

LAB 1

T1. Asenna ympäristö, T2. Opettele ympäristö, T3. HelloWorld App

T4. Study following object oriented concepts

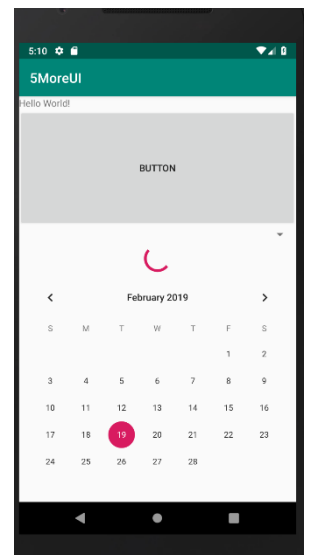
1. Object
Luokan ilmentymä
2. Class
Olion tyyppi
3. Instantiation of object (creating an object)
Luodaan olio eli ilmentymä luokasta.
4. Visibility (public / private / protected)
public = näkyy "koko maailmassa", private = näkyy vain luokassa, ei aliluokissa, protected = näkyy aliluokissa että omassa pakkauksessa
5. Member datas / methods
6. Inheritance
Periytyminen (inheritance) on luokan piirteiden - kenttien ja metodien - siirtymistä toiselle luokalle. Saava luokka on aliluokka (subclass), antava luokka on ylliluokka (superclass). Aliluokkaa kutsutaan joskus myös laajennetuksi (extended) tai johdetuksi (derived) luokaksi.
7. Interface
8. Polymorphism
9. Overriding
10. Abstract classes

T5. More UI

Pieni esimerkki UI:ssä käytettävistä komponenteista..

T6. Study Android fundamental concepts

1. What programming languages you can use for Android app development?
Java, Kotlin
2. What is .apk file?
3. How Android system runs apps?
4. Name four types of Android components. Describe each.
aktiviteetti,
5. What is manifest file and what is its purpose?
manifest.xml kuvaa sovelluksen sisällön ja ominaisuudet. Siellä voidaan määritellä esimerkiksi, kuinka sovellus käynnistetään sekä kerrotaan sovelluksen tukema minimi android API-versio sekä sovelluksen tarvitsemat käyttöoikeudet.



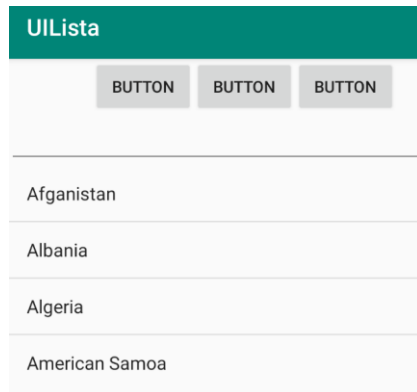
6. What are resources? Why they are needed?

Resurssitiedostoja ovat mm. kuva- tai muu binääritiedosto, merkkijonoja, animaatioita, näytön tai valikon rakennekuvauksia, aktiviteettejä ja sisällöntarjoajia.

LAB 2

T1. U ILista

Taulukko, ListView, ArrayAdapter. Painikkeiden toiminnallisuus myöhemmin tehtävässä 4.



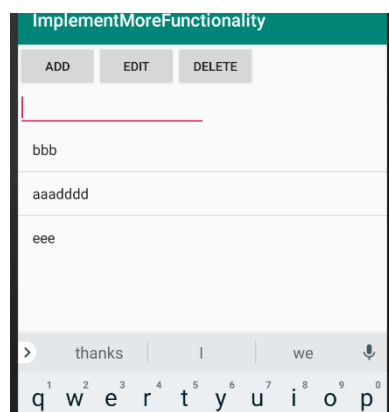
T2. UI Big Button

Kasvattaa lukumäärää painallusten mukaan.

