

Отладка

Open Source

Полезные ресурсы

Публикуем приложения в Google Play и зарабатываем миллионы

[Открыть страницу приложения в Google Play](#)

В статье присутствуют скриншоты, когда магазин приложений назывался "Android Market". Уже выросло поколение котов, которые никогда не слышали об этом названии. Но общий принцип не изменился. Оставил для истории. Я не могу заново пройти регистрацию, чтобы сделать новые скриншоты.

Итак, за 30 дней вы научились создавать собственные приложения. Пора поделиться ими со всем миром и получить заветный миллион на блюдечке с голубой каёмочкой.

Так как мы волнуемся, то потренируемся на кошках. Возьмем, например, пример из статьи [Пишем справочник](#) и попытаемся выложить его в Google Play. Если все получится, то дальше будет проще.

Прежде чем приступать к дальнейшим операциям, убедитесь, что у вас готовы собственные значки для приложений и название программы. Будет очень странно увидеть в магазине приложений программу со стандартным значком и названием "HelloWorld". А также проверьте все записи в файле манифеста.


Шаг первый. Он трудный самый

Любое приложение, выкладываемое в магазин, должно иметь подписанный сертификат. Сертификат позволяет идентифицировать вас как автора программы. И если кто-то попытается выложить программу с таким же именем как у вас, то ему будет отказано из-за конфликта имён. Под именем приложения имеется в виду полное название пакета.

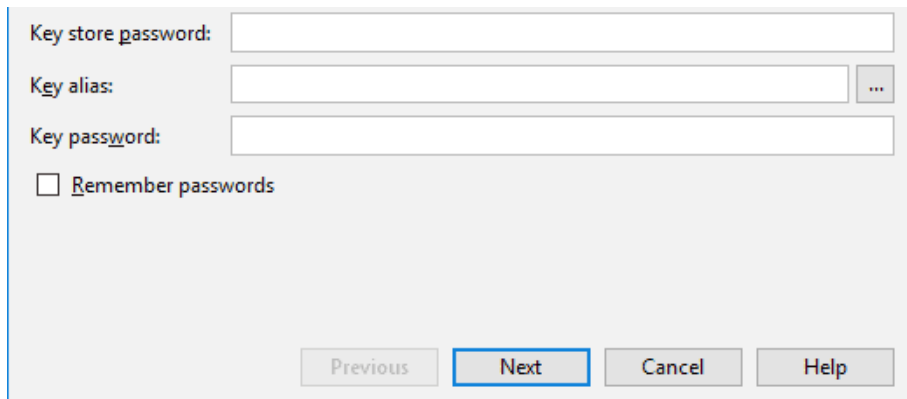
Когда вы запускали свои приложения на эмуляторе или своём телефоне, то среда разработки автоматически подписывала программу отладочным сертификатом. Для распространения через магазин отладочный сертификат не подходит, и вам нужно подписать приложение своим уникальным сертификатом. Это бесплатно, без регистрации и смс.

Создадим подписанный APK-файл, который является что-то типа исполняемым файлом как **notepad.exe** в Windows. Если у вас открыта среда разработки Android Studio, то выберите в меню **Build | Generate Signed APK...**

Появится диалоговое окно мастера, которое необходимо заполнить данными.

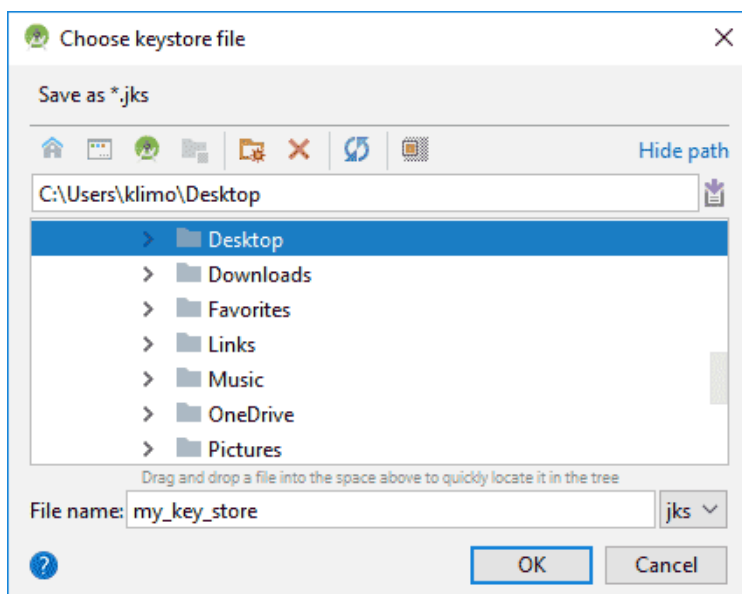
 Generate Signed APK





В первом поле следует указать путь к хранилищу ключей. Если вам раньше приходилось создавать программы раньше, в том числе и в Eclipse, то можете указать уже существующее хранилище через кнопку **Choose existing....** Если вы создаёте хранилище первый раз, то выбирайте кнопку **Create new....** Появится новое диалоговое окно.

В первом поле **Key store path** нужно выбрать папку через кнопку ... и ввести имя для файла с хранилищем, которому будет присвоено расширение **jks**.



