Ministerul Educaţiei al Republicii Moldova

Universitatea Tehnică a Moldovei

Catedra: Automatica și Tehnologii Informaționale

RAPORT

Lucrare de laborator
Nr.6 *la M.I.D.P.S.*

A efectuat: st. gr. TI-143
Cheptanaru Anatolie

A verificat: lector.univ.
Cojocaru Svetlana

Chişinău 2016

LUCRAREA DE LABORATOR 6

Obiectivele lucrării

- Crearea unei aplicatii complexe in echipa.
- Divizarea sarcinilor pe membrii echipei

1. Laboratory work Requirements:

- Lucreaza la proiect in echipa de 2-3 persoane
- Divizeaza task-urile si descrie-le in raport, indicind pentru fiecare cine este responsabil pentru el.
- Inainte de a trece la dezvoltarea proiectului, creeaza o schema cit mai apropiata de rezultatul final *(schema trebuie sa fie primul commit)*
- Fiecare din membrii echipei va lucra pe propriul branch in git, iar una din persoane va avea grija sa faca merge cu master.
- Proiectul se poate afla doar in repozitoriul unui membru al echipei.
- Fiecare din membru va avea propriul raport care va include propriile observatii si concluzii.
  - *Basic Level* (nota 5 || 6):
    -
  - *Normal Level* (nota 7 || 8):
    -
  - *Advanced Level* (nota 9 || 10):
    - Dezvoltarea unei aplicatii:
      - Desktop
      - Mobile
      - Web
      - Browser Extension
      - Game development (web, mobile, desktop)
      - Service application
      - Internet application
      - Client application