T3. UI without layout

T4. UI

Listan käsittelyä: Listaan voi lisätä rivin, muokata olemassa olevaa riviä tai poistaa rivin.



```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    ArrayList<String> COUNTRIES = new ArrayList<>();
    ListView lv;
    EditText editText;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        findViewById(R.id.addbtn).setOnClickListener(this);
        findViewById(R.id.editbtn).setOnClickListener(this);
        findViewById(R.id.deletebtn).setOnClickListener(this);

        editText = findViewById(R.id.editText);
        lv = (ListView) findViewById(R.id.listView);
    }

    private String getTextFieldText() {
        String teksti = editText.getText().toString();
        editText.setText(null);
        return teksti;
    }

    @Override
    public void onClick(View view) {
        if (view.getId() == R.id.addbtn) {
            String kaupunki = getTextFieldText();
            COUNTRIES.add(kaupunki);
            paivita();
        } else if (view.getId() == R.id.editbtn) {
            haeitem();
        } else if (view.getId() == R.id.deletebtn) {
            poistaitem();
        }
    }

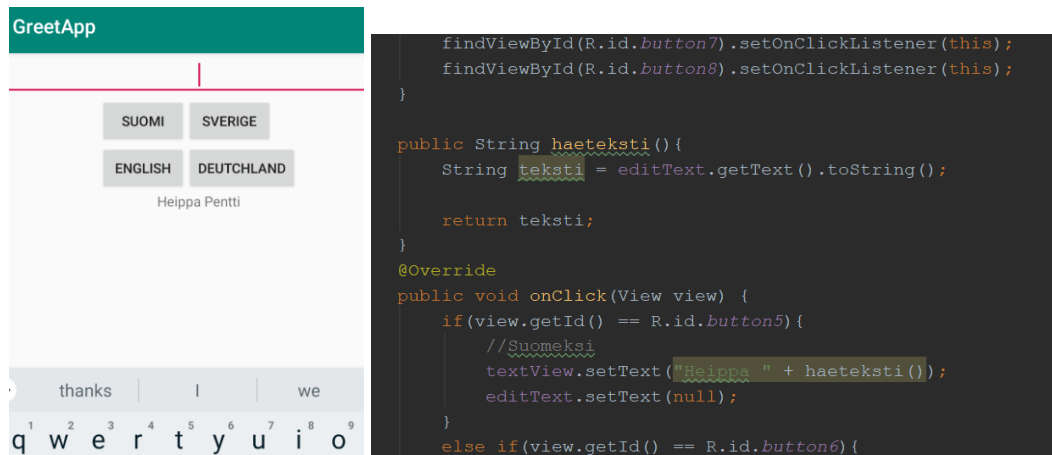
    public void paivita() {
        final ArrayAdapter<String> aa;
        aa = new ArrayAdapter<String>(context, this, android.R.layout.simple_list_item_1, COUNTRIES);
        lv.setAdapter(aa);
    }

    public void haeitem() {
        lv.setOnItemClickListener((parent, view, position, id) -> {
            String selectedItem = (String) parent.getItemAtPosition(position);
            editText.setText(selectedItem);
            COUNTRIES.remove(position);
            paivita();
        });
    }

    public void poistaitem() {
        lv.setOnItemClickListener((parent, view, position, id) -> {
            COUNTRIES.remove(position);
            paivita();
        });
    }
}
```

