# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc345251089)

[ВВЕДЕНИЕ 3](#_Toc345251090)

[1 ПОСТАНОВКА ЗАДАЧИ 4](#_Toc345251091)

[2 ОПИСАНИЕ ПРОГРАММЫ 6](#_Toc345251092)

[2.1 Общие сведения 6](#_Toc345251093)

[2.2 Структура программы 6](#_Toc345251094)

[2.3 Входные и выходные данные 7](#_Toc345251095)

[2.4 Вызов и загрузка 7](#_Toc345251096)

[3 ОПИСАНИЕ ПРИМЕНЕНИЯ 8](#_Toc345251097)

[3.1 Требования к техническим и программным средствам 8](#_Toc345251098)

[3.2 Требования техники безопасности и охраны труда при эксплуатации программы 8](#_Toc345251099)

[3.3 Энерго- и ресурсосбережение 9](#_Toc345251100)

[4 РУКОВОДСТВО ПРОГРАММИСТА 11](#_Toc345251101)

[5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 14](#_Toc345251102)

[ЗАКЛЮЧЕНИЕ 17](#_Toc345251103)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 18](#_Toc345251104)

[ПРИЛОЖЕНИЕ А ТЕКСТ ПРОГРАММЫ 20](#_Toc345251105)

# ВВЕДЕНИЕ

В современном мире наибольшую ценность имеет информация. Как сказал Натан Ротшильд: «Кто владеет информацией — тот владеет миром». Совершенно естественно, что доступ к ценной информации необходимо как-то ограничивать. Для решения этой проблемы были придуманы пароли. В общем случае пароль – это некий набор символов, известный лишь владельцу информации. Этот набор символов и обеспечивает доступ к информации. Очевидно главное требование пароля – надежность. Обеспечить надежность пароля можно исключив «человеческий фактор» при его создании, т. к. обычно люди не придумывают случайную последовательность, а значит имея определенное количество информации о человеке пароль можно подобрать.

Цель курсового проекта – создать программное средство (приложение) для платформы Android, которое позволяло бы генерировать пароли, согласно заранее определенным требованиям прямо на телефоне, а также производить оценку пароля придуманного лично.

# ПОСТАНОВКА ЗАДАЧИ

Для достижения цели курсового проектирования, а именно, создания программного средства (приложения) для платформы Android способного генерировать пароли согласно требований пользователя, а также проводить проверку надежности введенных паролей, необходимо решить следующие проблемы:

* разработать алгоритм генерации пароля;
* разработать алгоритм оценки введенного пароля

Генерация пароля будет производится с помощью специального класса java.util.Random. А для, непосредственно, оценки пароля, будет подсчитываться количество символов, а так же их тип (строчная буква, прописная буква и т. п.) Информация, необходимая для приложения, будет получаться непосредственно в процессе работы путем указания требуемых настроек пароля (в случае его генерации) либо ввода пароля для оценки - входные данные. Выходные данные – сгенерированный пароль, либо графическая оценка введенного пароля. Приложение разрабатывается для широкого круга людей, а, следовательно, должно быть максимально простым в использовании (быстрая установка, логичные и понятные настройки и т. д.). Так как приложение разрабатывается под мобильную платформу, оно должно иметь не перегруженный элементами управления интерфейс, в идеале их минимальное количество.

Аппаратные требования:

* процессор с частотой не менее 500 MHz
* объем оперативной памяти не менее 256 Mb
* дисплей с диагональю не менее 3”
* свободное место в памяти устройства непосредственно для исполняемого файла приложения – 1 Mb

Система Android в качестве платформы для реализации задачи была выбрана по совокупности следующих факторов.

Во-первых, согласно данных компании International Data Corporation (аналитическая фирма, специализирующаяся на исследованиях рынка информационных технологий) на рынке мобильных платформ представлены всего несколько крупных компаний со своими разработками: Google Android — 40%, Nokia Symbian — 21%, Apple iOS — 18%, RIM BlackBerry — 14%, Windows Phone - 4%. На долю всех остальных участников пришлось 3% рынка.

Во-вторых, инструменты для разработки Android приложений предоставляются абсолютно бесплатно и не накладывают ограничений на программную среду, в отличие, например, от Apple iOS, где для разработки необходимо использовать компьютер под управлением операционной системы MacOS.

В-третьих, платформа Android продолжает свое стабильное развитие в отличие от Nokia Symbian, у которой в последнее время появились серьезные проблемы с поддержкой от производителей телефонов.

Таким образом, написание приложения для операционной системы Android позволит максимальному числу людей использовать его, а также минимизирует стоимость разработки.