Codul sursa:

```
package air.spidergame;

import air.com.adobe.appentry.R;
import android.R.id;
import android.app.Activity;
import android.app.AlertDialog.Builder;
import android.app.Dialog;
import android.content.ActivityNotFoundException;
import android.content.ComponentName;
import android.content.Context;
import android.content.DialogInterface;
import android.content.DialogInterface.OnCancelListener;
import android.content.DialogInterface.OnClickListener;
import android.content.Intent;
import android.content.ServiceConnection;
import android.content.pm.PackageManager.NameNotFoundException;
import android.content.res.Configuration;
import android.content.res.Resources.Theme;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.os.Bundle;
import android.os.IBinder;
import android.os.Process;
import android.util.AttributeSet;
import android.view.ContextMenu;
import android.view.ContextMenu.ContextMenuInfo;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.MenuItem;
import android.view.MotionEvent;
import android.view.View;
import android.view.WindowManager.LayoutParams;
import dalvik.system.DexClassLoader;
import java.lang.reflect.Method;
import java.net.URISyntaxException;

public class AppEntry extends Activity {
    private static final String LOG_TAG = "AppEntry";
    private static String RUNTIME_PACKAGE_ID;
    private static Object sAndroidActivityWrapper;
    private static Class<?> sAndroidActivityWrapperClass;
    private static boolean sDexLoaded;
    private static DexClassLoader sDloader;

    /* renamed from: air.spidergame.AppEntry.1 */
    class C00001 implements OnClickListener {
        C00001() {
        }

        public void onClick(DialogInterface dialog, int which) {
            AppEntry.this.launchMarketPlace(AppEntry.RUNTIME_PACKAGE_ID);
            System.exit(0);
        }
    }
```

```
/* renamed from: air.spidergame.AppEntry.2 */
class C00012 implements OnClickListener {
    C00012() {
    }

    public void onClick(DialogInterface dialog, int which) {
        System.exit(0);
    }
}

/* renamed from: air.spidergame.AppEntry.3 */
class C00023 implements OnCancelListener {
    C00023() {
    }

    public void onCancel(DialogInterface dialog) {
        System.exit(0);
    }
}

/* renamed from: air.spidergame.AppEntry.4 */
class C00034 implements ServiceConnection {
    public void onServiceConnected(ComponentName name, IBinder service) {
        AppEntry.this.unbindService(this);
        AppEntry.this.loadDexAndCreateActivityWrapper();
        if (AppEntry.sDexLoaded) {
            AppEntry.this.InvokeWrapperOnCreate();
        } else {
            AppEntry.KillSelf();
        }
    }

    C00034() {
    }

    public void onServiceDisconnected(ComponentName name) {
    }
}

static {
    sDexLoaded = false;
    sAndroidActivityWrapper = null;
    RUNTIME_PACKAGE_ID = "com.adobe.air";
}

public void BroadcastIntent(String action, String data) {
    try {
        startActivity(Intent.parseUri(data, 0).setAction(action).addFlags(268435456));
    } catch (URISyntaxException e) {
    } catch (ActivityNotFoundException e2) {
    }
}

public void launchMarketPlace(String packageName) {
    try {
        BroadcastIntent("android.intent.action.VIEW", "market://details?id=" + packageName);
        } catch (Exception e) {
        }
    }

    public boolean isRuntimeInstalled() {
        try {
            getPackageManager().getPackageInfo(RUNTIME_PACKAGE_ID, 256);
            return true;
        } catch (NameNotFoundException e) {
            NameNotFoundException nfe = e;
            return false;
        }
    }

    public void showDialog() {
        Builder alertDialogBuilder = new Builder(this);
        alertDialogBuilder.setTitle(R.string.dialog_title);
        alertDialogBuilder.setMessage(R.string.dialog_text);
        alertDialogBuilder.setPositiveButton(R.string.button_yes, new C00001());
        alertDialogBuilder.setNegativeButton(R.string.button_no, new C00012());
        alertDialogBuilder.setOnCancelListener(new C00023());
        alertDialogBuilder.show();
    }

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (isRuntimeInstalled()) {
            loadDexAndCreateActivityWrapper();
            if (sDexLoaded) {
                InvokeWrapperOnCreate();
                return;
            } else {
                launchAIRService();
                return;
            }
        }
        showDialog();
    }

    private void launchAIRService() {
        try {
            Intent intent = new Intent("com.adobe.air.AIRServiceAction");
            intent.setClassName(RUNTIME_PACKAGE_ID, "com.adobe.air.AIRService");
            bindService(intent, new C00034(), 1);
        } catch (Exception e) {
        }
    }

    private void InvokeWrapperOnCreate() {
        try {
            Method method = sAndroidActivityWrapperClass.getMethod("onCreate", new
Class[]{Activity.class, String[].class});