T5. Greet App

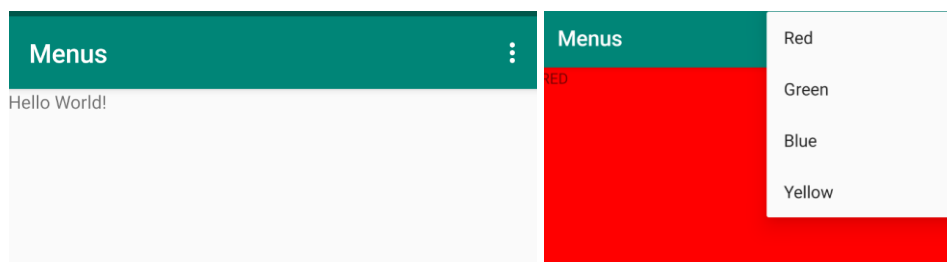
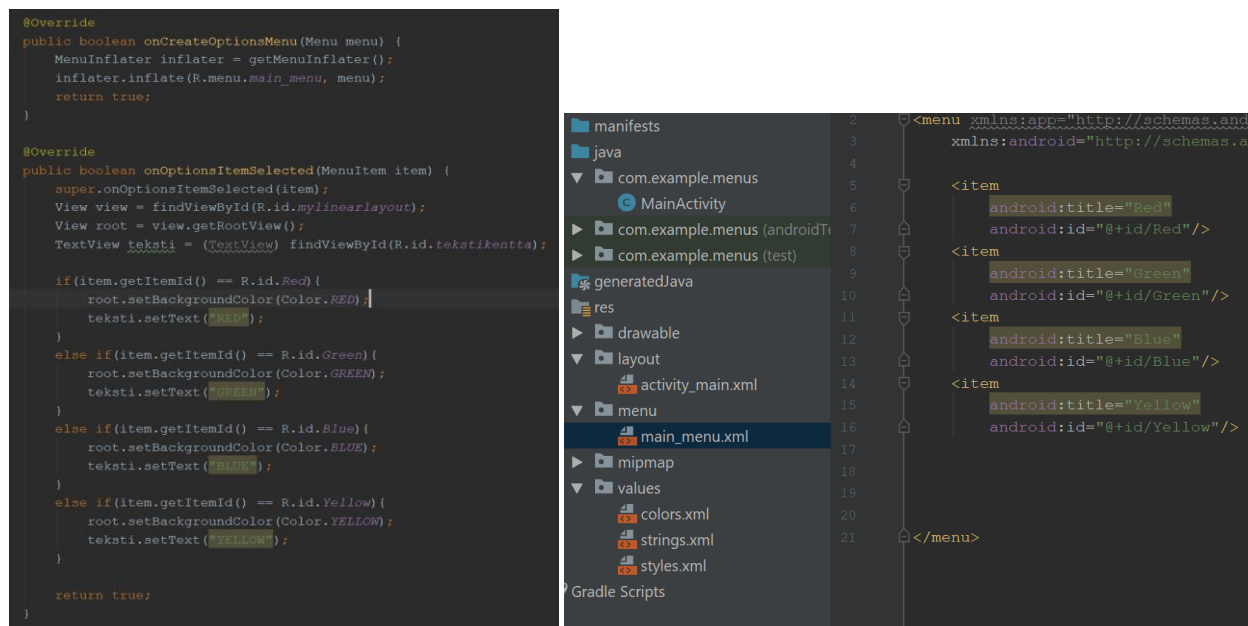
Sovellus tervehtii halutulla kielellä.



LAB 3

T1. Menu

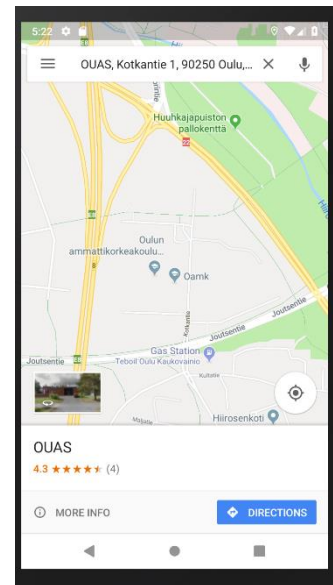
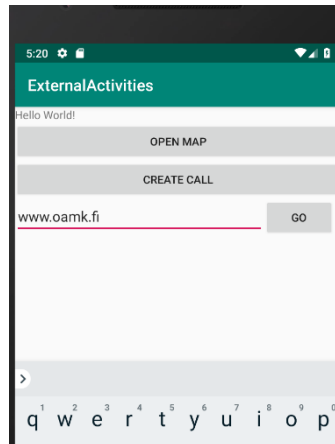
Tehtävä johdatteli menu-valikon käyttöön.



T2. External Activities

Kolme toimintoa: Open Map avaa Google Mapsin,
Create Call puhelimen sovelluksen ja edit text-ruutuun
voi syöttää urlin, joka avataan selaimella.

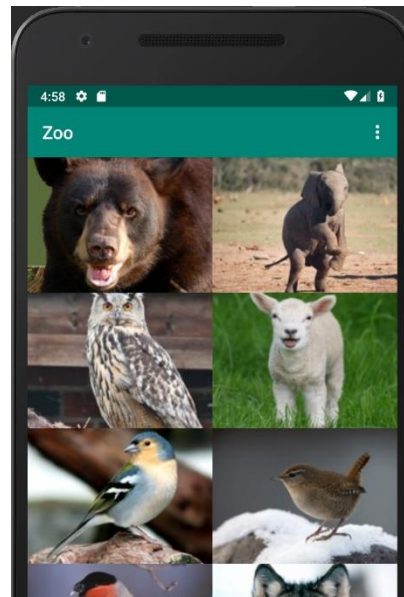
```
button.setOnClickListener(v -> {
    String url = "https://www.google.fi/maps/place/Oamk/#64.999488,25.3096";
    Intent intent = new Intent(Intent.ACTION_VIEW);
    intent.setData(Uri.parse(url));
    startActivity(intent);
});
button2.setOnClickListener(v -> {
    Uri number = Uri.parse("tel:5551234");
    Intent callIntent = new Intent(Intent.ACTION_DIAL, number);
    startActivity(callIntent);
});
button3.setOnClickListener(v -> {
    String url2 = "https://";
    url2 = url2+editText.getText().toString();
    Uri webpage = Uri.parse(url2);
    Intent webIntent = new Intent(Intent.ACTION_VIEW, webpage);
    startActivity(webIntent);
});
```



