

תוכנה 1

סמסטר א' תשע"א

תרגול מס' 6

מנשקים, דיאגרמות וביטים*
רובי בוים ומתי שמרת

* ליח' בוטריח בסדר החד

General Tips on Programming

- Write your code modularly
 - top-down approach
- Compile + test functionality "on the fly"
 - Start with an "empty" program/classes
 - Add content gradually and keep testing
 - If something goes wrong, probably the bug is in the latest change...

General Tips on Programming

- Use the Debugger to follow your execution flow and find what went wrong
- Understanding is good but not enough – you must practice!

Even More Tips

- Based on a true story:
 - Do not send us code
 - Do not send us emails saying "the code you gave us does not work", before you make sure the **original** given code is problematic
 - When you ask questions be as specific as you can, give as much information about the problem and your trials as possible

Even More Tips

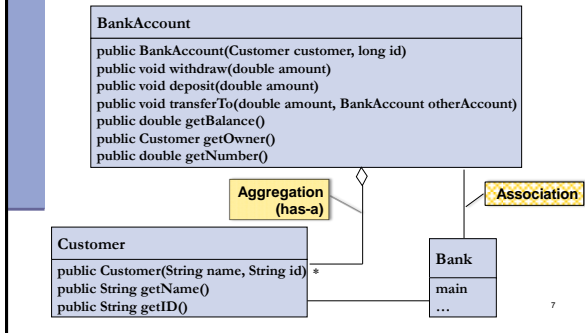
- Late submissions / appeals on homework grades?
- Please contact the graders
 - odedelba@post.tau.ac.il

המערכת הבנקאית

- נתאר את מערכת התוכנה שלנו בעזרת דיאגרמות דיאגרמות סטטיות:
- תיאור היחסים בין המחלקות השונות במערכת דיאגרמות דינאמיות:
- תיאור ההתנהגות של המערכת בזמן ריצה
 - מצב האובייקטים
 - תיאור של תרחיש



Class Diagram



המחלקה Customer

```

public class Customer {
    public Customer(String name, String id) {
        this.name = name;
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public String getID() {
        return id;
    }

    private String name;
    private String id;
}
    
```

Toy Bank Program

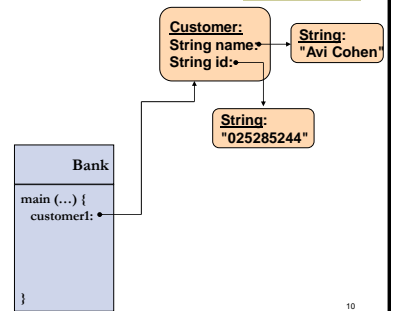
```

public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");
        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer1, 2984);

        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(100, account3);
        account2.withdraw(300);

        System.out.println("account1 has " + account1.getBalance());
        System.out.println("account2 has " + account2.getBalance());
    }
}
    
```

Object Diagram



Toy Bank Program

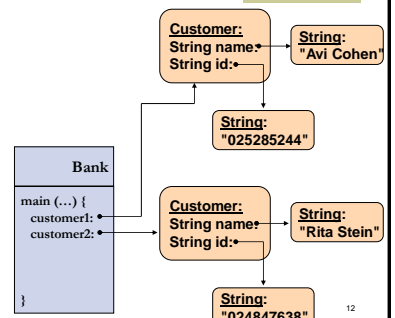
```

public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");
        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer1, 2984);

        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(100, account3);
        account2.withdraw(300);

        System.out.println("account1 has " + account1.getBalance());
        System.out.println("account2 has " + account2.getBalance());
    }
}
    
```

Object Diagram



Toy Bank Program

```
public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");

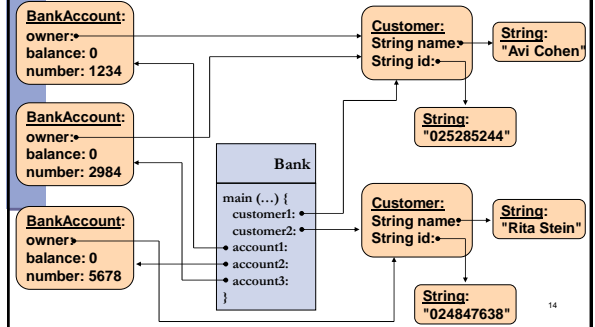
        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer1, 2984);

        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(100, account3);
        account2.withdraw(300);

        System.out.println("account1 has " + account1.getBalance());
        System.out.println("account2 has " + account2.getBalance());
    }
}
```