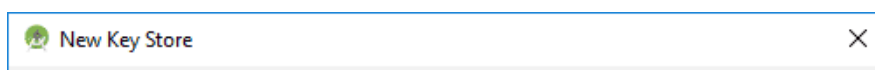
Далее вы вернётесь обратно и продолжаете заполнять поля. Поля **Password** и **Confirm** в объяснении не нужны.

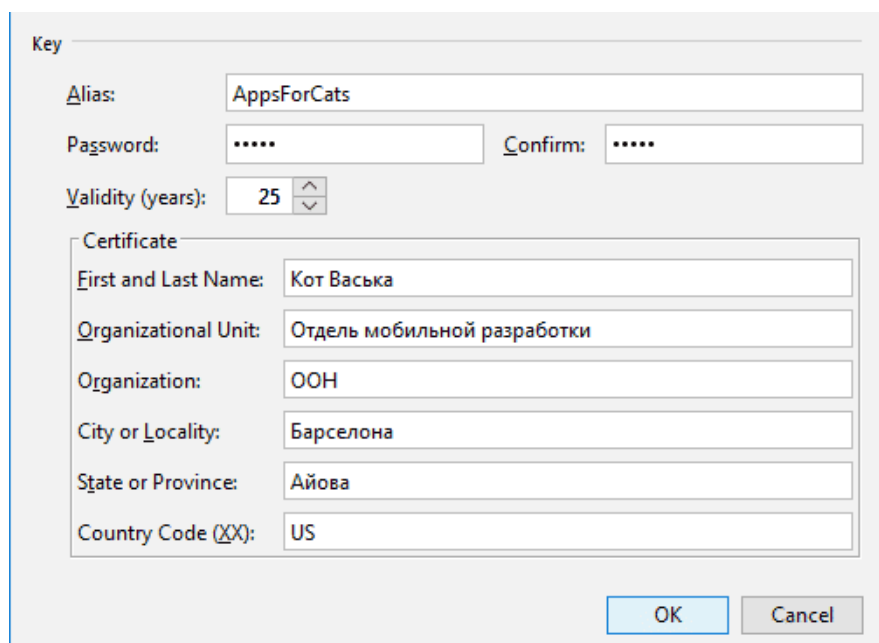
Теперь создаёте ключ для приложения. В поле **Alias** (Псевдоним) вводите понятное вам и котам название ключа. Не обязательно создавать псевдоним для каждого приложения, можете использовать один псевдоним для своих приложений и отдельные псевдонимы для приложений под заказ.

Для ключа также нужно создать пароль и подтвердить его.

Ключ рассчитан на 25 лет. Поле **Validity (years)** оставляем без изменений (если у вас нет веских причин в обратном).

Напоследок заполняете данные о себе.





Key

Alias: AppsForCats

Password: Confirm:

Validity (years): 25

Certificate

First and Last Name: Кот Васька

Organizational Unit: Отдель мобильной разработки

Organization: ООН

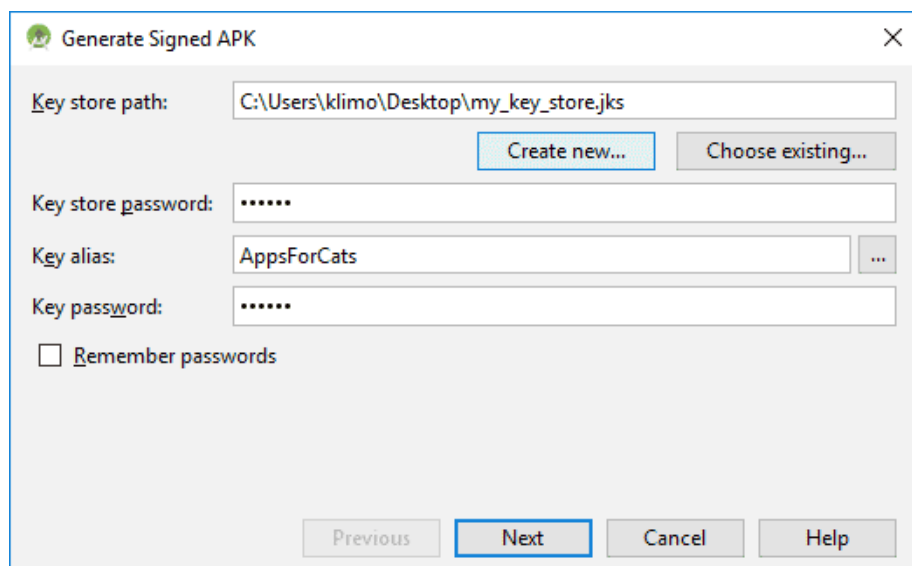
City or Locality: Барселона

State or Province: Айова

Country Code (XX): US

OK Cancel

Заполнив поля, вы вернёмся к самому первому окну мастера.



Generate Signed APK

Key store path: C:\Users\klimo\Desktop\my_key_store.jks

Create new... Choose existing...