При создании приложения необходимо использовать интегрированную среду разработки Eclipse, т. к. именно она рекомендуется компанией Google как основное средство разработки проектов для операционной системы Android, кроме того среда бесплатна, в отличие от большинства аналогов, что положительно скажется на стоимости разработки. В обязательном порядке следует использовать Android Software Development Kit - свободно распространяемый компанией Google набор инструментов для разработки Android приложений. Непосредственно сам проект будет написан на языке Java, на базе бесплатного Oracle Java Development Kit – набор инструментов для программирования мобильных приложений на языке Java.

Выбор средств разработки по большему счету сводится к выбору среды разработки, т. к. инструментов альтернативных Android Software Development Kit Android и Oracle Java Development Kit не существует.

# ОПИСАНИЕ ПРОГРАММЫ

## Общие сведения

Основанием для разработки программы «Password Generator» является выполнение задания курсового проектирования. Разработчиком программы является Орлов Юрий Николаевич, учащийся группы 281021 учреждения образования «Институт информационных технологий «Белорусского государственного университета информатики и радиоэлектроники»».

Разрабатываемая программа должна осуществлять генерацию пароля, на основании требований пользователя, а также проводить оценку пароля, введенного пользователем.

Для функционирования программы необходимо использовать мобильный телефон с установленной операционной системой Android 4.0 Ice Cream Sandwich либо более поздней версии.

Программа написана на языке Java в интегрированной среде разработки Eclipse Juno с использованием инструментов Java Development Kit и Android Software Development Kit.

## Структура программы

При реализации данной программы использовался метод восходящего проектирования - способ разработки программ, при котором программирование ведется методом «снизу вверх», от деталей к общему решению. Алгоритм решения задачи разбивается на элементарные части, подзадачи, которые решаются независимо друг от друга. При этом составляется план решения всей задачи, пунктами которого и являются выделенные части. После того, как все подзадачи решены, они объединяются, образуя общее решение – итоговый проект.

Ниже представлен список основных методов, используемых в программе, с описанием выполняемых ими действий:

* String generatePassword(int passwordLen, SharedPreferences preferences) – генерация пароля, на основе переданных данных;
* int getPasswordRating(String password) – оценка надежности введенного пароля;
* void onCreate(Bundle savedInstanceState) – создание главного окна программы;
* void onResume() – метод, выполняющийся при обновлении окна программы;
* void onStart() – метод, выполняющийся при появлении окна программы;
* boolean onCreateOptionsMenu(Menu menu) – создание главного меню;
* boolean onOptionsItemSelected(MenuItem item) – определение выбранного пункта меню;
* Dialog onCreateDialog(int id) – создание диалога «О программе»
* static String getVersionName(Context context) – определение версии программы;

## Входные и выходные данные

Входными данными являются:

* длина пароля
* алфавит пароля
* самостоятельно введенный пароль для проверки

Выходными данными являются:

* сгенерированный пароль
* оценка сложности введенного пароля

## Вызов и загрузка

После установки программу можно запустить нажатием на её ярлык на рабочем столе либо нажатием на ярлык программы в общем меню. Сразу после запуска доступно окно выбора режима работы программы.

# ОПИСАНИЕ ПРИМЕНЕНИЯ

## Требования к техническим и программным средствам

Для полноценного функционирования программы и выполнения всех её функций необходим мобильный телефон со следующими аппаратными характеристиками:

* процессор с тактовой частотой не менее 500 MHz;
* объем оперативной памяти не менее 256 Mb;
* дисплей с диагональю не менее 3``;
* свободное место в памяти устройства непосредственно для исполняемого файла приложения – 1 Mb.

На мобильном телефоне должна быть установлена операционная система Android 4.0 Ice Cream Sandwich либо более поздняя версия.

## Требования техники безопасности и охраны труда при эксплуатации программы

Так как программа предназначена для использования на мобильном телефоне, необходимо соблюдать основные принципы безопасного использования мобильных телефонов:

* используйте только рекомендованные аккумуляторы и зарядные устройства. Несовместимые аккумуляторы и зарядные устройства могут привести к серьезным травмам и повреждению устройства;
* никогда не сжигайте использованные аккумуляторы или телефон;
* никогда не помещайте аккумуляторы и телефоны внутрь или на поверхность нагревательных приборов, например, микроволновых печей, духовок или радиаторов. При перегреве аккумуляторы могут взорваться;
* не сдавливайте и не прокалывайте аккумулятор. Избегайте внешнего давления на аккумулятор, так как это может привести к внутреннему короткому замыканию и перегреву;
* не используйте телефон на автозаправочных станциях (станциях техобслуживания), а также вблизи емкостей с топливом или химическими веществами. Выключайте телефон в соответствии с предупреждающими знаками и инструкциями. Телефон может стать причиной взрыва или пожара в непосредственной близости от объектов хранения топлива или химических веществ, а также в местах выполнения взрывных работ. Не храните и не перевозите горючие жидкости, газы или взрывчатые вещества вместе с телефоном, его компонентами или аксессуарами;
* не используйте телефон при управлении автомобилем и соблюдайте все правила, ограничивающие использование мобильных телефонов при вождении автомобиля. Используйте устройства громкой связи для обеспечения безопасности;
* телефон может создавать помехи в работе медицинского оборудования в больницах или других медицинских учреждениях. Следуйте всем требованиям, предупреждениям и указаниям медицинского персонала.

