My GPA Calculator

Eamonn Conway, Abdul Samad and Zachary Ballard

Siena College

Loudonville, NY 12211

Abstract

We use apps everyday and we're interested in how they are made. After some research, we found that a popular language used to code apps that we haven't used before is Kotlin. Kotlin is a new language with a lot of potential. This paper demonstrates how we learned many of the unique features Kotlin has to offer by designing and creating a mobile application to calculate a user's GPA.

1 Overview

For our project, we decided to study the programming language Koltin. Kotlin is statically typed language just like haskell. It is a very complex and capable language that shares numerous similarities with Java, however it is much newer. Kotlin is most popularly used to create smartphone applications through Android software because it has all the features that Java has, and then some. One extra feature is safer code that allows fewer errors which in turn causes fewer crashes and system failures. Its compiler is also better because it detects more errors at compile time instead of run time. And, it allows for inline functions which prevents memory allocation of the function over and over again, and the jumping of CPU at runtime. This can make applications run more efficiently. (Kotlin API) These are just a few examples of why

Koltin is probably the best language to use to make an application that runs on android software.

In order to fully understand the language, we decided to create a simple GPA calculator application. To begin, we have downloaded the Android Studio software on our team computers in the software engineering lab. We then began creating our project. When setting up a new project we simply selected Kotlin as our desired language and selected an empty template. We then began working on the layout of the app.

2 The Display

We decided to display 6 class options where there is a row for class number, number of credits, and grade for the class. We chose 6 class options even though most students take 4-5 classes per semester, because some may take 6. In order to create this layout, we used an XML file with a code block for each individual item. The course numbers were XML TextView objects which each displayed their corresponding course number. These are assigned by the programmer and remain unchanged throughout the applications life. The course credits are displayed as XML EditText objects. Inside these object descriptions, we added the line

```
android:inputType="number"
```

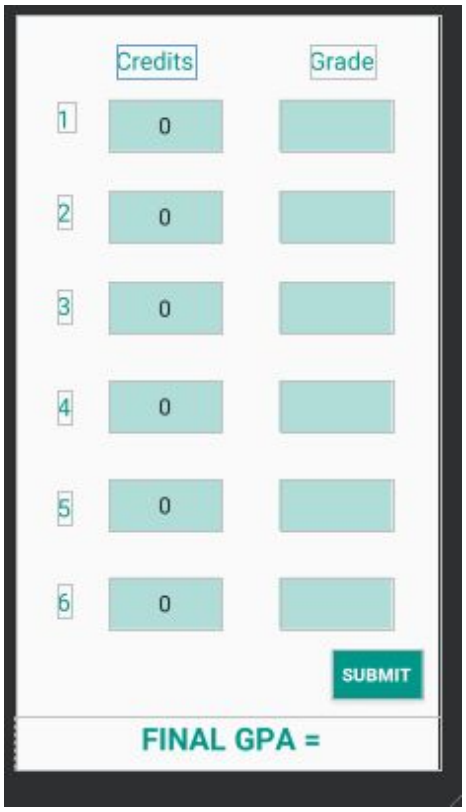
which made sure the user of the app could only input a number. The dropdown boxes in Kotlin can be implemented using the Spinner class.They can be created in the app layout by using XML Spinner objects. (CodeAndroid) Spinners in android development are similar to dropdown list. We created an options array in our main Kotlin file and used the ArrayAdapter Kotlin Class to bind each spinner with the arraylist. This allowed for the dropdown to be populated with the elements of the options array.A code snippet of this is shown below.

```
val options = arrayOf(" ", "A", "A-", "B+", "B", "B-", "C+", "C",
"C-", "D+", "D", "D-","F")
spinner1.adapter =
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,options
)
spinner2.adapter =
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,options
)
spinner3.adapter =
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,options
)
spinner4.adapter =
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,options
)
spinner5.adapter =
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,options
)
spinner6.adapter =
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,options
)
```

The final two items were a submit button which was an XML Button object which when selected will call our Kotlin onClick function to calculate the final gpa and an XML TextView object to display the “Final GPA =”. Once the items were created and put on the app main page, we had to constrain them to the page. We did this by using the constraint layout features, and

we did everything by percentages. A screenshot of the display and the code included to constrain one of the spinner objects are included below.

```
app:layout_constraintBottom_toBottomOf="parent"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.85"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toTopOf="parent"

app:layout_constraintVertical_bias="0.65999997"
```



App 1. This is the bare bones of the app with nothing input.

3 The Back End

Once we formatted everything, we began to work on the back-end Koltin code. One interesting thing that was new to us was the way that Kotlin declares variables. Val and var both are used to declare a variable. Var is like general variable and it's known as a mutable variable in Kotlin and can be assigned multiple times. Val is like Final variable and it's known as immutable in Kotlin and can be initialized only once. (Mgbemena)

After watching some tutorial videos, we discovered it would be necessary to use Listeners on the Spinners and text boxes to read in the data from the user. The listener that we implemented takes action once an item from the dropdown is selected. For the purposes of our app, it takes that item, converts its letter grade value into a double GPA score and places it into the correct index of an array. Each time a user selects a new item from the spinner, the listener will update the array accordingly. (CodeAndroid) Honestly, using listeners in Kotlin was exactly the same as using them in Java, because we had to declare the use of the onItemSelectedListener and then override the methods that we wanted to use.