Key store password:

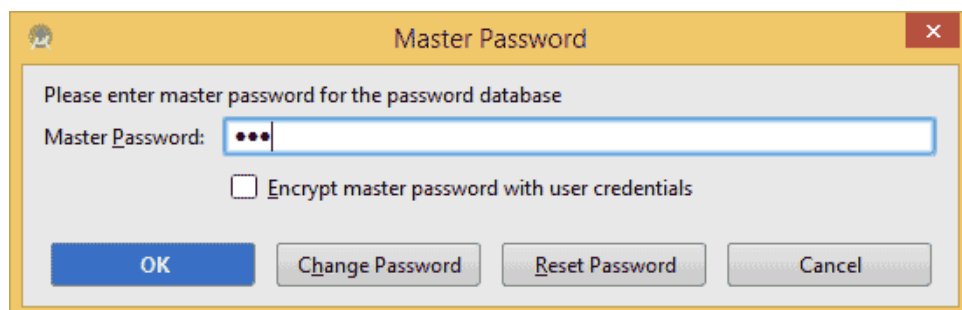
Key alias: AppsForCats

Key password:

☐ Remember passwords

Previous Next Cancel Help

Нажимаем на кнопку **Next** и в следующем окне вводим ещё один пароль для доступа к базе паролей.



Master Password

Please enter master password for the password database

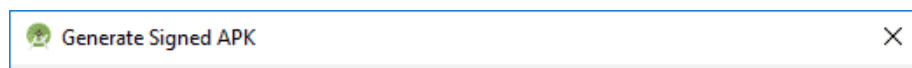
Master Password: ...

☐ Encrypt master password with user credentials

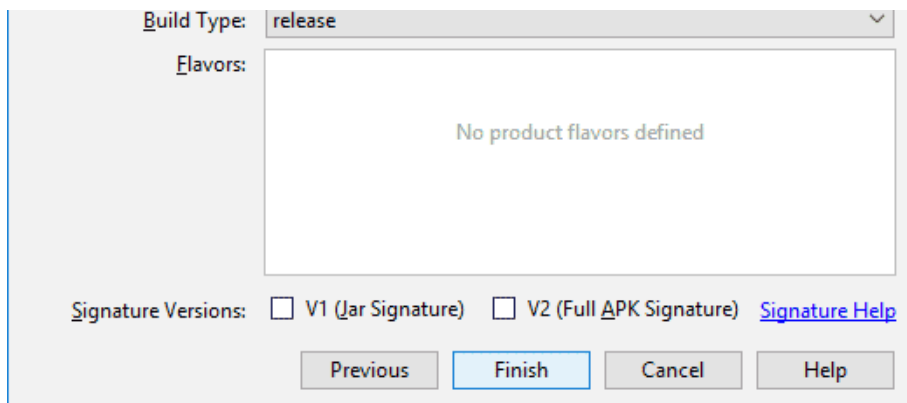
OK Change Password Reset Password Cancel

Возможно, этого шага у вас не будет. Он может появиться, если вы отметили флажком опцию "Запомнить пароль". Подробности не помню, разберётесь сами.

Последний шаг - нажать кнопку **Finish**.

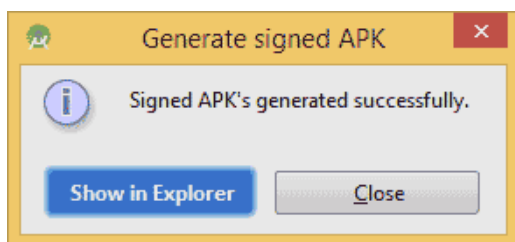


Generate Signed APK



Раньше это был последний шаг. Сейчас появились новые флажки **V1 (Jar Signature)** и **V2 (Full APK Signature)**. Отметьте как минимум первый вариант **V1** - это соответствует старому способу. Второй способ считается более надёжным в плане взлома и его можно использовать для поздних API, на ваше усмотрение (см. ниже).

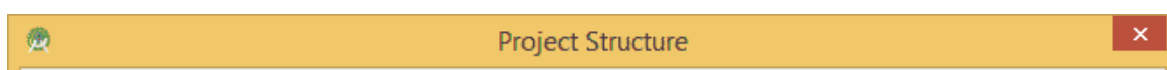
В результате сложных манипуляций с диалоговыми окнами у вас появится долгожданный APK - ваша прелесть, которая откроет дверь в мир богатства и счастья.