## Энерго- и ресурсосбережение

Основной ресурс при использовании мобильно телефона – заряд аккумуляторной батареи. Чем чаще происходит зарядка батареи, тем больше расходуется электричество. Следуя нескольким простым правилам, можно существенно увеличить время разрядки аккумуляторной батареи, тем самым уменьшив расход электроэнергии.

Для экономии заряда батареи в мобильных телефонах, работающих под управлением операционной системы Android необходимо выполнять следующие правила:

* включать GPS только в случае необходимости и выключать сразу после завершения требуемого действия;
* включать Bluetooth непосредственно перед использованием и выключать сразу после завершения требуемого действия;
* отключить режим Wireless Network Positioning, в **Settings > Location > Use wireless networks;**
* отключить режим Always-On Mobile Data. Режим Always-On Mobile Data установлен по умолчанию, но может быть отключен в **Settings > Wireless & networks > Mobile networks > Enable always-on mobile data.** Этот режим позволяет телефону все время быть соединенным с сетью, но для большинства пользователей это не критично. Режим не стоит выключать, если используется больше 10 приложений, требующих активного интернет соединения.
* настроить автоматическое отключение экрана. После некоторого времени бездействия экран автоматически выключается. Это время называется Screen Timeout. Настроить этот параметр можно в **Settings > Screen & display > Screen timeout;**
* включить режим автоматической настройки яркости в **Settings > Screen & display > Brightness;**
* **использовать темные цветовые схемы оформления в совокупности с минимальным количеством виджетов;**
* настроить интервалы обновлений для приложения требующих автоматических обновлений своих данных в **Settings > Accounts & sync.**

Контролировать расход заряда батареи можно средствами встроенной функции, отображающей использование батареи приложениями. Статистика использования батареи доступна в **Settings > About phone > Battery > Battery.**

# РУКОВОДСТВО ПРОГРАММИСТА

Для установки программы и выполнения всех её функций необходим мобильный телефон со следующими аппаратными характеристиками:

* процессор с тактовой частотой не менее 500 MHz;
* объем оперативной памяти не менее 256 Mb;
* дисплей с диагональю не менее 3”;
* свободное место в памяти устройства непосредственно для исполняемого файла приложения – 1 Mb.

На мобильном телефоне должна быть установлена операционная система Android 4.0 Ice Cream Sandwich либо более поздняя версия.

В целях исключения возможности использования приложения на телефонах, не соответствующих минимальным системным требованиям, дистрибуция осуществляется через магазин приложений Google Play. Приложение можно найти по прямой ссылке https://play.google.com/store/apps/details?id=by.bsuir.passgen, либо используя поисковый запрос «bsuir».

На рисунке 4.1 изображено предупреждение системы Google Play при попытке установить приложение на телефонный аппарат Huawei U8800 с установленной операционной системой Android 2.2 Froyo

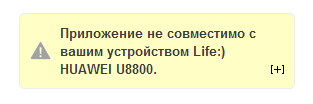


Рисунок 4.1 - Предупреждение о несовместимости с устройством

Для старта процесса установки, достаточно нажать кнопку «Install» (рис. 4.2)

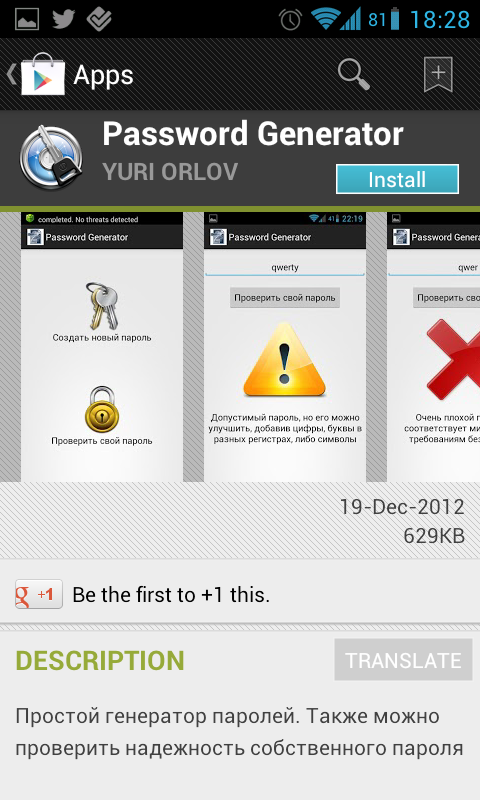


Рисунок 4.2. Установка программы

и подтвердить согласие на загрузку и установку (рис. 4.3)