We also had to use a Switch statement to convert the letter grade inputs to double gpa grades.The switch statement in Kotlin is different from Java. An example is as follows:

```
when (str) {
            " " -> grades[int] = 0.0
            "A" -> grades[int] = 4.0
            "A-" -> grades[int] = 3.7
```

```
            "B+" -> grades[int] = 3.3
            "B" -> grades[int] = 3.0
            "B-" -> grades[int] = 2.7
            "C+" -> grades[int] = 2.3
            "C" -> grades[int] = 2.0
            "C-" -> grades[int] = 1.7
            "D+" -> grades[int] = 1.3
            "D" -> grades[int] = 1.0
            "D-" -> grades[int] = 0.7
            "F" -> grades[int] = 0.0
 }
```

(Antonio) Instead of switch and case Kotlin uses when and is respectively. It makes the code more readable. We don't have to use break statements at the end of each case either.

We also had to implement a TextChangedListener for when a user enters credits. There were three methods which we had to override for the textchanged listener, which are afterTextChanged(), before TextChanged() and onTextChanged(). Since we wanted to update the credits array every time a user changes any of the credit boxes, we chose to edit the onTextChanged() method. (CodeAndroid) The parameters that are passed into this method are shown below.

```
override fun onTextChanged(s: CharSequence, start: Int, before: Int,
count: Int)
```

The issue we had with this is that the first input is of the type CharSequence, which we have never seen before.

Kotlin has an interesting way of using their primitive data types. The user types in their credits as integers but the XML passes them in as Char Sequences, which represents a readable sequence of Char values. The only way to convert from char sequences to int was to convert it into a string first, for which we used the toString() method. (Kotlin API) Kotlin also gives us a toInt() method to convert the strings to integers, which was very helpful. We used this to store each class credits in an array.

Kotlin also has an interesting way of dealing with arrays, as shown in the example line of code below. (Kotlin API) In our opinion, this is about the same readability as most other complicated programming languages, but it increases writeability.

```
val grades = DoubleArray(6) { 0.0 }
```

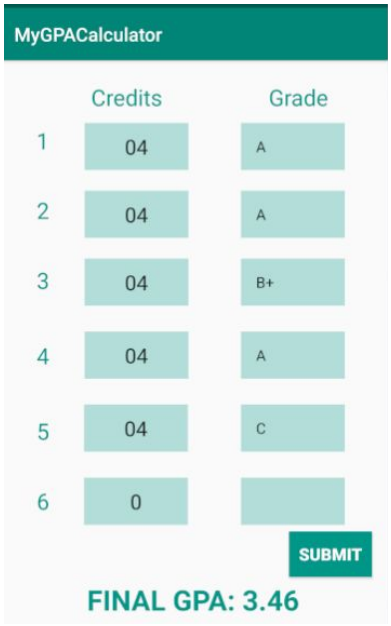
The final portion of code in our program was for when a user presses the submit button. When this occurs, our app takes all of the data currently stored in the two arrays called grades and credits, calculates the final GPA, and displays it on the bottom of the page. In this code, we had to include another listener, this time a setOnClickListener on the submit button. The code block for this is included below. An interesting note on this is Kotlin's way of implementing for loops. (Koltin API) They are very similar to how python does it, and this increases the readability of the code tremendously.

```
val btn_click_me = findViewById(R.id.button) as Button
        btn_click_me.setOnClickListener {
            var finalgpa = 0.0
            var totalCredits = 0
```

```
            for(i in 0..5) {
                finalgpa += (grades[i] * credits[i])
                totalCredits += credits[i]
            }
            finalgpa /= totalCredits
            result.text = "Final GPA: " + finalgpa
        }
```

4 Final

The final version of the app is shown below:



App 2. This is the app with random values input and the gpa calculated.

5 Conclusion

In conclusion, our study and working with Kotlin has really helped us thrive and complete our app. Kotlin was easier to learn because it is like a mix of the two main languages we learned here at Siena, Java and Python. That made learning Kotlin a little less stressful as it took stuff from both languages and mashed them together.

Bibliography

1. Mgbemena, Chike, et al. “Kotlin From Scratch: Variables, Basic Types, and Arrays.” *Code Envato Tuts+*, 11 Aug. 2017, code.tutsplus.com/tutorials/kotlin-from-scratch-variables-basic-types-arrays-type-inference-and-comments--cms-29328.
2. “Using Kotlin for Android Development.” *Kotlin*, https://kotlinlang.org/docs/reference/android-overview.html.
3. CodeAndroid, director. *Android Tutorial (Kotlin) - 12 - Spinner*. *YouTube*, YouTube, 26 June 2017, www.youtube.com/watch?v=D5I7MNIqA3M.
4. Antonio Leiva. “Using ‘When’ Expression in Kotlin: The ‘Switch’ with Super Powers.” *Antonio Leiva*, 28 May 2019, antonioleiva.com/when-expression-kotlin/.