T3. Zoo

Sovellus käyttää MediaPlayer -luokkaa. Äänet tallennettu
raw-kansioon ja kuvat drawable-kansioon.

```
public boolean onTouch(View v, MotionEvent event) {
    if(v.getId() == R.id.mammal1){
        mediaPlayer = MediaPlayer.create(context, this, R.raw.bear);
        mediaPlayer.start();
    }
    else if(v.getId() == R.id.mammal2){
        mediaPlayer = MediaPlayer.create(context, this, R.raw.elephant);
        mediaPlayer.start();
    }
    else if(v.getId() == R.id.mammal3){
        mediaPlayer = MediaPlayer.create(context, this, R.raw.lamb);
        mediaPlayer.start();
    }
    else if(v.getId() == R.id.mammal4){
        mediaPlayer = MediaPlayer.create(context, this, R.raw.bird);
        mediaPlayer.start();
    }
}
```

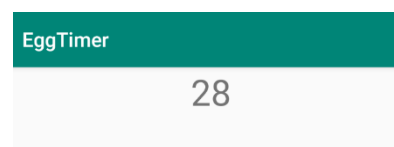
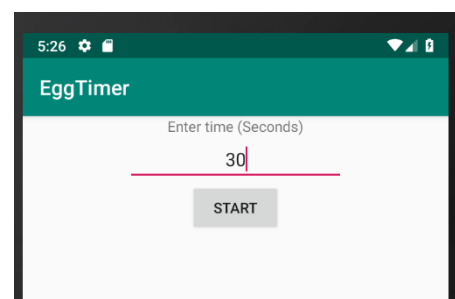


T4. EggTimer

Ajasin. Syötetty aika intent-extrana ajastimelle.

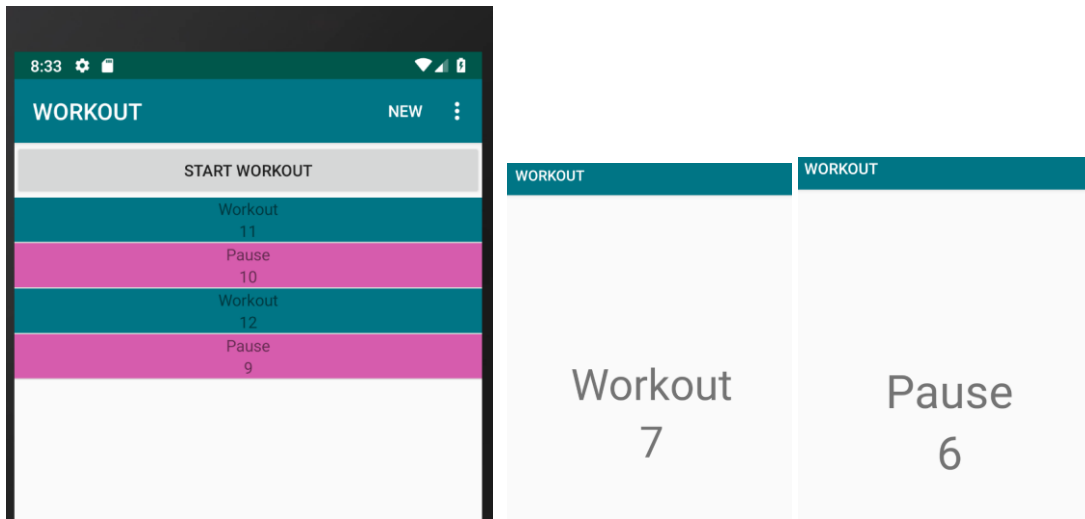
```
button.setOnClickListener(v -> {
    String aika = editText.getText().toString();
    Intent intent = new Intent(packageContext, MainActivity.this, AjastinActivity.class);
    intent.putExtra(name: "aika", aika);
    startActivity(intent);
});
```

```
CountDownTimer countdownTimer = new CountDownTimer(time, countDownInterval 1000) {
    @Override
    public void onTick(long millisUntilFinished) {
        textView.setText("00:" + millisUntilFinished/1000);
    }
    @Override
    public void onFinish() {
        textView.setText("Done!");
        Intent intent = new Intent(packageContext, AjastinActivity.this, MainActivity.class);
        startActivity(intent);
    }
}.start();
```