            Boolean isADL = new Boolean(false);
            Boolean isDebuggerMode = new Boolean(false);
            String[] args = new String[]{"", "", "-nodebug", isADL.toString(),
isDebuggerMode.toString()};
```

```
            InvokeMethod(method, this, args);
        } catch (Exception e) {
        }
    }

    private Object InvokeMethod(Method method, Object... args) {
        if (!sDexLoaded) {
            return null;
        }
        Object retval = null;
        if (args != null) {
            try {
                retval = method.invoke(sAndroidActivityWrapper, args);
            } catch (Exception e) {
            }
        } else {
            retval = method.invoke(sAndroidActivityWrapper, new Object[0]);
        }
        return retval;
    }

    private static void KillSelf() {
        Process.killProcess(Process.myPid());
    }

    public void onStart() {
        super.onStart();
    }

    public void onRestart() {
        super.onRestart();
        try {
            if (sDexLoaded) {
                InvokeMethod(sAndroidActivityWrapperClass.getMethod("onRestart", new Class[0]),
new Object[0]);
            }
        } catch (Exception e) {
        }
    }

    public void onPause() {
        super.onPause();
        try {
            if (sDexLoaded) {
                InvokeMethod(sAndroidActivityWrapperClass.getMethod("onPause", new Class[0]), new
Object[0]);
            }
        } catch (Exception e) {
        }
    }

    public void onResume() {
        super.onResume();
        try {
            if (sDexLoaded) {
                InvokeMethod(sAndroidActivityWrapperClass.getMethod("onResume", new Class[0]),
new Object[0]);
            }
        } catch (Exception e) {
        }
    }

    public void onStop() {
        super.onStop();
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onStop", new Class[0]), new
Object[0]);
        } catch (Exception e) {
        }
    }

    public void onDestroy() {
        super.onDestroy();
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onDestroy", new Class[0]), new
Object[0]);
        } catch (Exception e) {
        }
    }

    public void onConfigurationChanged(Configuration newConfig) {
        super.onConfigurationChanged(newConfig);
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onConfigurationChanged", new
Class[]{Configuration.class}), newConfig);
        } catch (Exception e) {
        }
    }

    public void onLowMemory() {
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onLowMemory", new Class[0]),
new Object[0]);
        } catch (Exception e) {
        }
    }

    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        try {
            if (sDexLoaded) {
                InvokeMethod(sAndroidActivityWrapperClass.getMethod("onActivityResult", new
Class[]{Integer.TYPE, Integer.TYPE, Intent.class}), Integer.valueOf(requestCode),
Integer.valueOf(resultCode), data);
            }
        } catch (Exception e) {
        }
    }

    protected void onNewIntent(Intent aIntent) {
        super.onNewIntent(aIntent);
        try {
```

```
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onNewIntent", new
Class[]{Intent.class}), ii);
    } catch (Exception e) {
    }
  }

  private void loadDexAndCreateActivityWrapper() {
    try {
      if (!sDexLoaded) {
        sDloader = new DexClassLoader(RUNTIME_PACKAGE_ID,
getFilesDir().getAbsolutePath(), null, createPackageContext(RUNTIME_PACKAGE_ID,
3).getClassLoader());
        sAndroidActivityWrapperClass =
sDloader.loadClass("com.adobe.air.AndroidActivityWrapper");
        if (sAndroidActivityWrapperClass != null) {
          sDexLoaded = true;
        }
      }
      sAndroidActivityWrapper =
sAndroidActivityWrapperClass.getMethod("CreateAndroidActivityWrapper", new
Class[]{Activity.class}).invoke(null, new Object[]{this});
    } catch (Exception e) {
    }
  }

  public void finishActivityFromChild(Activity child, int requestCode) {
    super.finishActivityFromChild(child, requestCode);
    try {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("finishActivityFromChild", new
Class[]{Activity.class, Integer.TYPE}), child, Integer.valueOf(requestCode));
    } catch (Exception e) {
    }
  }

  public void finishFromChild(Activity child) {
    super.finishFromChild(child);
    try {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("finishFromChild", new
Class[]{Activity.class}), child);
    } catch (Exception e) {
    }
  }

  public void onAttachedToWindow() {
    super.onAttachedToWindow();
    try {
      InvokeMethod(sAndroidActivityWrapperClass.getMethod("onAttachedToWindow", new
Class[0]), new Object[0]);
    } catch (Exception e) {
    }
  }

  public void onBackPressed() {
    super.onBackPressed();
    try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onBackPressed", new Class[0]),
new Object[0]);
        } catch (Exception e) {
        }
    }

    public void onContentChanged() {
        super.onContentChanged();
