Lab assignment 3

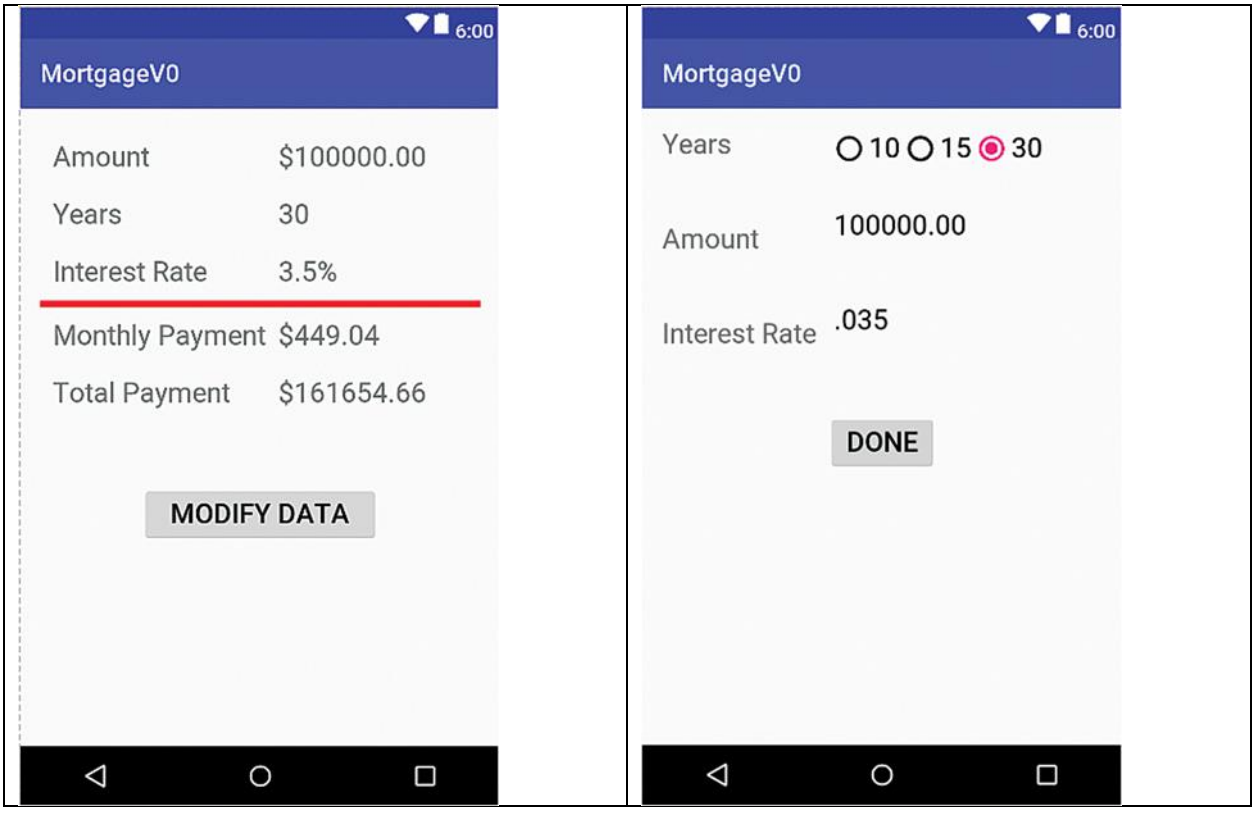
Due date: Friday, November 3

40 points

Problem

In this lab assignment, you will learn how to code several activities, how to go from one to another and back, how to share data between activities, how to set up transitions between them, and how to save the state of an app and retrieve it whenever the user starts the app again (i.e., how to make the data persistent). We build a mortgage calculator app.

Design



Modify button – display the screen on the right. Allow user to input the number of years, amount, and interest rate.

Done button – display the screen on the left.

For this app, the Model is simple and is only composed of one class that encapsulates a mortgage calculator, the Mortgage class. The class below is Java for your sample. You might want to implement in Java.

Mortgage class

```
import java.text.DecimalFormat

class Mortgage {
  val MONEY: DecimalFormat = DecimalFormat("$#,##0.00")

  private var amount = 0f
  private var years = 0
  private var rate = 0f

  fun Mortgage() {
    setAmount(100000.0f)
    setYears(30)
    setRate(0.035f)
  }

  fun setAmount(newAmount: Float) {
    if (newAmount >= 0) amount = newAmount
  }

  fun setYears(newYears: Int) {
    if (newYears >= 0) years = newYears
  }

  fun setRate(newRate: Float) {
    if (newRate >= 0) rate = newRate
  }

  fun getAmount(): Float {
    return amount
  }

  fun getFormattedAmount(): String? {
    return MONEY.format(amount)
  }

  fun getYears(): Int {
    return years
  }

  fun getRate(): Float {
    return rate
  }

  fun monthlyPayment(): Float {
    val mRate = rate / 12 // monthly interest rate
    val temp = Math.pow((1 / (1 + mRate)).toDouble(), (years *
12).toDouble())
    return amount * mRate / (1 - temp).toFloat()
```

```
  }

  fun formattedMonthlyPayment(): String? {
    return MONEY.format(monthlyPayment())
  }

  fun totalPayment(): Float {
    return monthlyPayment() * years * 12
  }

  fun formattedTotalPayment(): String? {
    return MONEY.format(totalPayment())
  }

}
```

Grading submission

1. Submit the project in a zip file (extension zip)
2. A PDF document that contains the code of all the files
3. A Zoom link or YouTube that demonstrates the features of the Mortgage Calculator app (less than 4 minutes). Write the link in the comment section in the Dropbox.
4. Write the full names of the team members and the completion of each team member. All the team members must agree to the completion of each team member.

Note:
The app is running correctly and met all the requirements is 10 points, but
Incomplete item 1(-40 points)
Incomplete item 2 (-20 points)
Incomplete item 3 (-10 points)
Incomplete item 4 (-10 points)