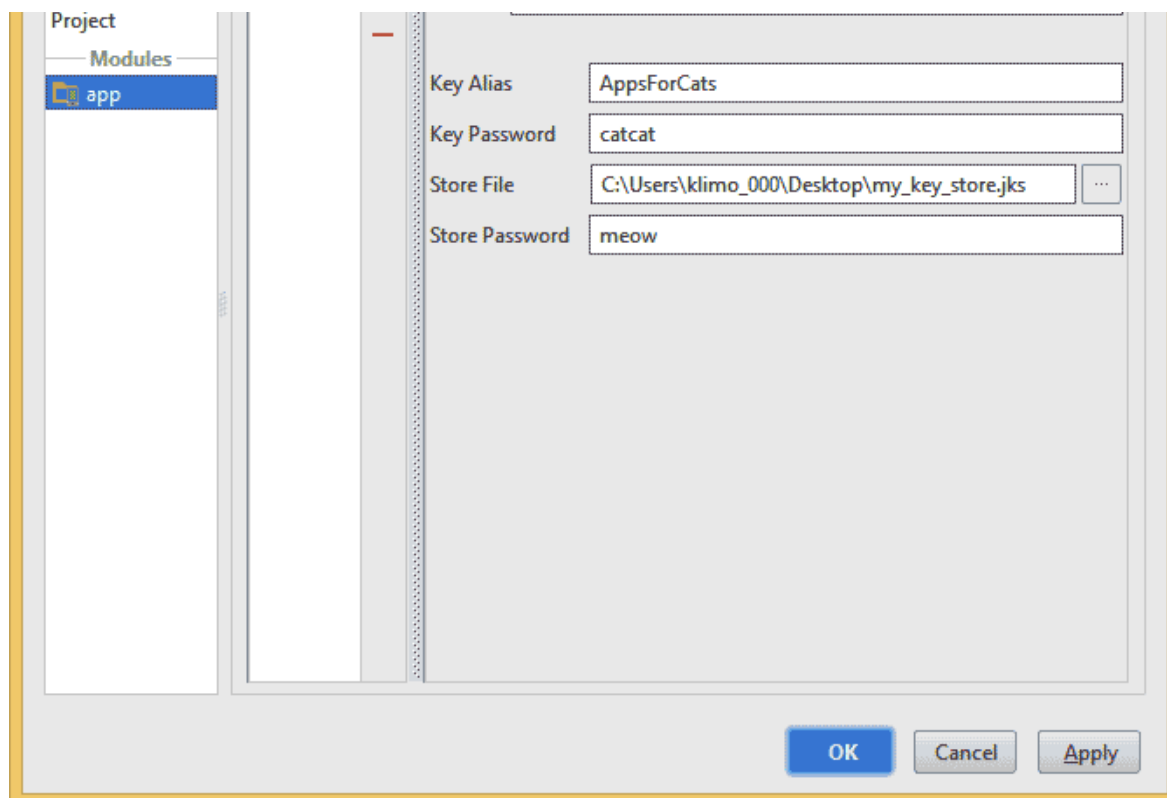


Нажав кнопку **Show in Explorer**, вы запустите Проводник на вашем компьютере с папкой, в которой находится подписанный файл.

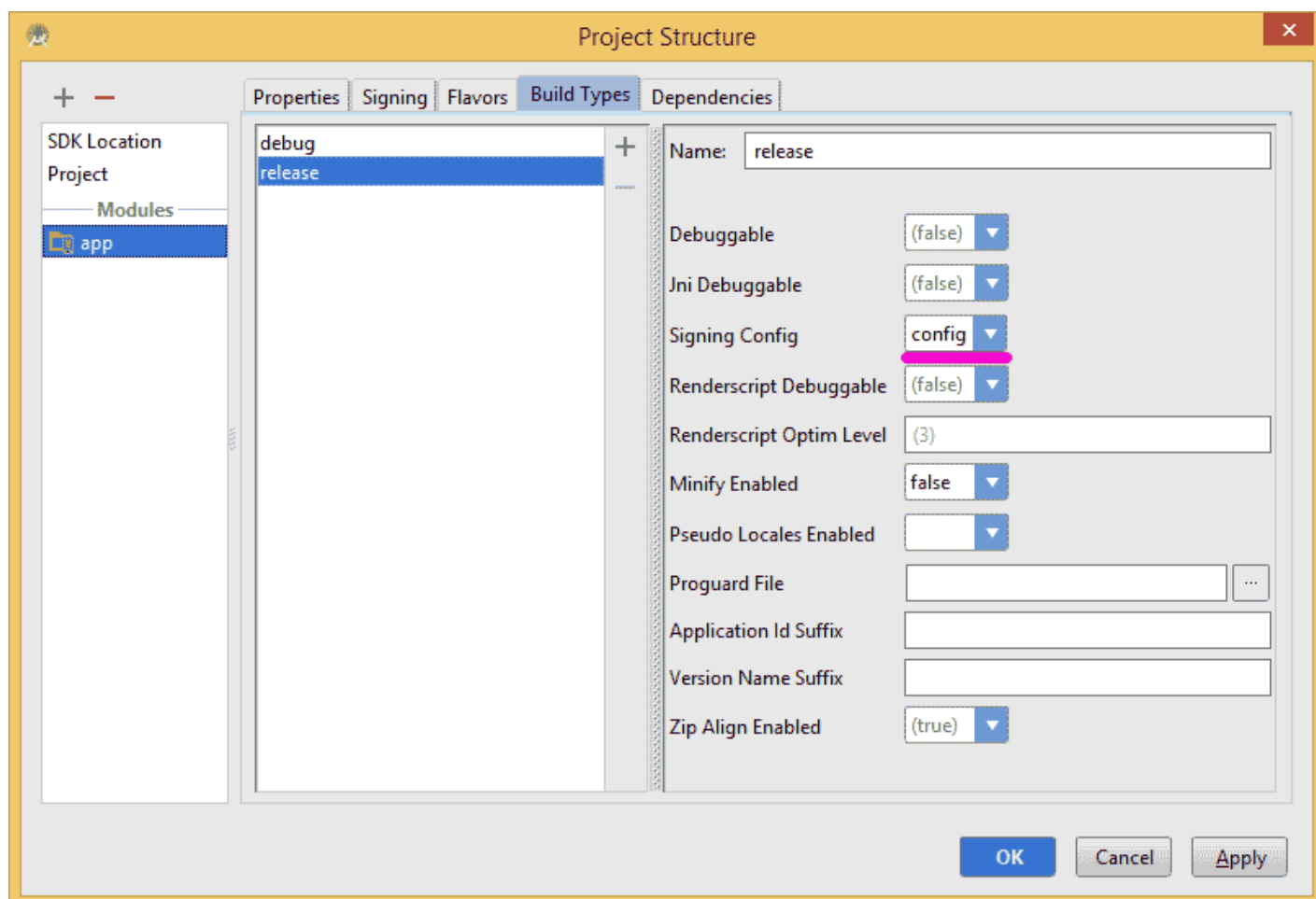
Будьте аккуратны с созданным ключом. Именно он является гарантией, что новая версия программы написана вами. Поэтому, если вы потеряете созданный ключ, вам придется выкладывать программу под другим именем с новым ключом.