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onContentChanged", new
Class[0]), new Object[0]);
        } catch (Exception e) {
        }
    }

    public boolean onContextItemSelected(MenuItem item) {
        boolean retval = super.onContextItemSelected(item);
        try {
            return ((Boolean)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onContextItemSelected", new
Class[]{MenuItem.class, Boolean.TYPE}), item, Boolean.valueOf(retval))).booleanValue();
        } catch (Exception e) {
            return retval;
        }
    }

    public void onContextMenuClosed(Menu menu) {
        super.onContextMenuClosed(menu);
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onContextMenuClosed", new
Class[]{Menu.class}), menu);
        } catch (Exception e) {
        }
    }

    public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo menuInfo) {
        super.onCreateContextMenu(menu, v, menuInfo);
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onCreateContextMenu", new
Class[]{ContextMenu.class, View.class, ContextMenuInfo.class}), menu, v, menuInfo);
        } catch (Exception e) {
        }
    }

    public CharSequence onCreateDescription() {
        CharSequence retval = super.onCreateDescription();
        try {
            return (CharSequence)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onCreateDescription", new
Class[]{CharSequence.class}), retval);
        } catch (Exception e) {
            return retval;
        }
    }

    public boolean onCreateOptionsMenu(Menu menu) {
        boolean retval = super.onCreateOptionsMenu(menu);
        try {
            return ((Boolean)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onCreateOptionsMenu", new
Class[]{Menu.class, Boolean.TYPE}), menu, Boolean.valueOf(retval))).booleanValue();
        } catch (Exception e) {
            return retval;
        }
    }

    public boolean onCreatePanelMenu(int featureId, Menu menu) {
        boolean retval = super.onCreatePanelMenu(featureId, menu);
        try {
            return ((Boolean)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onCreatePanelMenu", new
Class[]{Integer.TYPE, Menu.class, Boolean.TYPE}), Integer.valueOf(featureId), menu,
Boolean.valueOf(retval))).booleanValue();
        } catch (Exception e) {
            return retval;
        }
    }

    public View onCreatePanelView(int featureId) {
        View retval = super.onCreatePanelView(featureId);
        try {
            return (View)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onCreatePanelView", new
Class[]{Integer.TYPE, View.class}), Integer.valueOf(featureId), retval);
        } catch (Exception e) {
            return retval;
        }
    }

    public boolean onCreateThumbnail(Bitmap outBitmap, Canvas canvas) {
        boolean retval = super.onCreateThumbnail(outBitmap, canvas);
        try {
            return ((Boolean)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onCreateThumbnail", new
Class[]{Bitmap.class, Canvas.class, Boolean.TYPE}), outBitmap, canvas,
Boolean.valueOf(retval))).booleanValue();
        } catch (Exception e) {
            return retval;
        }
    }

    public View onCreateView(String name, Context context, AttributeSet attrs) {
        View retval = super.onCreateView(name, context, attrs);
        try {
            return (View) InvokeMethod(sAndroidActivityWrapperClass.getMethod("onCreateView",
new Class[]{String.class, Context.class, AttributeSet.class, View.class}), name, context, attrs,
retval);
        } catch (Exception e) {
            return retval;
        }
    }
    public void onDetachedFromWindow() {
        super.onDetachedFromWindow();
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onDetachedFromWindow", new
Class[0]), new Object[0]);
        } catch (Exception e) {
        }
    }

    public boolean onKeyDown(int keyCode, KeyEvent event) {
        boolean retval = super.onKeyDown(keyCode, event);
        try {
            return ((Boolean) InvokeMethod(sAndroidActivityWrapperClass.getMethod("onKeyDown",
new Class[]{Integer.TYPE, KeyEvent.class, Boolean.TYPE}), Integer.valueOf(keyCode), event,
Boolean.valueOf(retval))).booleanValue();
        } catch (Exception e) {
            return retval;
        }
    }

    public boolean onKeyLongPress(int keyCode, KeyEvent event) {
        boolean retval = super.onKeyLongPress(keyCode, event);
        try {
            return ((Boolean)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onKeyLongPress", new
Class[]{Integer.TYPE, KeyEvent.class, Boolean.TYPE}), Integer.valueOf(keyCode), event,
Boolean.valueOf(retval))).booleanValue();
        } catch (Exception e) {
            return retval;
        }
    }

    public boolean onKeyMultiple(int keyCode, int repeatCount, KeyEvent event) {
        boolean retval = super.onKeyMultiple(keyCode, repeatCount, event);
        try {
