Lappeenrannan teknillinen yliopisto
School of Business and Management

Sofware Development Skills

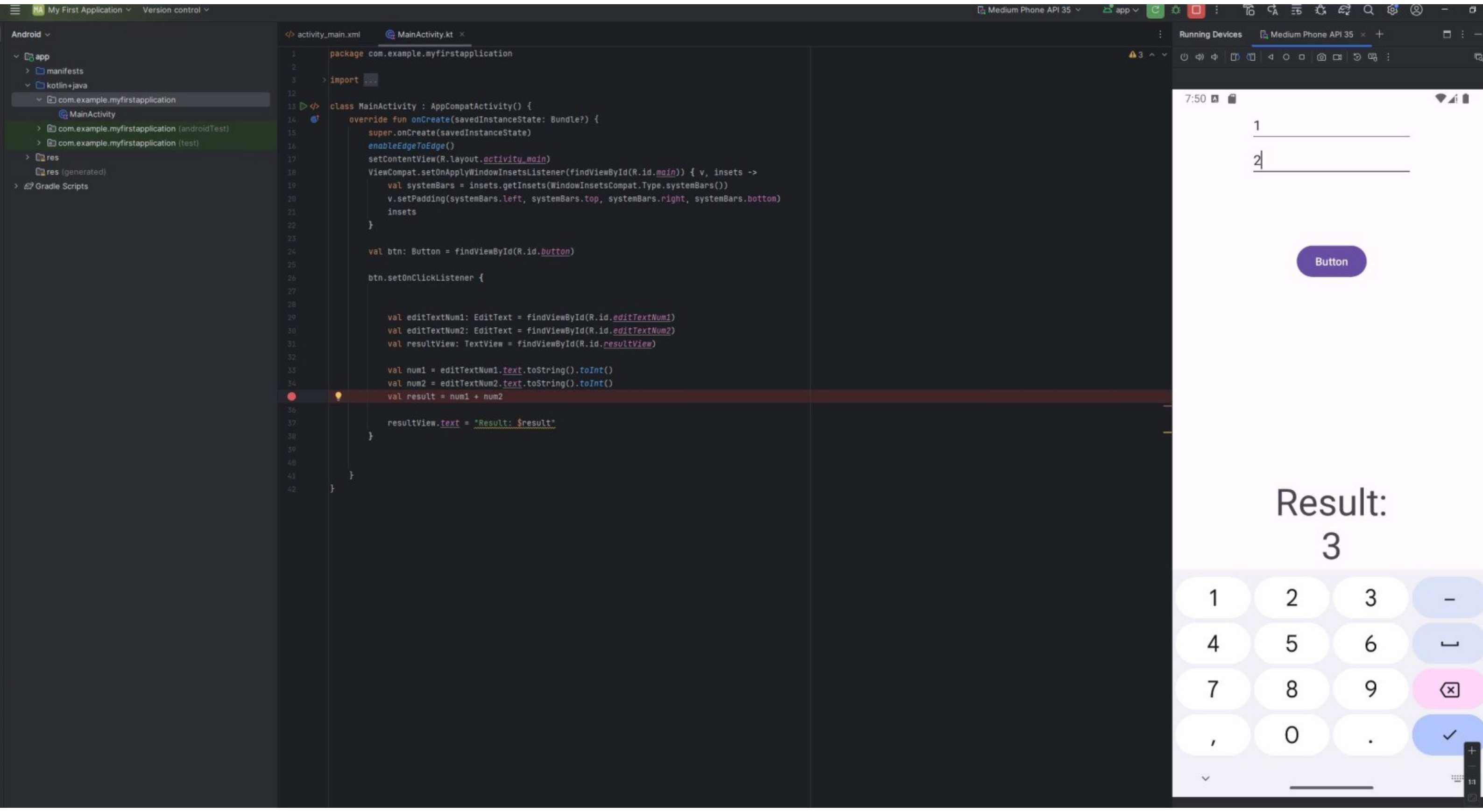
Joonas Liedes, 611625

LEARNING DIARY, MOBILE DEVELOPMENT MODULE

LEARNING DIARY

26.8.2024

Read through the course information and installed all necessary software. I chose Android Studio as my development platform as I've previously found it reasonably straightforward to work with. I watched through the first chapter of the tutorial and followed along in Android Studio for the layout portion of the tutorial. However, I decided to implement the functional parts of the tutorial in Kotlin, as I haven't used it before and thought this might be a good time to learn some. For this purpose, I watched a short introduction to Kotlin in YouTube and asked ChatGPT some basic syntax questions. I also refreshed my memory regarding Android Studio's debugging system. Moreover, I set up Git with Github Desktop to manage version control.



```kotlin
package com.example.myfirstapplication

import ...