LAB 4 ja LAB 5 (T1)

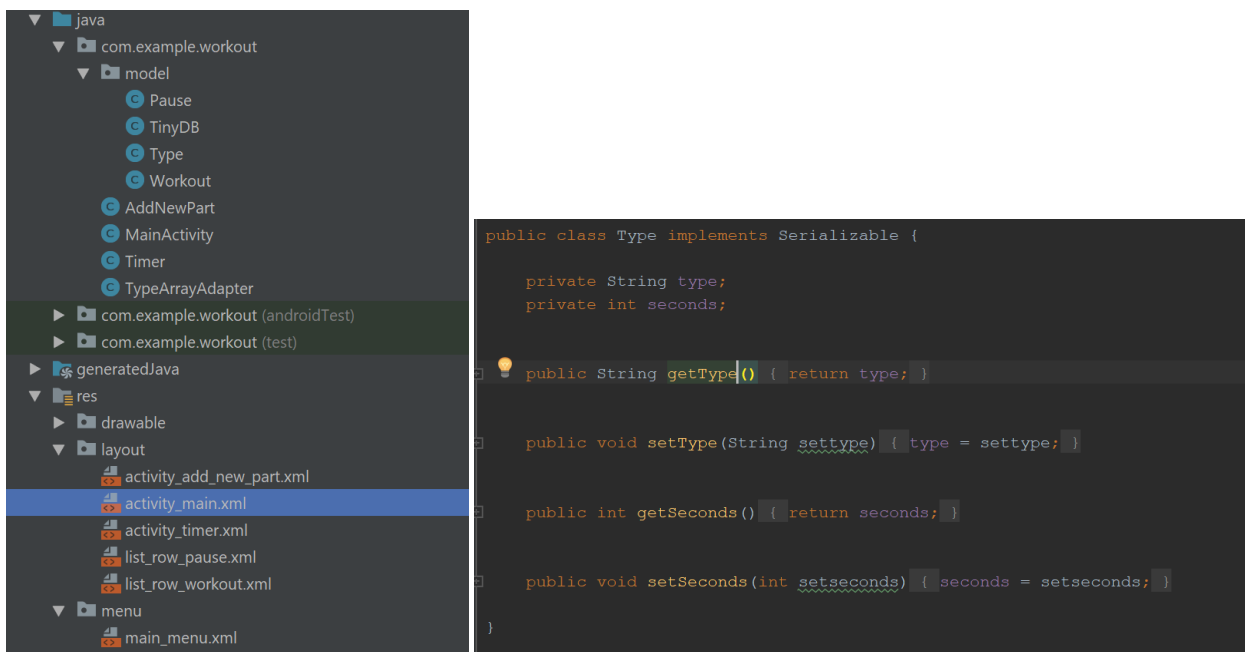
Treenisovellus, johon voi lisätä suoritettavan toimenpiteen ja keston. Starttia painamalla treenin voi aloittaa. Valikosta löytyy myös tallenna ja tyhjennä napit. Tallennusnäppäin tallentaa treenin puhelimen muistiin. Tämä treenisovellus tuntee vain kaksi toimenpidettä: "workout" ja "pause". Näiden yläluokka tunnetaan ohjelmassa nimeltä "Type".



Are you sure?

NO YES

AlertDialog: positive and negative buttons



```

99         alert.show();
100     }
101     return true;
102 }
103 public void saveArrayList(ArrayList<Type> list, String key){
104     SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(context MainActivity.this);
105     SharedPreferences.Editor editor = prefs.edit();
106     Gson gson = new Gson();
107     String json = gson.toJson(list);
108     editor.putString(key, json);
109     editor.apply(); // This line is IMPORTANT !!!
110 }
111
112 public List<Type> getTasksFromSharedPrefs(Context context) {
113     SharedPreferences appSharedPrefs = PreferenceManager.getDefaultSharedPreferences(context.getApplicationContext());
114     Gson gson = new Gson();
115     String json = appSharedPrefs.getString(key: "TYPES", defValue: "");
116     types = gson.fromJson(json, new TypeToken<ArrayList<Type>>() {}.getType());
117     return types;
118 }
119
120 protected void onActivityResult(int requestCode, int resultCode, Intent data){
121     super.onActivityResult(requestCode, resultCode, data);
122     if(requestCode == ADD_WORK_ID && resultCode == RESULT_OK){
123         type = (Type) data.getSerializableExtra( name: "TYPE");
124         secondsLeft = type.getSeconds();
125         typeName = type.getType();
126         types.add(type);
127     }
128 }
129
130 @Override
131 public void onClick(View v) {
132     Intent intent = new Intent( packageContext: MainActivity.this, Timer.class);
133     intent.putExtra( name: "TYPES", types);
134     startActivity(intent);
135 }