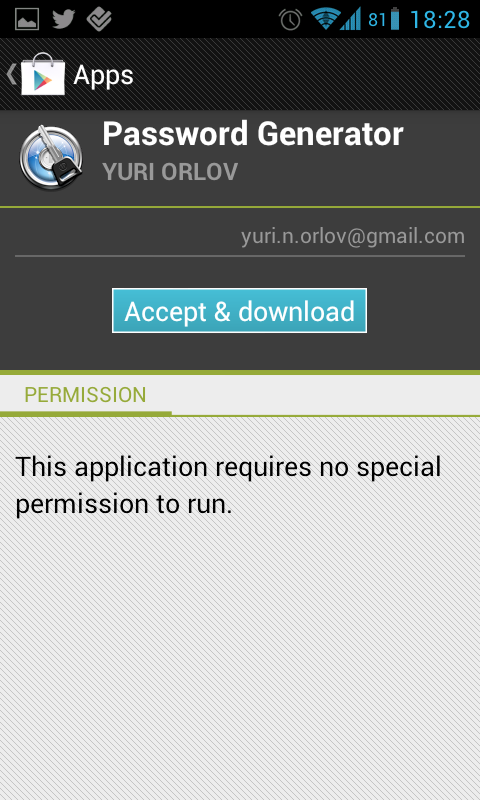


Рисунок 4.3. Загрузка приложения

Как только приложение будет загружено и установлено на телефон, можно начинать работу, никакие дополнительные настройки не требуются (рис. 4.4)

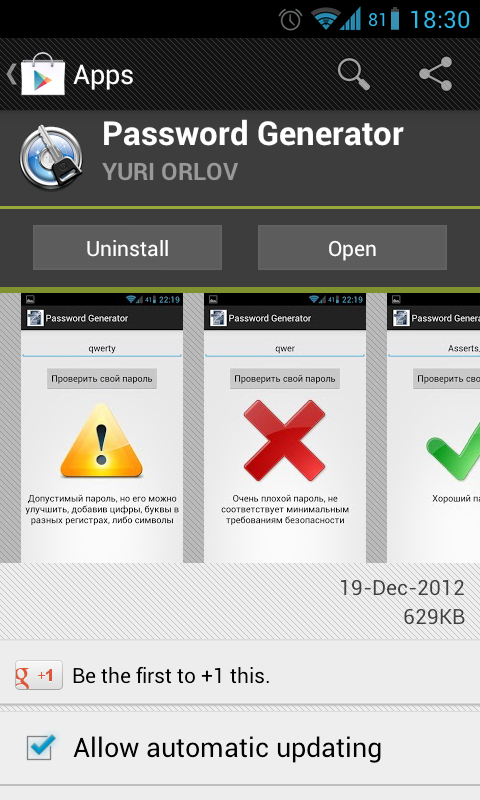


Рисунок 4.4 Приложение загружено и установлено

Вывод каких-либо сообщений программисту не предполагается.

# РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Для полноценного функционирования программы и выполнения всех её функций необходим мобильный телефон со следующими аппаратными характеристиками:

* процессор с тактовой частотой не менее 500 MHz
* объем оперативной памяти не менее 256 Mb
* дисплей с диагональю не менее 3``
* свободное место в памяти устройства непосредственно для исполняемого файла приложения – 1 Mb

На мобильном телефоне должна быть установлена операционная система Android 4.0 Ice Cream Sandwich либо более поздняя версия.

Запуск программы осуществляется нажатием её пиктограмму (рис. 5.1)

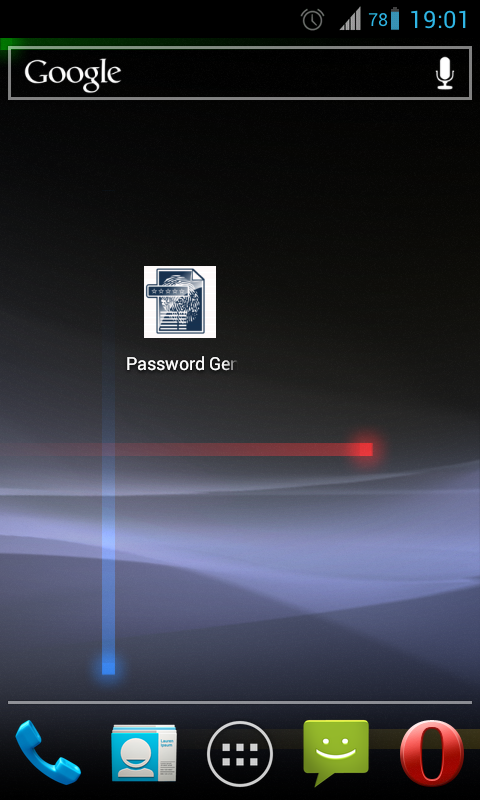


Рисунок 5.1 Пиктограмма программы

Приложение может работать в одном из двух режимов: генерация пароля, или оценка надежности пароля (рис. 5.2)