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
        }

        val btn: Button = findViewById(R.id.button)

        btn.setOnClickListener {


            val editTextNum1: EditText = findViewById(R.id.editTextNum1)
            val editTextNum2: EditText = findViewById(R.id.editTextNum2)
            val resultView: TextView = findViewById(R.id.resultView)

            val num1 = editTextNum1.text.toString().toInt()
            val num2 = editTextNum2.text.toString().toInt()
            val result = num1 + num2

            resultView.text = "Result: $result"
        }


    }
}
```

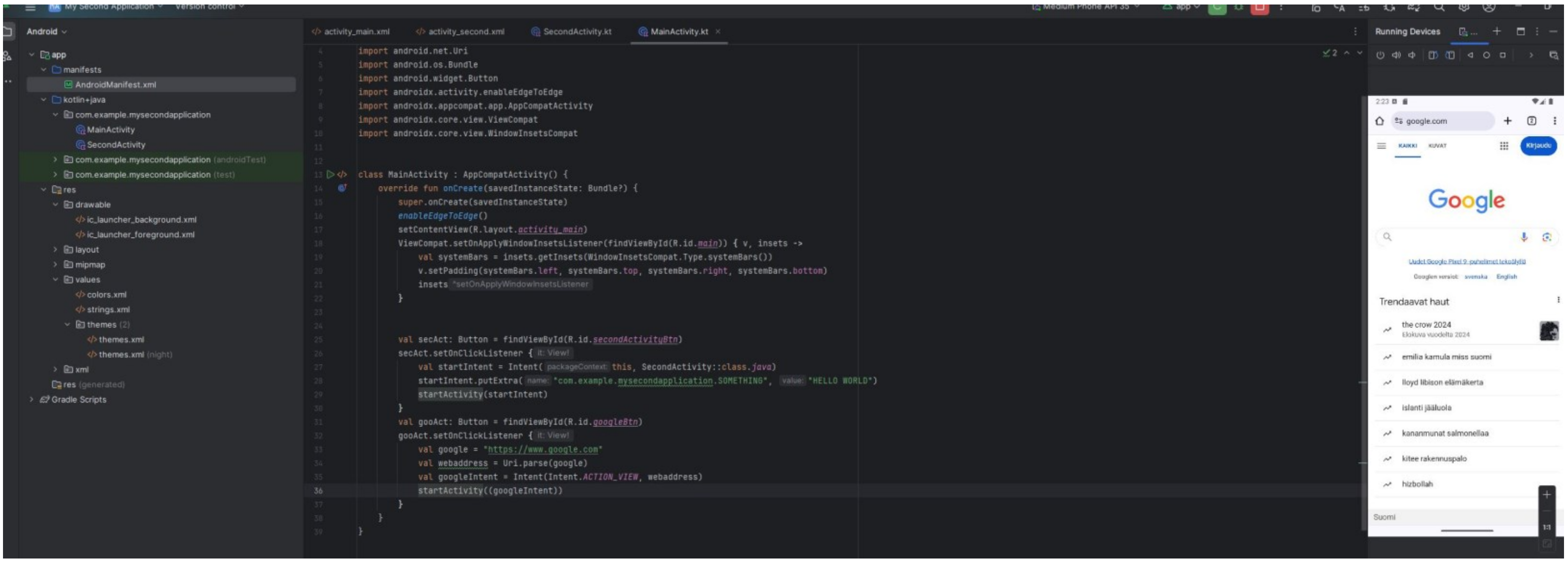
1

2

Result:
3

26.8.2024 – Part 2

Went over the part 2 of the tutorial and adapted it for Kotlin where necessary. Learned the use of multiple views and how to transfer information via intents. I've been mostly working with Python and data science problems lately and it's been a while since I've done anything Java or anything app development related, so this has been quite laborious, especially because Kotlin is a new language for me.