```

Pätkä MainActivity:n tärkeimmistä

```

public class AddNewPart extends AppCompatActivity{

    RadioGroup radioGroup;
    EditText editText;
    Button button;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_new_part);
        radioGroup = findViewById(R.id.radiobuttongroup);
        editText = findViewById(R.id.editSeconds);
        button = findViewById(R.id.btnAdd);

        button.setOnClickListener((v) -> {
            int seconds = Integer.parseInt(editText.getText().toString());
            if(radioGroup.getCheckedRadioButtonId() == findViewById(R.id.radiobtnwork).getId()){
                Workout workout = new Workout();
                workout.setType("Workout");
                workout.setSeconds(seconds);
                returnData(workout);
            }
            else {
                Pause pause = new Pause();
                pause.setType("Pause");
                pause.setSeconds(seconds);
                returnData(pause);
            }
        });
    }

    private void returnData(Type data){
        Intent returnIntent = new Intent();
        returnIntent.putExtra( name: "TYPE", data);
        setResult(Activity.RESULT_OK, returnIntent);
        finish();
    }
}

```

Add new -toiminto

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_timer);
    typeview = findViewById(R.id.typeView);
    secondsView = findViewById(R.id.secondLeftView);
    types = (ArrayList<Type>) getIntent().getSerializableExtra( name: "TYPES");
    startTimer(types.get(currentIndex));
    Type type = types.get(currentIndex);
}

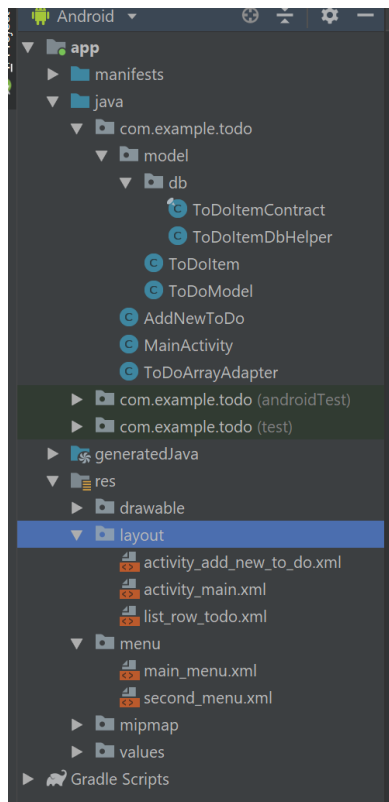
private void startTimer(Type type) {
    final Type current = types.get(currentIndex);
    CountdownTimer countDownTimer = new CountdownTimer( millisInFuture: current.getSeconds()*1000, (countDownInterval: 1000) {
        @Override
        public void onTick(long millisUntilFinished) {
            typeview.setText(current.getType());
            secondsView.setText(" " + millisUntilFinished/1000);
        }
        @Override
        public void onFinish() {
            typeview.setText("Done!");
            currentIndex++;
            if(currentIndex < types.size())
            {
                startTimer(types.get(currentIndex));
            }
            else{
                finish();
            }
        }
    }).start();
}
}

```

Toiminnallisuus näkymälle, kun treeni on käynnissä.

LAB 5 t2.

ToDo -appi, jonka todoot tallennetaan paikalliseen tietokantaan sql:llä.



ToDoModel-luokassa on lisää, poista, päivitä metodit tietokannalle. Omalla ToDoItemDbHelperillä luodaan tietokannan rakenne, ToDoItemContractissa on tietokannan sarakkeiden nimet määritelty, ToDoItem-luokassa määritellään objektin muuttujat.