В студии предусмотрен режим автоматического создания подписанного приложения. Щёлкните правой кнопкой мыши на папке **app** и в контекстном меню выберите пункт **Open Module Settings**. Выберите раздел **app** в секции **Modules**. Выберите вкладку **Signing**. Нажимаете на кнопку с плюсиком и заполняете поля.





Переходите на вкладку **Build Types** и выбираете сборку **release**. В выпадающем списке **Signing Config** выбираете только что созданную конфигурацию. По умолчанию она имеет имя **config**.



Нажимаем **OK** для сохранения результатов.

v2 Full APK

В 2017 году Google немного изменила процесс подписания. Теперь существуют две схемы получения подписи APK: v1 JAR и v2 Full APK.

Подпись v1 (который существовал с самого начала) не защищает некоторые части APK, такие как метаданные ZIP. Верификатор APK должен обрабатывать множество ненадёжных структур данных, а затем отбрасывать данные, которые не подписаны, что предоставляет большой простор для атаки. Кроме того, верификатор APK должен распаковать все сжатые записи, что тратит много времени и памяти. Для решения проблем была разработана новая версия v2 Full APK, которую вы встретите при подписании вашего приложения.

Схема v2 работает в Android 7.0 Nougat (API 25). Схема обеспечивает более быструю установку приложения и хорошую защиту от несанкционированных изменений в APK. Содержимое APK хешируется и подписывается, затем полученный блок подписи APK вставляется в APK.

Новый формат обратно совместим, поэтому APK, подписанные новой схемой, могут быть установлены на более ранних устройствах (которые будут просто игнорировать новую подпись), если эти APK также подписаны схемой v1.

В старых приложениях я оставляю флажок у первой версии. Возможно, позже заставят переходить на вторую версию принудительно. Важно учитывать, что подписывать схемой v1 нужно до подписания схемой v2, поскольку APK не пройдет проверку по схеме v2, если он будет подписан дополнительными сертификатами после подписания схемой v2.

Строго говоря, вы можете арк-файл выложить у себя на сайте, и все ваши посетители могут его скачать и установить на телефон. Но это как-то несолидно в наш век нанотехнологий. Поэтому переходим к следующему шагу.

Шаг второй. Надо, Федя, надо

Следующий шаг очень неприятный. Вам нужно подарить 25 вечнозеленых чужому человеку. Чтобы вам было не так обидно, данную операцию назвали регистрационным взносом. Вам понадобится кредитная карточка с указанной суммой. Учтите, что Visa Electron, а уж тем более дисконтная карточка сети магазинов "Перекресток" вам не подойдут. Если у вас уже есть нужная карточка, то пропускаете этот абзац. Остальным могу посоветовать завести QIWI-кошелёк и там завести виртуальную карточку. Именно так я и поступил, так как идти в банк и писать всякие заявления было неохота.

Идём на [страницу разработчиков](#), заполняем необходимые поля и расстаёмся с нужной суммой. Если операция пройдет успешно, то вы сможете продвинуться дальше. В моем случае мне дважды отказывали, так как я пожадничал и положил на карточку меньшую сумму, чем требовалось (не учёл комиссию). Если вы позже доложили необходимую сумму, то не ждите, что её автоматически у вас спишут. Снова зайдите на страницу разработчика и отредактируйте данные о карточке (нужно еще раз написать код безопасности).

приложение разработчика - это ваш единственный путь к новым возможностям Android.

Имя разработчика:
Появится под названием приложения.

Адрес эл. почты:

URL веб-сайта:

Номер телефона:
Используйте формат: +7-123-456-7890. [Зачем нам эти данные?](#)

Новости: ☐ Присылать мне информацию о новых возможностях разработки и распространения приложений для Android Маркета

[Продолжить »](#)

© 2011 Google - [Условия использования Android Маркета](#) - [Политика конфиденциальности](#)

alexander.klimoff@gmail.com | [Главная страница](#) | [Справка](#)



Зарегистрируйтесь как разработчик

Регистрационный взнос: 25,00 \$

Регистрационный взнос позволяет размещать программы на Android Маркете. Имя и платежные реквизиты, с которыми вы зарегистрировались, будут использоваться в [Соглашении о распространении ПО через Android Маркет](#). Внимательно проверьте эти данные!