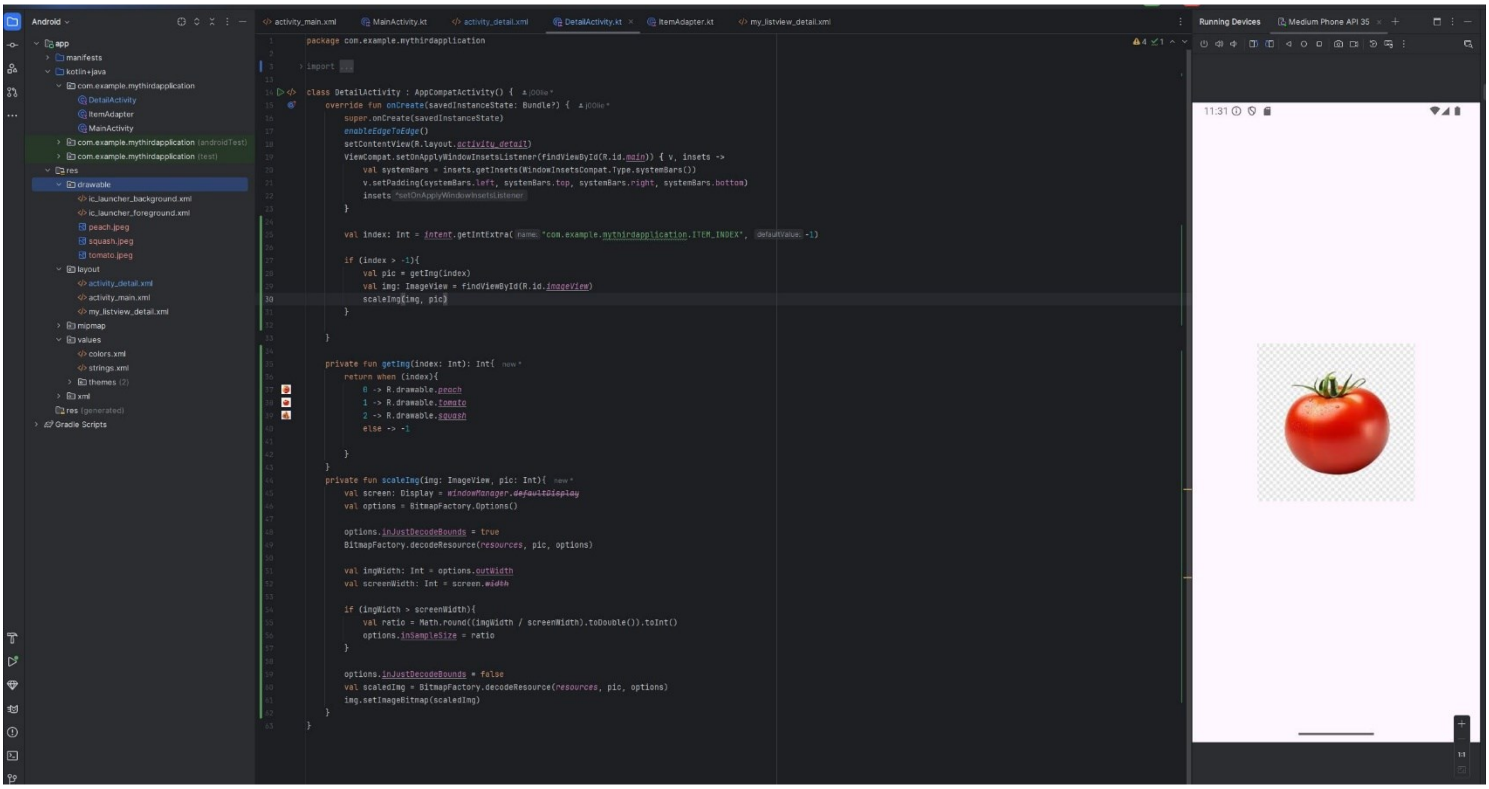
```kotlin
import android.net.Uri
import android.os.Bundle
import android.widget.Button
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat


class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets ^setOnApplyWindowInsetsListener
        }


        val secAct: Button = findViewById(R.id.secondActivityBtn)
        secAct.setOnClickListener { it: View!
            val startIntent = Intent( packageContext: this, SecondActivity::class.java)
            startIntent.putExtra( name: "com.example.mysecondapplication.SOMETHING",  value: "HELLO WORLD")
            startActivity(startIntent)
        }
        val gooAct: Button = findViewById(R.id.googleBtn)
        gooAct.setOnClickListener { it: View!
            val google = "https://www.google.com"
            val webaddress = Uri.parse(google)
            val googleIntent = Intent(Intent.ACTION_VIEW, webaddress)
            startActivity((googleIntent))
        }
    }
}
```