13

Object Diagram



14

Message Sequence Chart

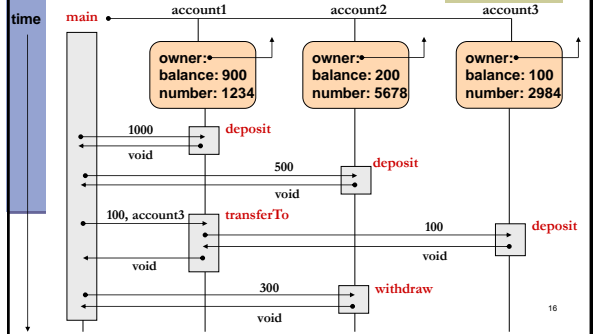
```
public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");
        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer1, 2984);

        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(100, account3);
        account2.withdraw(300);

        System.out.println("account1 has " + account1.getBalance());
        System.out.println("account2 has " + account2.getBalance());
    }
}
```

15

Message Sequence Chart



16

Output

```
public class Bank {
    public static void main(String[] args) {
        Customer customer1 = new Customer("Avi Cohen", "025285244");
        Customer customer2 = new Customer("Rita Stein", "024847638");
        BankAccount account1 = new BankAccount(customer1, 1234);
        BankAccount account2 = new BankAccount(customer2, 5678);
        BankAccount account3 = new BankAccount(customer1, 2984);

        account1.deposit(1000);
        account2.deposit(500);
        account1.transferTo(100, account3);
        account2.withdraw(300);

        System.out.println("account1 has " + account1.getBalance());
        System.out.println("account2 has " + account2.getBalance());
    }
}
```

output: account1 has 900.0
account2 has 200.0

17

מנשקים

- מנשק (interface) הוא מבנה תחבירי ב Java המאפשר לחסוך בבוד לוקו
- קוד אשר משתמש במנשק יוכל בזמן ריצה לעבוד עם מגוון מחלקות המממשות את המנשק הזה (ללא צורך בשכפול הקוד עבור כל מחלקה)
- דוגמא: נגן מוזיקה אשר מותאם לעבוד עם קובצי מוזיקה (mp3) ועם קובצי וידאו (mp4)

Playing Mp3

```
public class MP3Song {
    public void play() {
        // audio codec calculations,
        // play the song...
    }

    // does complicated stuff
    // related to MP3 format...
}

public class Player {
    private boolean repeat;
    private boolean shuffle;

    public void playSongs(MP3Song[] songs) {
        do {
            if (shuffle)
                Collections.shuffle(Arrays.asList(songs));

            for (MP3Song song : songs)
                song.play();
        } while (repeat);
    }
}
```

Playing VideoClips

```
public class VideoClip {
    public void play() {
        // video codec calculations,
        // play the clip ...
    }

    // does complicated stuff
    // related to MP4 format ...
}

public class Player {
    // same as before...
    public void playVideos(VideoClip[] clips) {
        do {
            if (shuffle)
                Collections.shuffle(Arrays.asList(clips));

            for (VideoClip videoClip : clips)
                videoClip.play();
        } while (repeat);
    }
}
```

שכפול קוד

```
public void playSongs(MP3Song[] songs) {
    do {
        if (shuffle)
            Collections.shuffle(Arrays.asList(songs));

        for (MP3Song song : songs)
            song.play();
    } while (repeat);
}

public void playVideos(VideoClip[] clips) {
    do {
        if (shuffle)
            Collections.shuffle(Arrays.asList(clips));

        for (VideoClip videoClip : clips)
            videoClip.play();
    } while (repeat);
}
```

למרות ששני השרתים נקראים `play()`
אלו פונקציות שונות!

נרצה למזג את שני קטעי הקוד

שימוש בממשק

```
public void play (Playable[] items) {
    do {
        if (shuffle)
            Collections.shuffle(Arrays.asList(items));

        for (Playable item : items)
            item.play();
    } while (repeat);
}

public interface Playable {
    public void play();
}
```

מימוש הממשק ע"י הספקים

```
public class VideoClip implements Playable {
    @Override
    public void play() {
        // render video, play the clip on screen...
    }

    // does complicated stuff related to video formats...
}

public class MP3Song implements Playable {
    @Override
    public void play() {
        // audio codec calculations, play the song...
    }

    // does complicated stuff related to MP3 format...
}
```

מערכים פולימורפים

```
Playable[] playables = new Playable[3];

playables[0] = new MP3Song();
playables[1] = new VideoClip();
playables[2] = new MP4Song(); // new Playable class

Player player = new Player();
// init player...
player.play(playables);

public void play (Playable [] items) {
    do {
        if (shuffle)
            Collections.shuffle(Arrays.asList(items));

        for (Playable item : items)
            item.play();
    } while (repeat);
}
```

עבור כל איבר במערך
יקרא ה `play()` המתאים