            return ((Boolean)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onKeyMultiple", new
Class[]{Integer.TYPE, Integer.TYPE, KeyEvent.class, Boolean.TYPE}), Integer.valueOf(keyCode),
Integer.valueOf(repeatCount), event, Boolean.valueOf(retval))).booleanValue();
        } catch (Exception e) {
            return retval;
        }
    }

    public boolean onKeyUp(int keyCode, KeyEvent event) {
        boolean retval = super.onKeyUp(keyCode, event);
        try {
            return ((Boolean) InvokeMethod(sAndroidActivityWrapperClass.getMethod("onKeyUp",
new Class[]{Integer.TYPE, KeyEvent.class, Boolean.TYPE}), Integer.valueOf(keyCode), event,
Boolean.valueOf(retval))).booleanValue();
        } catch (Exception e) {
            return retval;
        }
    }

    public boolean onMenuItemSelected(int featureId, MenuItem item) {
        boolean retval = super.onMenuItemSelected(featureId, item);
        try {
            return ((Boolean)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onMenuItemSelected", new
Class[]{Integer.TYPE, MenuItem.class, Boolean.TYPE}), Integer.valueOf(featureId), item,
Boolean.valueOf(retval))).booleanValue();
        } catch (Exception e) {
            return retval;
        }
    }

    public boolean onMenuOpened(int featureId, Menu menu) {
        boolean retval = super.onMenuOpened(featureId, menu);
        try {
            return ((Boolean)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onMenuOpened", new
Class[]{Integer.TYPE, Menu.class, Boolean.TYPE}), Integer.valueOf(featureId), menu,
Boolean.valueOf(retval))).booleanValue();
        } catch (Exception e) {
            return retval;
        }
    }

    public boolean onOptionsItemSelected(MenuItem item) {
        boolean retval = super.onOptionsItemSelected(item);
        try {
            return ((Boolean)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onOptionsItemSelected", new
Class[]{MenuItem.class, Boolean.TYPE}), item, Boolean.valueOf(retval))).booleanValue();
        } catch (Exception e) {
            return retval;
        }
    }

    public void onOptionsMenuClosed(Menu menu) {
        super.onOptionsMenuClosed(menu);
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onOptionsMenuClosed", new
Class[]{Menu.class}), menu);
        } catch (Exception e) {
        }
    }

    public void onPanelClosed(int featureId, Menu menu) {
        super.onPanelClosed(featureId, menu);
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onPanelClosed", new
Class[]{Integer.TYPE, Menu.class}), Integer.valueOf(featureId), menu);
        } catch (Exception e) {
        }
    }

    public boolean onPrepareOptionsMenu(Menu menu) {
        boolean retval = super.onPrepareOptionsMenu(menu);
        try {
            return ((Boolean)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onPrepareOptionsMenu", new
Class[]{Menu.class, Boolean.TYPE}), menu, Boolean.valueOf(retval))).booleanValue();
        } catch (Exception e) {
            return retval;
        }
    }

    public boolean onPreparePanel(int featureId, View view, Menu menu) {
        boolean retval = super.onPreparePanel(featureId, view, menu);
        try {
            return ((Boolean)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onPreparePanel", new
Class[]{Integer.TYPE, View.class, Menu.class, Boolean.TYPE}), Integer.valueOf(featureId), view,
menu, Boolean.valueOf(retval))).booleanValue();
        } catch (Exception e) {
            return retval;
        }
    }

    public Object onRetainNonConfigurationInstance() {
        Object retval = super.onRetainNonConfigurationInstance();
        try {
            return
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onRetainNonConfigurationInstance",
new Class[]{Object.class}), retval);
        } catch (Exception e) {
            return retval;
        }
    }

    public boolean onSearchRequested() {
        boolean retval = super.onSearchRequested();
        try {
            return ((Boolean)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onSearchRequested", new
Class[]{Boolean.TYPE}), Boolean.valueOf(retval))).booleanValue();
        } catch (Exception e) {
            return retval;
        }
    }

    public boolean onTouchEvent(MotionEvent event) {
        boolean retval = super.onTouchEvent(event);
        try {
            return ((Boolean)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onTouchEvent", new
Class[]{MotionEvent.class, Boolean.TYPE}), event, Boolean.valueOf(retval))).booleanValue();
        } catch (Exception e) {
            return retval;
        }
    }

    public boolean onTrackballEvent(MotionEvent event) {
        boolean retval = super.onTrackballEvent(event);
        try {
            return ((Boolean)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onTrackballEvent", new
Class[]{MotionEvent.class, Boolean.TYPE}), event, Boolean.valueOf(retval))).booleanValue();
        } catch (Exception e) {
            return retval;