27.8 – 28.8.2024

Completed the third module of the series and built the example app demonstrating the use of ListView, a custom layout component and ImageView. Building the functionality with Kotlin differed somewhat syntactically from the Java example, which required some studying, but in the end I'm enjoying the brevity of Kotlin vs. Java. The when – statement seems powerful in Kotlin compared to switch-case in Java. I used YouTube and ChatGPT to learn more about Kotlin for this section.



```kotlin
package com.example.mythirdapplication

import ...

class DetailActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_detail)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
        }

        val index: Int = intent.getIntExtra("com.example.mythirdapplication.ITEM_INDEX", -1)

        if (index > -1){
            val pic = getImg(index)
            val img: ImageView = findViewById(R.id.imageView)
            scaleImg(img, pic)
        }

    }

    private fun getImg(index: Int): Int{
        return when (index){
            0 -> R.drawable.peach
            1 -> R.drawable.tomato
            2 -> R.drawable.squash
            else -> -1

        }
    }
    private fun scaleImg(img: ImageView, pic: Int){
        val screen: Display = windowManager.defaultDisplay
        val options = BitmapFactory.Options()

        options.inJustDecodeBounds = true
        BitmapFactory.decodeResource(resources, pic, options)

        val imgWidth: Int = options.outWidth
        val screenWidth: Int = screen.width

        if (imgWidth > screenWidth){
            val ratio = Math.round((imgWidth / screenWidth).toDouble()).toInt()
            options.inSampleSize = ratio
        }

        options.inJustDecodeBounds = false
        val scaledImg = BitmapFactory.decodeResource(resources, pic, options)
        img.setImageBitmap(scaledImg)
    }
}
```

29.8.2024

Started building the final project. The plan is to build a task manager, where the user can set tasks and delete them. They can also tap open the tasks and examine their details. Tasks may also be deleted when completed. The UI will also include a check box for each task to keep track of their status. Inspiration for this project came from this YouTube video. Today I managed to build the layout for the opening screen containing the recycler view that holds all the tasks and the layout for an individual task. In addition, I completed most of the logic for these layouts by building a class for the tasks and an Adapter for the recycler view. Had to use view binding in the adapter as the synthetic technique used in the video seems to be deprecated.

30.8.2024

Finished building the logic for the layouts. Finding workarounds for the deprecated sections of the tutorial took some time. In addition, I built the layouts for an individual detailed view of task that opens when clicked. Within this view the name and status of the task is shown along with an option to pick a deadline for the task. The deadline is saved using a button, which then redirects the user to the main view. The updated information of the task with the deadline is then show in the main view as well. After building the layouts, I built the logic for this section in a similar way to the first section of the project. View binding was used to interact with the layout elements. Data is transferred using intents between the activities. A list is used to store the data. A database could of course be implemented using Android's Room layer; however, I felt this might have been overkill for this project and the scope of this course. The application seems to be working nicely and I was not able to produce any bugs. In summary, this project was pretty challenging for me, as most of my prior development experience is with data science and machine learning type of things, especially because I had no prior experience with Kotlin. However, I enjoyed the challenge and I’m happy with the outcome, even though it is a rather basic application.