Уплатить регистрационный взнос с помощью



Fast checkout through Google

[Продолжить »](#)

[Вернуться для изменения профиля](#)

© 2011 Google - [Условия использования Android Маркета](#) - [Политика конфиденциальности](#)

alexander.klimoff@gmail.com | [Главная страница](#) | [Справка](#)



Прочитайте и примите условия [Соглашения о распространении ПО через Android Маркет](#)

android

Russia

Соглашение разработчика о распространении продуктов на Android Маркете

Определения

Google – корпорация Google, зарегистрированная в штате Делавэр, США, главный офис которой расположен по адресу: 1600 Amphitheatre Parkway, Mountain View, CA 94043, США

Лицо, зарегистрированное для этого аккаунта:

Alexander Klimov

RU

☒ Я принимаю условия соглашения и хочу связать данные моей кредитной карты и данные аккаунта, указанные выше, с [Соглашением о распространении ПО через Android Маркет](#).

[Принимаю, продолжить »](#)

[Отменить эту регистрацию](#)

Ваш регистрационный взнос не будет взыскан

© 2011 Google - [Условия использования Android Маркета](#) - [Политика конфиденциальности](#)

Шаг третий. Со счастливым концом

Если платёж прошел успешно, то ссылка на следующий шаг будет доступна и вы попадаете в специальный личный кабинет, где можете добавлять свои приложения.

Весь интерфейс на русском. Поэтому трудностей у вас не возникнет. Заливаем подготовленный APK-файл, а также необходимые картинки-скриншоты и значок

Редактирование приложения

Опубликовать
Сохранить

Сведения о продукте
Файлы APK

Загрузка ресурсов

Скриншоты
не менее двух

Выберите изображение

Файл не выбран

Требования:
320 x 480, 480 x 800 или 480 x 854;
1280 x 720, 1280 x 800;
24-битный PNG или JPEG (без
альфа-канала);
без рамки, под обрез,
можно загрузить скриншоты с
горизонтальной ориентацией. Их
уменьшенные копии будут
отображаться вертикально, но,
нажав на них, пользователь
откроет изображение в исходной
ориентации.

Значок приложения в
высоком разрешении
[Подробнее](#)

Выберите изображение

Файл не выбран

Требования:
– 512 x 512;
– 32-битный PNG или JPEG;
– максимальный размер: 1024 КБ.

Значок для рекламы
необязательно

Выберите изображение

Файл не выбран

Требования:
180 (ш) x 120 (в)
24-битный PNG или JPEG (без
альфа-канала)
Без рамки, под обрез

Значок для раздела
"Рекомендуемые"
необязательно
[Подробнее](#)

Выберите изображение

Файл не выбран

Требования:
– 1024 x 500;
– 24-битный PNG или JPEG (без
альфа-канала);
– изображения будут уменьшены
до размера "мини" или "микро".

Рекламное видео
необязательно

Добавьте ссылку на видео
<http://>

Требования:
– Поддерживаются только

alexander.klimov@gmail.com | [Главная страница](#) | [Справка](#) | [Android.com](#) | [Войти](#)

Ваша регистрация на Android Маркете одобрена!

Теперь вы можете добавлять программы на Android Маркет.

Редактирование приложения

Опубликовать
Сохранить

Сведения о продукте
Файлы APK

Активные
нет

Новая
Код версии: 1
Версия: 1.0
[развернуть >](#)

Уровень API: 8-15+
Поддерживаемые экраны: small-large
Текстуры OpenGL: все

[Загрузить APK](#)

Активны ранее
нет

[Перейти к простому режиму >](#)

[Активировать](#) | [Удалить](#)

alexander.klimov@gmail.com | [Главная страница](#) | [Справка](#) | [Android.com](#) | [Войти](#)

Ваша регистрация на Android Маркете одобрена!

Теперь вы можете добавлять программы на Android Маркет.

Редактирование приложения

Опубликовать
Сохранить

Сведения о продукте
Файлы APK

Загрузка ресурсов

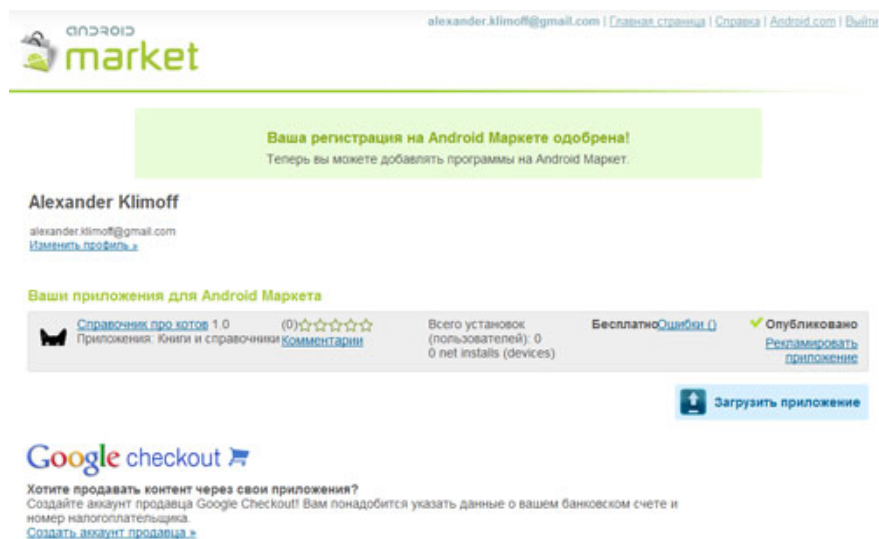
Скриншоты
не менее двух
[добавить еще](#)

Требования:
320 x 480, 480 x 800 или 480 x 854;
1280 x 720, 1280 x 800;
24-битный PNG или JPEG (без
альфа-канала);
без рамки, под обрез,
можно загрузить скриншоты с
горизонтальной ориентацией. Их
уменьшенные копии будут
отображаться вертикально, но,
нажав на них, пользователь
откроет изображение в исходной
ориентации.