        }
    }

    public void onUserInteraction() {
        super.onUserInteraction();
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onUserInteraction", new
Class[0]), new Object[0]);
        } catch (Exception e) {
        }
    }

    public void onWindowAttributesChanged(LayoutParams params) {
        super.onWindowAttributesChanged(params);
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onWindowAttributesChanged",
new Class[]{LayoutParams.class}), params);
        } catch (Exception e) {
        }
    }

    public void onWindowFocusChanged(boolean hasFocus) {
        super.onWindowFocusChanged(hasFocus);
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onWindowFocusChanged", new
Class[]{Boolean.TYPE}), Boolean.valueOf(hasFocus));
        } catch (Exception e) {
        }
    }

    protected void onApplyThemeResource(Theme theme, int resid, boolean first) {
        super.onApplyThemeResource(theme, resid, first);
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onApplyThemeResource", new
Class[]{Theme.class, Integer.TYPE, Boolean.TYPE}), theme, Integer.valueOf(resid),
Boolean.valueOf(first));
        } catch (Exception e) {
        }
    }

    protected void onChildTitleChanged(Activity childActivity, CharSequence title) {
        super.onChildTitleChanged(childActivity, title);
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onChildTitleChanged", new
Class[]{Activity.class, CharSequence.class}), childActivity, title);
        } catch (Exception e) {
        }
    }

    protected Dialog onCreateDialog(int id) {
        Dialog retval = super.onCreateDialog(id);
        try {
            return (Dialog)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onCreateDialog", new
Class[]{Integer.TYPE, Dialog.class}), Integer.valueOf(id), retval);
        } catch (Exception e) {
            return retval;
        }
    }

    protected Dialog onCreateDialog(int id, Bundle args) {
        Dialog retval = super.onCreateDialog(id, args);
        try {
            return (Dialog)
InvokeMethod(sAndroidActivityWrapperClass.getMethod("onCreateDialog", new
Class[]{Integer.TYPE, Bundle.class, Dialog.class}), Integer.valueOf(id), args, retval);
        } catch (Exception e) {
            return retval;
        }
    }

    protected void onPostCreate(Bundle savedInstanceState) {
        super.onPostCreate(savedInstanceState);
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onPostCreate", new
Class[]{Bundle.class}), savedInstanceState);
        } catch (Exception e) {
        }
    }

    protected void onPostResume() {
        super.onPostResume();
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onPostResume", new Class[0]),
new Object[0]);
        } catch (Exception e) {
        }
    }

    protected void onPrepareDialog(int id, Dialog dialog) {
        super.onPrepareDialog(id, dialog);
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onPrepareDialog", new
Class[]{id.class, Dialog.class}), Integer.valueOf(id), dialog);
        } catch (Exception e) {
        }
    }

    protected void onPrepareDialog(int id, Dialog dialog, Bundle args) {
        super.onPrepareDialog(id, dialog, args);
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onPrepareDialog", new
Class[]{id.class, Dialog.class, Bundle.class}), Integer.valueOf(id), dialog, args);
        } catch (Exception e) {
        }
    }
    protected void onRestoreInstanceState(Bundle savedInstanceState) {
        super.onRestoreInstanceState(savedInstanceState);
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onRestoreInstanceState", new
Class[]{Bundle.class}), savedInstanceState);
        } catch (Exception e) {
        }
    }

    protected void onSaveInstanceState(Bundle outState) {
        super.onSaveInstanceState(outState);
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onSaveInstanceState", new
Class[]{Bundle.class}), outState);
        } catch (Exception e) {
        }
    }

    protected void onTitleChanged(CharSequence title, int color) {
        super.onTitleChanged(title, color);
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onTitleChanged", new
Class[]{CharSequence.class, Integer.TYPE}), title, Integer.valueOf(color));
        } catch (Exception e) {
        }
    }

    protected void onUserLeaveHint() {
        super.onUserLeaveHint();
        try {
            InvokeMethod(sAndroidActivityWrapperClass.getMethod("onUserLeaveHint", new
Class[0]), new Object[0]);
        } catch (Exception e) {
        }
    }
}
```

Concluzie

Pe parcursul efectuarii lucrarii de laborator am obtinut cunoștințele si abilitățile de crearea a unei aplicatii pentru Android. Pentru crearea unei aplicatii trebuie sa fie cunoscute mai multe lucruri in special limbajul de programare de nivel inalt JavaScirpt care este foarte des folosit pentru crearea aplicatiilor .Există de asemenea și alte limbaje de creare a unei aplicatii spre exemplu Python.