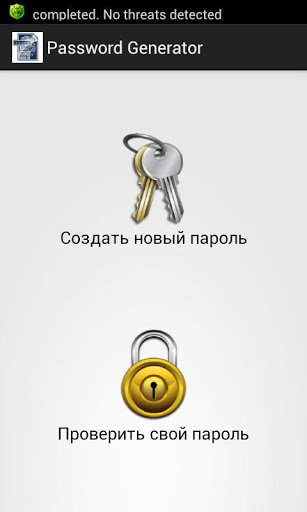


Рисунок 5.2 Выбор режима работы

Для оценки пароля, необходимо ввести пароль в специальное поле, после чего нажать кнопку «Проверить свой пароль», результат проверки будет отображен в графическом виде с пояснениями (рис. 5.3)

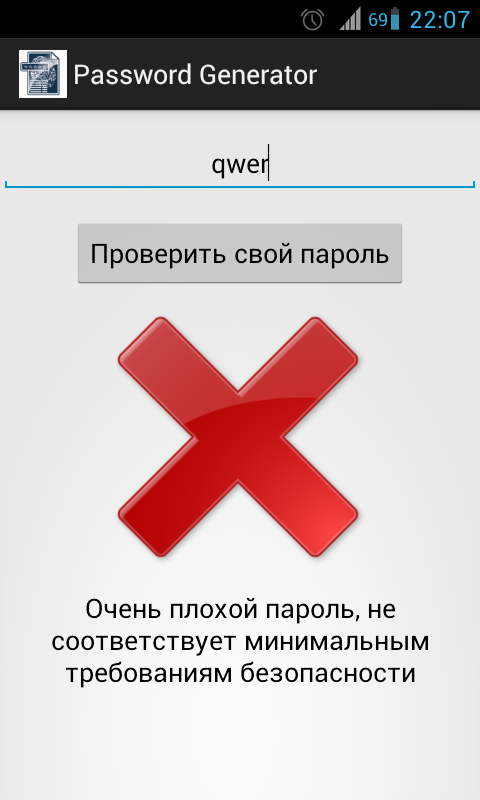
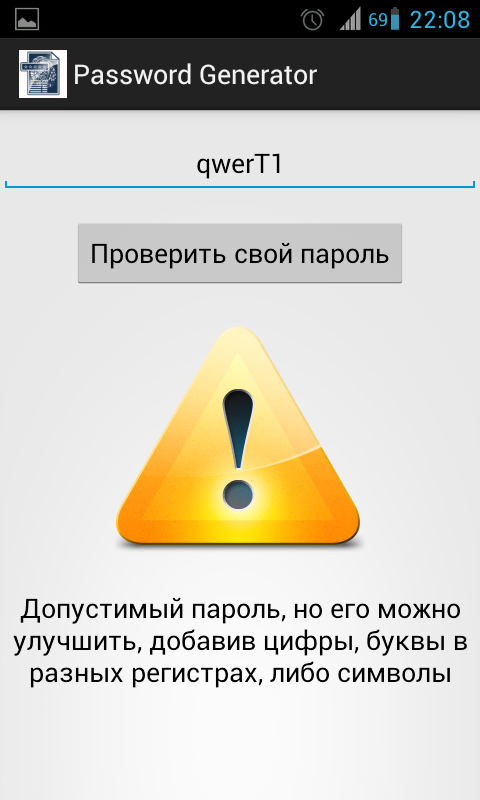
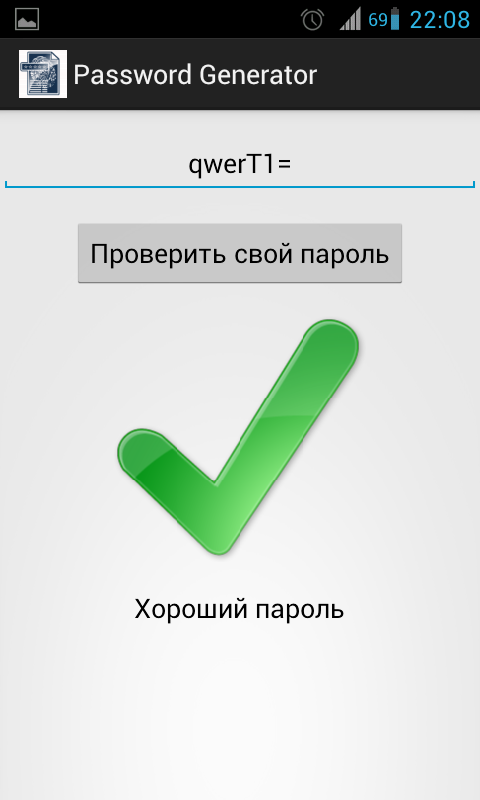
  

