

VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE

FAKULTA INFORMATIKY A STATISTIKY

KATEDRA INFORMAČNÍCH TECHNOLOGIÍ

STUDIJNÍ PROGRAM: APLIKOVANÁ INFORMATIKA

OBOR: INFORMAČNÍ SYSTÉMY A TECHNOLOGIE

Prototyp aplikace pro ověření identity studenta u písemné zkoušky

DIPLOMOVÁ PRÁCE

Student: Bc. Ondřej Schrek
Vedoucí: Ing. Libor Gála, Ph.D.

2019

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze kterých jsem čerpal

V Praze dne 2. listopadu 2018

.....
Bc. Ondřej Schrek

Poděkování

Text poděkování.

Abstrakt

Doplnit text.

Klíčová slova

Doplnit klíčová slova.

Abstract

Doplnit anj text.

Keywords

Doplnit anj ključová slova.

Obsah

Úvod	1
1 Charakteristika problému	2
1.1 Současný stav systému	2
1.2 Problémy stávajícího systému	2
1.3 Souhrn problémů a podmínek řešení	2
1.4 Návrh řešení	2
1.5 Cíl práce	2
Závěr	3

Úvod

Text úvodu.

Kapitola 1

Charakteristika problému

1.1 Současný stav systému

1.2 Problémy stávajícího systému

1.3 Souhrn problémů a podmínek řešení

1.4 Návrh řešení

1.5 Cíl práce

Závěr

Text závěru.

Literatura

```
1  import javax.swing.JApplet;
2  import java.awt.Graphics;
3
4  public class Hello extends JApplet {
5      public void paintComponent(Graphics g) {
6          g.drawString("Hello, world!", 65, 95);
7      }
8  }
```

```
1
2
3
4
5  package com.example.admin.androidactivity;
6  import android.os.Bundle;
7  import android.support.annotation.Nullable;
8  import android.support.design.widget.FloatingActionButton;
9  import android.support.design.widget.Snackbar;
10 import android.support.v7.app.AppCompatActivity;
11 import android.support.v7.widget.Toolbar;
12 import android.view.View;
13 import android.view.Menu;
14 import android.view.MenuItem;
15
16 public class ActivityExample extends AppCompatActivity {
17     @Override
18     protected void onStart() {
19         super.onStart();
20     }
21
22     @Override
23     protected void onCreate(@Nullable Bundle savedInstanceState) {
```



```

24         super.onCreate(savedInstanceState);
25     }
26
27     @Override
28     protected void onStop() {
29         super.onStop();
30     }
31
32     @Override
33     protected void onResume() {
34         super.onResume();
35     }
36
37     @Override
38     protected void onPause() {
39         super.onPause();
40     }
41
42     @Override
43     protected void onDestroy() {
44         super.onDestroy();
45     }
46 }

```

Listing 1.1: Simple code listing.

```

1     /* Block comment */
2
3     package hello
4
5     import kotlin.collections.* // line comment
6
7
8     /**
9      * Doc comment here for 'SomeClass'
10     * @see Iterator#next()
11     */

```

```

9      @Deprecated("Deprecated class")
10     private class MyClass<out T : Iterable<T>>(var prop1 : Int) {
11         fun foo(nullable : String?, r : Runnable, f : () -> Int,
12             fl : FunctionLike, dyn: dynamic) {
13             println("length\nis ${nullable?.length} \e")
14             val ints = java.util.ArrayList<Int?>(2)
15             ints[0] = 102 + f() + fl()
16             val myFun = { -> "" };
17             var ref = ints.size
18             ints.lastIndex + globalCounter
19             ints.forEach lit@ {
20                 if (it == null) return@lit
21                 println(it + ref)
22             }
23             dyn.dynamicCall()
24             dyn.dynamicProp = 5
25         }
26
27         val test = """hello
28                     world
29                     kotlin"""
30         override fun hashCode(): Int {
31             return super.hashCode() * 31
32         }
33     }
34     fun Int?.bar() {
35         if (this != null) {
36             println(message = toString())
37         }
38         else {
39             println(this.toString())
40         }
41     }
42     var globalCounter : Int = 5

```

```
43         get = field
44     abstract class Abstract {
45     }
46     object Obj
47     enum class E { A, B }
48     interface FunctionLike {
49         operator fun invoke() = 1
50     }
```