```

public long addToDb(ToDoItem addable) {
    // Gets the data repository in write mode
    SQLiteDatabase db = mDbHelper.getWritableDatabase();

    // Create a new map of values, where column names are the keys
    ContentValues values = new ContentValues();
    values.put(ToDoItemContract.ToDoItem.COLUMN_NAME_TITLE, addable.title);
    values.put(ToDoItemContract.ToDoItem.COLUMN_TODO_DATE, addable.date);
    values.put(ToDoItemContract.ToDoItem.COLUMN_TODO_DESCRIPTION, addable.description);

    // Insert the new row, returning the primary key value of the new row
    long newRowId = db.insert(ToDoItemContract.ToDoItem.TABLE_NAME, null, values);
    db.close();
    return newRowId;
}

```

```

        listView.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
            @Override
            public boolean onItemLongClick(AdapterView<?> parent, View view, int position,
                todoItem = todos.get(position);

                todoId = todoItem.getId();
                model.deleteTodo(todoId);
                onResume();
                return false;
            }
        });
    }

    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.second_menu, menu);
        return true;
    }

    public boolean onOptionsItemSelected(MenuItem item) {
        if (item.getItemId() == R.id.addnew) {
            Intent intent = new Intent( packageContext MainActivity.this, AddNewToDo.class);
            startActivity(intent);
        }

        if (item.getItemId() == R.id.deleteall) {
            model.deleteAll();
            onResume();
        }

        return true;
    }

    @Override
    protected void onResume() {
        super.onResume();
        todos = model.getRecords();
        ToDoArrayAdapter adapter = new ToDoArrayAdapter( context this, todos);
        listView.setAdapter(adapter);
    }
}

```

MainActivity:n toiminnallisuutta

```

public class AddNewToDo extends AppCompatActivity {

    ToDoModel model = null;

    EditText titleEdit;
    EditText descriptionEdit;
    CalendarView calendarView;
    String newDate;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_new_to_do);

        model = new ToDoModel( context this);
        titleEdit = findViewById(R.id.titleEdit);
        descriptionEdit = findViewById(R.id.descriptionEdit);
        calendarView = (CalendarView) findViewById(R.id.calendarView);
        calendarView.setOnDateChangeListener(new CalendarView.OnDateChangeListener() {
            @Override
            public void onSelectedDayChange(CalendarView view, int year, int month, int
                newDate = year+"-"+(month+1)+"-"+dayOfMonth;
            }
        });
    }

    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.main_menu, menu);
        return true;
    }

    public boolean onOptionsItemSelected(MenuItem item) {
        if (item.getItemId() == R.id.save) {
            String title = titleEdit.getText().toString();
            String description = descriptionEdit.getText().toString();
            long id = 0;
            ToDoItem todoItem1 = new ToDoItem(id, newDate, title, description );
            id = model.addToDb(todoItem1);
            todoItem1.setId(id);
            model.updateTodo(todoItem1);
            finish();
        }

        return true;
    }
}

```

AddNewToDo:n toiminnallisuutta.

ToDo

Add New Todo

Delete all

Muista soittaa takaisin äidille

Muista velka

Muista maksaa velat Sofialle

ToDo

SAVE

Otsikko

Kuvaus

Tapahtuman päivämäärä:

<

February 2019

>

S

M

T

W

T

F

S

1

2

Päänäkymän valikosta pääsee lisäämään Todoon tai vaikkapa tyhjentämään koko tietokannan. Yhtä todoota pitkään painamalla saa yksittäisen todoon poistettua tietokannasta.

LAB 6 t1

Yksinkertainen sovellus muistutuksen lisäämiseen. Sovelluksella valitaan muistettava aika NumberPickereiden avulla ja luodaan Notification omalla tekstillä. Notification ilmestyy androidin ilmoituspalkkiin.

REMINDERAPP

Set reminder time:

23

59

0

0

1

1

Set reminder text:

DONE

5:15

Tue, Feb 26

Google Play Store • now

Google Play Protect is turned on

This feature protects your device from harmful apps

REMINDERAPP • now

Tama on minun muistutus tekstini

5:47

Android Setup

Finish setting up your Android SDK built for x86

Copy your data, set wallpaper, and more

Settings

Virtual SD card

Tap to set up

Download Manager • Updating Play support libraries...

