המגישים:

קונסטנטין ויינשטיין 313881252

אורן קאיקוב 037832292

# שאלה 1

רצ"ב מסמך הקווים המנחים לקידוד באנגלית.

# שאלה 2

פלט עבור The Little Engine that could:

flesch: 91.93720730397423

פלט עבור grapeswrath00cablgoog:

flesch: 83.46207884059234

# שאלה 3

הפלט של Wallet.main():

wallet contains coin of value 10: false

wallet contains coin of value 10: true

wallet total value: 16.6

wallet size: 5

wallet pay minimum: 15.0

wallet pay : 0.6

ב1. מאחר ופסקת ה-@effects של המפרט היא:

*tries to match at least the sum "sum" with the minimum number of coins available from the wallet […]*

ניתן ליצור רשימת מטבעות חדשה ממוינת בסדר יורד, לרוץ עליה עד שנגיע לסכום כסף גבוה או שווה לפרמטר *sum* ולהוריד מהרשימה המקורית *coinList* את כל המטבעות שעברנו עליהם.

ב2. המפרט של *payMinimum* **חזק** יותר מהמפרט של *pay* מאחר וכל המימוש של *payMinimum* יספק גם את המפרט של *pay*.

ב3. להלן המימוש שאנו מציעים (מופיע גל בקובץ Wallet.java):

*/\*\*  
 \** ***@modifies*** *this  
 \** ***@effects*** *tries to match at least the sum "sum" with the minimum number of coins available from the wallet.  
 \* If transaction is possible, removes the paid coins from the wallet; else; changes nothing  
 \** ***@return*** *the amount actually paid, 0 if amount could not be obtained  
 \*/***public double** payMinimum(**double** sum) {  
 **double** currentSum = 0;  
 List<Coin> sortedCoinList = **new** ArrayList<>();  
 sortedCoinList.addAll(**coinList**);  
 sortedCoinList.sort(Comparator.*comparing*(Coin::getValue).reversed());  
 List<Coin> coinsToRemove = **new** ArrayList<>();  
  
 **for** (Coin coin : sortedCoinList){  
 currentSum += coin.getValue();  
 coinsToRemove.add(coin);  
 **if** (currentSum >= sum) **break**;  
 }  
  
 **this**.**coinList**.removeAll(coinsToRemove);  
 **if** (currentSum >= sum){  
 **return** currentSum;  
 }  
 **return** 0;  
}

ג1. מאחר והמתודה *pay* לפי המפרט הקודם שלמה, כלומר תומכת בכל תחום הפרמטרים sum, המימוש הקודם שלה יספק גם את המפרט אם ה-precondition החדש sum < total money in wallet. ניתן, אך לא חובה, להוסיף בתחילת המתודה בדיקה אם אכן sum קטן מכמות הכסף ב-Wallet, ואם לא – לזרוק exception.

ג2. המפרט החדש **חלש יותר** מהמפרט הקודם מאחר ופסקת ה-@requires במפרט דורשת יותר תנאים מקדימים מהמשתמש.

ג3. עדיף לציין בפסקת ה-effects וה-returns, מה קורה, כאשר התנאי sum < total money in wallet אינו מתקיים על מנת שההתנהגות תהיה מוגדרת היטב.

# שאלה 4

רצ"ב (בחלק הרטוב) קובץ CoinCollection והקובץ CoinCollectionTb שבודק את CoinCollection.