Рисунок 5.3 Оценка пароля

Для генерации нового пароля необходимо выбрать режим «Создать новый пароль» и указать параметры нового пароля (алфавит, количество символов), пароль будет сгенерирован автоматически, получить следующий вариант пароля с текущими настройками можно воспользовавшись кнопкой «Следующий пароль» (рис. 5.4)

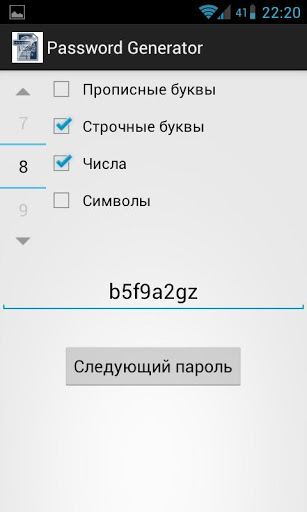


Рисунок 5.4 Режим генерации пароля

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта были на практике закреплены теоретические знания, полученные на протяжении всего периода обучения. Кроме того, в процессе написания дипломного проекта осуществлено самостоятельное изучение языка Java и системы контроля версий Git, а также инструментов для разработки Android приложений, таких как Eclipse Juno, Android Software Development Kit и Java Development Kit.

Программный продукт будет распространяться бесплатно, что должно способствовать быстрому росту популярности приложения среди пользователей. В дополнение к этому, исходные коды продукта доступны на условиях лицензии GNU General Public License. Этот факт позволит сторонним разработчикам, как улучшить продукт, так и адаптировать его под свои нужды.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 7.32-2001. Отчет о научно-исследовательской работе. Структура и правила оформления
2. ГОСТ 19.001-77 ЕСПД. Общие положения
3. ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов
4. ГОСТ 19.102-77 ЕСПД. Стадии разработки
5. ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению
6. ГОСТ 19.301-79 ЕСПД. Программа и методика испытаний. Требования к содержанию, оформлению и контролю качества
7. ГОСТ 19.402-2000 ЕСПД. Описание программы. Требования к содержанию, оформлению и контролю качества
8. ГОСТ 19.404-79 ЕСПД. Пояснительная записка. Требования к содержанию и оформлению
9. ГОСТ 19.502-78 ЕСПД. Описание применения. Требования к содержанию и оформлению
10. ГОСТ 19.504-79 ЕСПД. Руководство программиста
11. ГОСТ 19.505-79 ЕСПД. Руководство оператора
12. ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения
13. Android Notifications. Оповещения через Status Bar - http://habrahabr.ru/post/140928/
14. Многопоточность в Android. Часть 1: AsyncTask, асинхронная загрузка изображений из интернет - http://idev.by/android/468/
15. Разработка под Android - http://developer.alexanderklimov.ru/android/
16. Сервисы - это просто - http://blog.divenvrsk.org/2010/09/android.html
17. Темы и стили - http://developer.alexanderklimov.ru/android/theme.php
18. Учебник по Android. Уроки для начинающих - http://startandroid.ru/uroki/vse-uroki-spiskom.html
19. Хабраиндекс для статей по программированию под Android -http://habrahabr.ru/post/111603/
20. The Developer's Guide - http://developer.android.com
21. Parsing and traversing a Document - http://jsoup.org/cookbook/introduction/parsing-a-document
22. Usage - http://actionbarsherlock.com/usage.html
23. Голощапов А. Л. Google Android: Программирование для мобильных устройств.– СПб.: БХВ-Петербург, 2011. – 448 с.
24. Дэрси Л., Кондер Ш. Android за 24 часа. Программирование приложений под операционную систему Google. — М.: Рид Групп, 2011. – 464 с.
25. Эккель Б. Философия Java. Библиотека программиста. 4-е изд. – СПб.: Питер, 2009. – 640 с.
26. Маер Р. Android 2. Программирование приложений для планшетных компьютеров и смартфонов. – М.: Эксмо, 2011. – 672 с.
27. Хашими С., Коматинени С. и др. Разработка приложений для Android. – Спб.: Питер, 2011. – 736 с.