```

numberPicker1 = findViewById(R.id.numberPicker);
numberPicker1.setMinValue(00);
numberPicker1.setMaxValue(23);
numberPicker1.setOnValueChangedListener(onValueChangedListener);

numberPicker2 = findViewById(R.id.numberPicker2);
numberPicker2.setMinValue(00);
numberPicker2.setMaxValue(59);
numberPicker2.setOnValueChangedListener(onValueChangedListener2);

```

```

NumberPicker.OnValueChangeListener onValueChangeListener = (picker, oldVal, newVal) -> {
    hour = newVal;
};

NumberPicker.OnValueChangeListener onValueChangeListener2 = (picker, oldVal, newVal) -> {
    minute = newVal;
};

```

```

protected void showNotification() {
    NotificationCompat.Builder mBuilder = new NotificationCompat.Builder( context, this, RemainderApplication.CHANNEL_ID
        .setSmallIcon(R.drawable.icon)
        .setContentTitle(""+editText.getText().toString())
        .setContentText(""+hour+"":"+minute)
        .setPriority(NotificationCompat.PRIORITY_DEFAULT);

    NotificationManagerCompat notificationManager = NotificationManagerCompat.from(this);

    // notificationId is a unique int for each notification that you must define
    notificationManager.notify( id: 234, mBuilder.build());
}

protected void broadcastDemo() {
    AlarmManager alarmMgr;
    PendingIntent alarmIntent;

    alarmMgr = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
    Intent intent = new Intent( packageContext, this, AlarmBroadcastReceiver.class);
    alarmIntent = PendingIntent.getBroadcast( context, this, requestCode: 0, intent, flags: 0);

    alarmMgr.set(AlarmManager.ELAPSED_REALTIME_WAKEUP,
        triggerAtMillis: SystemClock.elapsedRealtime() +
            30 * 1000, alarmIntent);
}

```

```

public class RemainderApplication extends Application {

    static final String CHANNEL_ID = "NOTIFICATION_CHANNEL_ID";
    static RemainderApplication instance = null;

    TextAppModel model = new TextAppModel();

    @Override
    public void onCreate() {
        super.onCreate();
        createNotificationChannel();
        instance = this;

        Log.d( tag: "DEMO_6", msg: "APPLICATION onCreate");
    }

    private void createNotificationChannel() {
        // Create the NotificationChannel, but only on API 26+ because
        // the NotificationChannel class is new and not in the support library
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            CharSequence name = getString(R.string.channel_name);
            String description = getString(R.string.channel_description);
            int importance = NotificationManager.IMPORTANCE_DEFAULT;
            NotificationChannel channel = new NotificationChannel(CHANNEL_ID, name, importance);
            channel.setDescription(description);
            // Register the channel with the system; you can't change the importance
            // or other notification behaviors after this
            NotificationManager notificationManager = getSystemService(NotificationManager.class);
            notificationManager.createNotificationChannel(channel);
        }
    }

    public static RemainderApplication getInstance() { return instance; }

    public TextAppModel getModel() { return model; }

    public void setModel(TextAppModel model) { this.model = model; }
}

```

LAB6 t2

Yksinkertainen Note App, johon voi kirjoittaa tekstiä. Tallennus tapahtuu automaattisesti onStop()-metodissa kun sovellus suljetaan ja onStop() kutsutaan. Sovellus tallentaa OutputStreamWriter -luokan avulla txt-päätteiseen tekstitiedostoon muistion.



```
public void Save(String fileName) {
    try {
        OutputStreamWriter out =
            new OutputStreamWriter(openFileOutput(fileName, mode));
        out.write(editText.getText().toString());
        out.close();
        Toast.makeText(context, "Note Saved!", Toast.LENGTH_SHORT).show();
    } catch (Throwable t) {
        Toast.makeText(context, "Exception: " + t.toString(), Toast.LENGTH_LONG).show();
    }
}

protected void onStop() {
    Save(fileName);
    super.onStop();
    Save(fileName);
}

public boolean FileExists(String fname){
    File file = getBaseContext().getFileStreamPath(fname);
    return file.exists();
}

public String Open(String fileName) {
    String content = "";
    if (FileExists(fileName)) {
        try {
            InputStream in = openFileInput(fileName);
            if (in != null) {
                InputStreamReader tmp = new InputStreamReader(in);
                BufferedReader reader = new BufferedReader(tmp);
                String str;
                StringBuilder buf = new StringBuilder();
                while ((str = reader.readLine()) != null) {
                    buf.append(str + "\n");
                }
                in.close();
                content = buf.toString();
            }
        } catch (java.io.FileNotFoundException e) {} catch (Throwable t) {
            Toast.makeText(context, "Exception: " + t.toString(), Toast.LENGTH_LONG).show();
        }
    }
    return content;
}
```