

הצעה לסיור לימודי

רוני רובן ואודיה וסרלאוף

האתר: מרכז המבקרים הלימודי של בנק ישראל, תל אביב, רחוב לילנבלום 37.
כניסה מגיל 8, חינם.

תערוכת קבע – תערוכה נומיסמטית – "התפתחות הכסף".

[להזמנת ביקור \(בתיאום מראש בלבד\)](#)

מיקום	נושא	תאור הפעילות
פעילות 1	פעולות	נרצה לדמות את הפעולות שמתבצעות החשבון בנק, פעולת הפקדה: <code>deposit(String accountHolder, double amount)</code> פעולת משיכה: <code>withdraw(String accountHolder, double amount)</code> פעולה המציגה את מצב העו"ש: <code>getBalance(String accountHolder)</code>
פעילות 2	מערך חד ממדי	פעולת הצגת פרטי חשבונות: <code>displayAllBalances(String[] accountHolders)</code> פעולת הצפנת מס' חשבון: <code>encryptAccountNumber(String accountNumber)</code>
פעילות 3	מחלקות ועצמים	יצירת מחלקת "חשבון בנק", הגדרת בנאי ופעולות על העצמים בתוכנית ראשית שתדמה פעולות בנק



הנחיות פעילות מס' 1

מטרה : ליצור תוכנית ב- Java שמדמה מעקב עסקאות בנקאיות פשוטות בתוך המחלקה (BankAccount)

דרישות:

1. לממש את הפעולות הבאות:

- **פעולת הפקדה** – מקבלת את שם בעל החשבון ואת הסכום להפקדה, הפעולה מוסיפה את סכום ההפקדה לחשבון הנתון
deposit(String accountHolder, double amount)
- **פעולת משיכה** – מקבלת את שם בעל החשבון ואת הסכום למשיכה, הפעולה מעדכנת את סכום המשיכה מחשבון הנתון
withdraw(String accountHolder, double amount)
- פעולת מצב חשבון - מחזירה את היתרה הנוכחית של בעל החשבון המסוים
getBalance(String accountHolder)

פתרון

```
/**
 * deposit
 * @param amount
 */
public void deposit(double amount) {
    this.balance += amount;
}

/**
 * withdraw
 * @param amount
 */
// Withdraw method
public void withdraw(double amount) {
    if (balance >= amount) {
        this.balance -= amount;
    } else {
        System.out.println("Insufficient funds!");
    }
}

// Get balance method
public double getBalance() {
    return this.balance;
}
```

הנחיות פעילות מס' 2

מטרה: יצירת מחלקה ראשית ובה נגדיר מס' עצמים של חשבונות ופעולות על העצמים

דרישות:

1. הגדרת המחלקה הראשית **BankTransaction**

- נגדיר 3 חשבונות ונייצר מערך שיאגד את החשבונות
- נגדיר פעולה חיצונית שתקבל מערך של חשבונות ותדפיס את מצב החשבונות

2. פעולה להצגת מצב החשבונות

displayAllBalances()

פתרון:

```
public class BankTransaction {
    public static void main(String[] args) {
        // Create an array of BankAccount objects
        BankAccount[] accounts = new BankAccount[3];

        // Initialize accounts
        accounts[0] = new BankAccount(1, "Messi", 1000.0);
        accounts[1] = new BankAccount(2, "Lionel", 500.0);
        accounts[2] = new BankAccount(3, "Di Maria", 1200.0);

        public class BankTransaction {
            public static void main(String[] args) {
                // Create an array of BankAccount objects
                BankAccount[] accounts = new BankAccount[3];

                // Initialize accounts
                accounts[0] = new BankAccount(1, "Messi", 1000.0);
                accounts[1] = new BankAccount(2, "Lionel", 500.0);
                accounts[2] = new BankAccount(3, "Di Maria", 1200.0);
            }

            public static void displayAllBalances(BankAccount[] accounts){
                for (int i = 0; i < accounts.length; i++) {
                    System.out.println("account holder: "+accounts[i].getAccountHolder()+"\tbalance:
                    "+accounts[i].getBalance());
                }
            }
        }
    }
}
```

הנחיות פעילות מס' 3

מטרה: יצירת מחלקת חשבון בנק

דרישות:

1. הגדרת תכונות המחלקה:

- **accountNumber** - מס' חשבון, מספר שלם
- **accountHolder** - שם בעל החשבון, מחרוזת
- **balance** – מצב החשבון, מספר ממשי

2. הגדרת פעולה בונה

פתרון:

```
public class BankAccount {
    private int accountNumber;
    private String accountHolder;
    private double balance;

    //constructor
    public BankAccount(int accountNumber, String accountHolder, double balance) {
        this.accountNumber = accountNumber;
        this.accountHolder = accountHolder;
        this.balance = balance;
    }

    //getters and setters
    public int getAccountNumber() {
        return accountNumber;
    }

    public void setAccountNumber(int accountNumber) {
        this.accountNumber = accountNumber;
    }

    public String getAccountHolder() {
        return accountHolder;
    }

    public void setAccountHolder(String accountHolder) {
        this.accountHolder = accountHolder;
    }

    public double getBalance() {
        return balance;
    }

    public void setBalance(double balance) {
        this.balance = balance;
    }
}
```