Значок приложения в
высоком разрешении
[Подробнее](#)

Требования:
– 512 x 512;
– 32-битный PNG или JPEG;
– максимальный размер: 1024 КБ.

снова удалять. Иногда можно нажимать на кнопку **Сохранить**. Если вы сделаете что-то не так, то на странице появятся предупреждающие надписи. Если таких надписей нет, то можете смело нажимать на кнопку **Публиковать**. Всё! Ваше приложение доступно всему миру. За вами уже выехали.



Недавно в Google Play Developer Console добавили возможность тестирования приложений среди определённых пользователей. Если раньше вы загружали своё приложение и оно сразу становилось доступным всем, то теперь добавлены два промежуточных шага.

При загрузке новой версии приложения вам нужно выбрать раздел:

- АЛЬФА-ТЕСТИРОВАНИЕ
- БЕТА-ТЕСТИРОВАНИЕ
- РАБОЧАЯ ВЕРСИЯ

Если вы загрузите приложение в раздел Альфа-тестирования, то потом можете перевести его в бета-тестирование или сразу в Рабочую версию. Соответственно, из бета-тестирования можно перевести сразу в Рабочую версию. Обратно нельзя.

Если программа находится в стадии тестирования, то оно доступно только тестерам, другие пользователи не смогут найти вашу программу ни через поиск, ни по прямой ссылке.

Вам следует создать специальное сообщество в Google+ (указывается в настройках приложения) и пригласить туда нужных людей. Доверенные лица смогут затем перейти по ссылке play.google.com/apps/testing/com.yourdomain.package.

Ищет милиция, ищут пожарные

Когда вы публикуете свою программу, то она сразу попадает в магазин приложений без всякой модерации. Но найти её будет там не просто. Даже поиск по магазину вам не поможет. Нужно какое-то время, чтобы данные в магазине обновились и поисковый робот заметил вашу программу. Но я помогу вам бесплатным советом, как быстро найти свою программу в Google Play. А всё очень просто. Вспомните имя вашего пакета, который использовался при создании приложения - он и является ключом к разгадке. Заходим на сайт и вбиваем нужный адрес с указанием пакета в качестве ID:

<https://play.google.com/store/apps/details?id=ru.alexanderklimov.crib>

И вы сразу окажетесь в нужном месте и можете раздать ссылку своим друзьям. В последнее время добавление/обновление программы происходит достаточно медленно. Поэтому не стоит сразу искать

Где деньги, Зин

Какой вы быстрый, однако. Есть два способа зарабатывания денег на своем приложении. Либо надо зарегистрироваться как продавец, либо как участник рекламной сети AdMob.

Открыть страницу приложения в Google Play

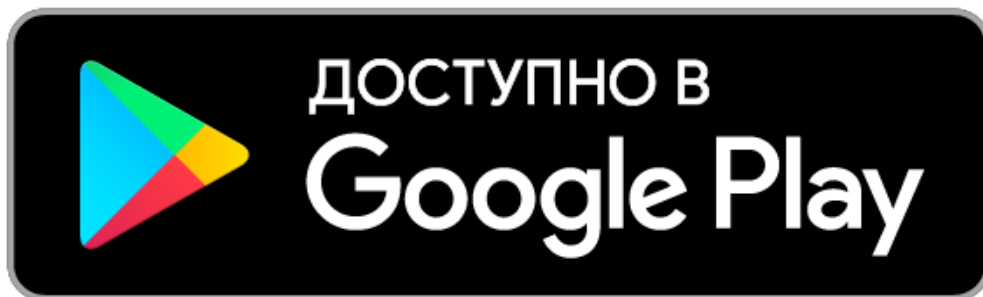
Пользователи скачивают приложение и забывают отблагодарить разработчика. Надо аккуратно ему напомнить о гражданском долге - поставить высшую оценку вашей программе. Добавляете кнопку с текстом **Оцените наше приложение** и пишете небольшой код:

```
Intent intent = new Intent(Intent.ACTION_VIEW);
intent.setData(Uri.parse("market://details?id=ru.alexanderklimov.crib"));
startActivity(intent);
```

Учтите, что на эмуляторе код не сработает, так как в нём нет приложения Google Play. А пока можете зайти через телефон по указанному адресу и поблагодарить меня. Коты вам скажут **Спасибо!**.

Продвижение

На странице https://play.google.com/intl/ru_ru/badges/ на вкладке **Badge Generator** вы можете указать необходимые параметры, формирующие код для кнопки-баннера, который затем можете использовать в своём блоге или других местах. Например, так:



Автоматическое обновление

Если вы создали новую версию программы, исправив различные баги и добавив новые фотографии кота, то вам нужно в манифесте увеличить на единицу номер версии (атрибут **versionCode**) и заменить **versionName** для себя (будет показана на странице Google Play). В последних версиях студии данные свойства находятся теперь не в манифесте, а в файле **build.gradle** модуля вашего приложения. Закачайте новую версию на Google Play и пользователи получат обновление в автоматическом режиме.