# ПРИЛОЖЕНИЕ А ТЕКСТ ПРОГРАММЫ

package by.bsuir.passgen;

import by.bsuir.passgen.utilities.Constants;

public class Checker {

private static final String START = ".\*[";

private static final String END = "]+.\*";

public static int getPasswordRating(String password) {

if (password.length() < Constants.MIN\_PWD\_LEN) {

return -1;

}

int passwordRating = 0;

if (password.matches(START + Constants.LOWER\_CASE + END)) {

passwordRating++;

}

if (password.matches(START + Constants.UPPER\_CASE + END)) {

passwordRating++;

}

if (password.matches(START + Constants.NUMBERS + END)) {

passwordRating++;

}

if (password.matches(START + "/!/№/;/%/:/?/\*/(/)/\_/+/=/-/~/\\/</>/,/./[/]/{/}" + END)) {

passwordRating++;

}

if (passwordRating <= 3) {

return 0;

} else {

return 1;

}

}

}

package by.bsuir.passgen;

import java.util.Random;

import by.bsuir.passgen.utilities.Alphabet;

import by.bsuir.passgen.utilities.Constants;

import android.content.SharedPreferences;

public class Generator {

public static String generatePassword(int passwordLen, SharedPreferences preferences) {

StringBuilder alphabet = new StringBuilder();

if (preferences.getBoolean(Alphabet.LOWER\_CASE.toString(), false)) {

alphabet.append(Constants.LOWER\_CASE);

}

if (preferences.getBoolean(Alphabet.UPPER\_CASE.toString(), false)) {

alphabet.append(Constants.UPPER\_CASE);

}

if (preferences.getBoolean(Alphabet.NUMBERS.toString(), false)) {

alphabet.append(Constants.NUMBERS);

}

if (preferences.getBoolean(Alphabet.SYMBOLS.toString(), false)) {

alphabet.append(Constants.SYMBOLS);

}

StringBuilder password = new StringBuilder();

if (alphabet.length() > 0) {

int alphabetLen = alphabet.length();

Random generator = new Random();

for (int i = 0; i < passwordLen; i++) {

password.append(alphabet.charAt(generator.nextInt(alphabetLen)));

}

}

return password.toString();

}

}

package by.bsuir.passgen;

import by.bsuir.passgen.utilities.Tool;

import android.app.AlertDialog;

import android.app.Dialog;

import android.app.DialogFragment;

import android.content.Intent;

import android.view.View.OnClickListener;

import android.net.Uri;

import android.os.Bundle;

import android.view.View;

import android.widget.LinearLayout;

import android.widget.TextView;

public class AboutDialog extends DialogFragment implements OnClickListener {

public Dialog onCreateDialog(Bundle savedInstanceState) {

AlertDialog.Builder dialog = new AlertDialog.Builder(getActivity());

dialog.setIcon(android.R.drawable.ic\_dialog\_info);

dialog.setTitle(getResources().getString(R.string.menu\_about));

LinearLayout viewDialog = (LinearLayout) getActivity().getLayoutInflater().inflate(R.layout.about\_layout, null);

TextView version=(TextView) viewDialog.findViewById(R.id.version);

version.setText(version.getText() + " " + Tool.getVersion(getActivity()));

TextView developer=(TextView) viewDialog.findViewById(R.id.developer);

developer.setOnClickListener(this);

dialog.setView(viewDialog);

return dialog.create();

}

public void onClick(View v) {

Intent browserIntent = new Intent(Intent.ACTION\_VIEW, Uri.parse(getString(R.string.developer\_page)));

startActivity(browserIntent);

}

}

package by.bsuir.passgen.activities;

import by.bsuir.passgen.AboutDialog;

import by.bsuir.passgen.R;

import android.app.Activity;

import android.app.DialogFragment;

import android.content.Intent;

import android.os.Bundle;

import android.view.KeyEvent;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

public class DashboardActivity extends Activity implements OnClickListener {

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.dashboard\_layout);

Button newPassword = (Button) findViewById(R.id.new\_password);

newPassword.setOnClickListener(this);

Button checkPassword = (Button) findViewById(R.id.check\_password);

checkPassword.setOnClickListener(this);

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.activity\_main, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case R.id.menu\_about:

DialogFragment about = new AboutDialog();

about.show(getFragmentManager(), "about");

return true;

default:

return super.onOptionsItemSelected(item);

}

}

@Override

public boolean onKeyDown(int keyCode, KeyEvent event) {

if (keyCode == KeyEvent.KEYCODE\_BACK) {

moveTaskToBack(true);

return true;

}

return super.onKeyDown(keyCode, event);

}

public void onClick(View v) {

switch (v.getId()) {

case R.id.new\_password:

startActivity(new Intent(this, NewPasswordActivity.class));

break;

case R.id.check\_password:

startActivity(new Intent(this, CheckPasswordActivity.class));

break;

