать умения разрабатывать apk, iOS, exe-приложений с помощью фреймворка Kivy.

Задание:

1. Изучить теоретические сведения.
2. Выполнить задания в соответствии с вариантом.

#### Теоретические сведения

##### ***Создание APK-пакетов для мобильных приложений под Android с помощью утилиты Buildozer***

Buildozer — это инструмент, который автоматизирует весь процесс сборки, загружает и настраивает все необходимые компоненты для этого, включая Android SDK и NDK, а затем создает apk, который можно автоматически отправить на устройство.

Ознакомиться с официальной информацией по работе с Buildozer можно:

<https://kivy.org/doc/stable/guide/packaging-android.html#packaging-android>

В операционной системе Windows утилита Buildozer не работает. Для сборки APK пакета требуется любая Linux система, либо ее виртуальный образ на компьютере под Windows.

1. В VirtualBox установить ISO-образ диска нужного дистрибутива Linux (Ubuntu, Mint, Lite или др.)
2. На виртуальную машину нужно установить PyCharm CE (Community Edition)
3. Через терминал Linux выполнить команды (потребуется ввод пароля суперпользователя)  
`sudo apt -y install python3-distutils  
sudo apt -y install python3-pip  
pip3 install setuptools`

4. Через терминал PyCharm установить фреймворк Kivy (библиотеку KivyMD):  
`pip install kivy`

5. Проверить работу файла .py в PyCharm (можно создать новый файл main.py, скопировать туда код из файла .py, созданного в Windows, запустить)

6. В терминале Linux перейти в папку проекта (где размещается ваш файл main.py), установить модуль buildozer, выполнив поочередно команды:

```
sudo apt install -y git  
git clone https://github.com/kivy/buildozer.git  
cd buildozer  
sudo python3 setup.py install
```

7. В терминале Linux вернуться в папку с приложением (`cd ..`)

8. Создать файл buildozer.spec, который содержит настройки для компиляции проекта (название приложения, версия, иконка и т.д.). Для этого выполнить команду  
`buildozer init`

9. Отредактировать файл buildozer.spec, выполнив команду  
`nano buildozer.spec`

Задать заголовок приложения, имя пакета, в requirements – указываются все необходимые библиотеки, в orientation – ориентация приложения.

Сохранить файл `CTRL + x`

10. В терминале Linux установить зависимости buildozer, выполнив команды  
`sudo apt update`

# из официальной документации buildozer:

```
# https://buildozer.readthedocs.io/en/latest/installation.html
```

```
sudo apt install -y git zip unzip openjdk-17-jdk python3-pip autoconf libtool  
pkg-config zlib1g-dev libncurses5-dev libncursesw5-dev libtinfo5 cmake  
libffi-dev libssl-dev
```

11. В терминале Linux установить модуль Cython (он понадобится для выполнения компиляции APK-пакета):

```
pip3 install --user --upgrade Cython==0.29.33 virtualenv  
# the --user should be removed if you do this in a venv  
  
# add the following line at the end of your ~/.bashrc file  
export PATH=$PATH:~/local/bin/
```

12. Запустить процесс компиляции (может занять некоторое время), Buildozer автоматически скачает все необходимые зависимости и соберет APK. Для этого в терминале Linux выполнить команду

```
buildozer android debug
```

При этом согласиться с лицензией и условиями (нажать Y)

Можно подключить свое устройство Android и запустить:

```
buildozer android debug deploy run
```

для сборки, отправки и автоматического запуска APK на вашем устройстве.

13. В папке с проектом создается папка bin с apk-файлом, который можно переслать на телефон (например, через почту) и установить.

Вместо виртуальных машин можно подключить подсистему Windows для Linux (Windows Subsystem for Linux, WSL): пуск → WindowsPowerShell (администратор) → wsl -install → перезагрузка.

Подробная инструкция по настройке WSL:

<https://learn.microsoft.com/ru-ru/windows/wsl/setup/environment>

Затем установить дистрибутив Linux: <https://docs.microsoft.com/en-us/windows/wsl/install>.

Можно использовать два подхода при создании среды для формирования APK-пакетов:

1. Создать отдельную папку в том же проекте, где ведется разработка приложения. Этот способ хорош тем, что процессы разработки приложения и сборки инсталляционных пакетов выполняются в рамках одного проекта. Однако при этом в терминальном окне среды разработки придется постоянно переключаться между папкой с основным проектом и папкой с инструментарием сборки инсталляционного пакета.

2. Создать отдельный проект, в котором будет выполняться только сборка инсталляционных пакетов. Перед каждой сборкой придется переносить разработанные модули из проекта разработки в проект сборки APK-пакета.

Выбор того или иного подхода зависит от предпочтений конкретного разработчика.

!!! При создании APK-пакетов приложений, написанных на Kivy, нужно учитывать одну особенность – стартовый файл приложения должен иметь имя main.py. Это связано с особенностями работы сборщика apk-пакетов Buildizer. Этот файл должен располагаться в корневой папке проекта. Остальные элементы проекта можно группировать и располагать в папках проекта с произвольными именами, которые, однако, должны иметь понятный для разработчика смысл. Например, папка Images – для хранения изображений, папка Icon – для хранения иконок, папка KV\_file – для хранения файлов .kv и т. д.

## **Задания**

### **Задание 1**

Напишите Kivy-приложение: по нажатию кнопки меняется цвет экрана на случайный.  
Создать .apk приложение для Android.

### **Задание 2**

Создать оконное приложение, изменив задание из ЛР «Функции и модули».  
Создать .apk приложение для Android.

### **Контрольные вопросы:**

1. Опишите назначение Buildozer.
2. Какой файл конфигурации используется Buildozer? Какие основные параметры в нем нужно задать?
3. Какие операционные системы поддерживает Buildozer для сборки приложений?
4. Какая команда используется для инициализации нового проекта в Buildozer?
5. Какие основные этапы сборки приложения с помощью Buildozer?