Следим за отзывами

Установите на своём устройстве приложение [Google Play Developer Console](#), чтобы не пропустить новый отзыв на вашу программу. Также вы можете просматривать статистику.

Меняем пароли хранилища и псевдонима ключа

Допустим, вы продали свою программу с исходниками другой компании. Чтобы она могла выкладывать обновления программы, компания должна подписывать приложение тем же ключом, которым подписывали вы. Иначе программа будет считаться другой и придётся менять название пакета. Но тогда старые пользователи не смогут получить обновления.

Но если вы все свои программы подписываете одним и тем же ключом и паролем, например **cat cat**, то компания может подписать этим же ключом и другие ваши приложения, разместив свои программы с таким же именем пакета, и вы никому ничего не докажете.

Поэтому вам нужно позаботиться о смене ключа для передачи новому владельцу.

Предположим наше хранилище имеет структуру:

```
Имя хранилища (keystore):
    old.keystore
Пароль от хранилища:
    cat1
Псевдоним:
    my_alias
Пароль от псевдонима:
    cat2
```

Сделайте копию вашего хранилища и сохраните его в другом месте. Это надо было сделать ещё при первом создании, потому что при потере хранилища вы не сможете восстановить доступ к своим программам при обновлении.

Сделайте копию вашего хранилища ещё раз и переименуйте его, например, **new.keystore**. С ним и будем работать.

Далее вам нужно изменить пароль хранилища, изменить псевдоним и изменить пароль псевдонима. Полученный файл передать новому владельцу.

Запускаем утилиту **keytool** с командой:

```
keytool -storepasswd -keystore new.keystore
```

Вам будет предложено ввести текущий пароль, а затем ввести новый пароль и повторить его. Приблизительно так:

```
Enter keystore password:
New keystore password:
Re-enter new keystore password:
```

Первая часть задачи выполнена, пароль от хранилища изменён.

Если вы хотите также изменить и пароль от псевдонима, то снова запускаем утилиту с командой:

```
keytool -keypasswd -keystore new.keystore -alias my_name
```

Вас попросят ввести текущий пароль от хранилища (ваш новый пароль), затем пароль для псевдонима. Вы можете ввести новый пароль и он заменит старый пароль.

```
Enter keystore password:
Enter key password for <my_alias>
```

Если изменения пароля вам недостаточно и вы хотите изменить имя псевдонима (может вы использовали имя любимой кошки, зачем другим об этом знать), то продолжаем работу.

Запускаем команду:

```
keytool -changealias -keystore new.keystore -alias my_alias -destalias my_new_alias
```

Вас попросят ввести пароль от хранилища, затем пароль для нового псевдонима (текущий пароль), затем новый пароль и повторить его. Имя псевдонима будет изменено.

Итак, нам понадобилось три шага, чтобы создать новое хранилище и псевдоним для передачи чужому человеку. Новый владелец должен проделать тоже самое, чтобы быть уверенным, что вы не воспользуетесь изменённым файлом в своих целях. Впрочем, это уже его проблемы.

Подписываем готовое приложение

Такой случай может подвернуться, когда у вас утеряны исходники и есть только APK. Скорее всего это актуально для пиратов, которые переподписывают другие приложения (не делайте так с чужими программами).

Сначала поменяйте расширение с apk на zip. В архиве удалите папку **META-INF**. Восстановите расширение. Вы удалили старую подпись.

Теперь нужно подписать приложение новым ключом. Введите команду.

```
jarsigner -keystore keystore-file.jks -storepass keystore_password -keypass alias_password --signedjar signed-apk-file.apk apk-file.apk alias_name
```

В успешном случае получите сообщение, что приложение подписано. Далее выполняем ещё одну команду.

```
ANDROID_SDK_PATH/build-tools/LAST_BUILD_TOOLS_VERSION/zipalign -v 4 signed-apk-file.apk aligned-apk-file.apk
```

В результате должен получиться APK-файл, подписанный новым ключом. Сам ни разу не применял.

Хранение ключей у Гугла

В 2017 году Google добавил новую возможность хранить ключи в облачном хранилище. Основное отличие заключается в том, что вы подписываете приложение специальным ключом загрузки, который Google проверяет и удаляет, заменяя его оригинальным ключом подписи приложения, который вы предоставили.

С его помощью можно управлять ключами подписи приложений как для новых, так и для опубликованных приложений, которые будут храниться у Google в их собственном хранилище ключей. Чтобы присоединиться к этой программе, необходимо подписаться на неё в своей Google Play Console. Стоит отметить, что отписаться от неё уже будет невозможно.

Такой способ очень полезен - при потере хранилища ключей Google Play App Signing позволит сбросить ключ для установки нового. Вам не придётся публиковать приложение повторно с новым именем пакета и ключом.

[Как управлять ключами подписи приложений - Справка - Play Console](#)

[Android-keystore-password-recover by MaxCamillo](#) - если вы утратили пароль, то попробуйте воспользоваться данным инструментом. Сам не пользовался, поэтому рассказывать не буду.

Реклама

Реклама