}

}

}

package by.bsuir.passgen.activities;

import by.bsuir.passgen.AboutDialog;

import by.bsuir.passgen.Generator;

import by.bsuir.passgen.R;

import by.bsuir.passgen.utilities.Alphabet;

import by.bsuir.passgen.utilities.Constants;

import android.os.Bundle;

import android.preference.PreferenceManager;

import android.app.Activity;

import android.app.DialogFragment;

import android.content.SharedPreferences;

import android.content.SharedPreferences.Editor;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.CheckBox;

import android.widget.EditText;

import android.widget.NumberPicker;

import android.widget.NumberPicker.OnValueChangeListener;

public class NewPasswordActivity extends Activity implements OnValueChangeListener {

private EditText password;

private NumberPicker passwordLen;

private SharedPreferences preferences;

private CheckBox upperCase, lowerCase, numbers, symbols;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.new\_password\_layout);

password = (EditText) findViewById(R.id.password);

passwordLen = (NumberPicker) findViewById(R.id.password\_len);

passwordLen.setMinValue(Constants.MIN\_PWD\_LEN);

passwordLen.setMaxValue(Constants.MAX\_PWD\_LEN);

passwordLen.setOnValueChangedListener(this);

upperCase = (CheckBox) findViewById(R.id.upper\_case);

lowerCase = (CheckBox) findViewById(R.id.lower\_case);

numbers = (CheckBox) findViewById(R.id.numbers);

symbols = (CheckBox) findViewById(R.id.symbols);

}

private void restorePreferences() {

upperCase.setChecked(preferences.getBoolean(Alphabet.UPPER\_CASE.toString(), false));

lowerCase.setChecked(preferences.getBoolean(Alphabet.LOWER\_CASE.toString(), false));

numbers.setChecked(preferences.getBoolean(Alphabet.NUMBERS.toString(), false));

symbols.setChecked(preferences.getBoolean(Alphabet.SYMBOLS.toString(), false));

passwordLen.setValue(preferences.getInt(Constants.PWD\_LEN, Constants.MIN\_PWD\_LEN));

next(new View(getApplicationContext()));

}

@Override

public void onResume() {

super.onResume();

preferences = PreferenceManager.getDefaultSharedPreferences(getApplicationContext());

restorePreferences();

}

public void onCheck(View view) {

CheckBox checkBox = (CheckBox) view;

Editor editor = preferences.edit();

switch (checkBox.getId()) {

case R.id.upper\_case:

editor.putBoolean(Alphabet.UPPER\_CASE.toString(), checkBox.isChecked()).commit();

break;

case R.id.lower\_case:

editor.putBoolean(Alphabet.LOWER\_CASE.toString(), checkBox.isChecked()).commit();

break;

case R.id.numbers:

editor.putBoolean(Alphabet.NUMBERS.toString(), checkBox.isChecked()).commit();

break;

case R.id.symbols:

editor.putBoolean(Alphabet.SYMBOLS.toString(), checkBox.isChecked()).commit();

break;

}

password.setText(Generator.generatePassword(passwordLen.getValue(), preferences));

}

public void onValueChange(NumberPicker arg0, int arg1, int arg2) {

Editor editor = preferences.edit();

editor.putInt(Constants.PWD\_LEN, passwordLen.getValue()).commit();

password.setText(Generator.generatePassword(arg2, preferences));

}

public void next(View view) {

password.setText(Generator.generatePassword(passwordLen.getValue(), preferences));

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.activity\_main, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case R.id.menu\_about:

DialogFragment about = new AboutDialog();

about.show(getFragmentManager(), "about");

return true;

default:

return super.onOptionsItemSelected(item);

}

}

}

package by.bsuir.passgen.activities;

import by.bsuir.passgen.AboutDialog;

import by.bsuir.passgen.Checker;

import by.bsuir.passgen.R;

import android.app.Activity;

import android.app.DialogFragment;

import android.os.Bundle;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.TextView;

public class CheckPasswordActivity extends Activity {

private EditText userPassword;

private ImageView verdictLogo;

private TextView verdictComment;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.check\_password\_layout);

userPassword = (EditText) findViewById(R.id.user\_password);

verdictLogo = (ImageView) findViewById(R.id.verdict);

verdictComment = (TextView) findViewById(R.id.verdict\_comment);

}

public void onCheckPassword(View view) {

int verdict = Checker.getPasswordRating(userPassword.getText().toString());

switch (verdict) {

case -1:

verdictLogo.setImageDrawable(getResources().getDrawable(R.drawable.ic\_bad\_password));

verdictComment.setText(getResources().getString(R.string.bad\_password));

break;

case 0:

verdictLogo.setImageDrawable(getResources().getDrawable(R.drawable.ic\_average\_password));

verdictComment.setText(getResources().getString(R.string.average\_password));

break;

case 1:

verdictLogo.setImageDrawable(getResources().getDrawable(R.drawable.ic\_good\_password));

verdictComment.setText(getResources().getString(R.string.good\_password));

break;

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.activity\_main, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case R.id.menu\_about:

DialogFragment about = new AboutDialog();

about.show(getFragmentManager(), "about");

return true;

default:

return super.onOptionsItemSelected(item);

}

}

}