

תוכנה 1

תרגול 5: ממשקים, סיביות, ועוד
הדס צור ואסף זריצקי

ממשקים - תזכורת

- כדי לתקשר בין הספק והלקוח עליהם להגדיר **ממשק** (interface, ממשק) ביניהם
- בתהליך פיתוח תוכנה תקין, כתיבת הממשק תעשה בתחילת תהליך הפיתוח
- כל מודול מגדיר מהם השירותים שלהם הוא זקוק ממודולים אחרים ע"י ניסוח ממשק רצוי
- ממשק זה מהווה בסיס לכתיבת הקוד הן בצד הספק, שימש את הפונקציות הדרושות והן בצד הלקוח, שמשמש בפונקציות (קורא להן) ללא תלות במימוש שלהן

מאת: ד"ר דוד גורן
אנדרטת תל אביב

2

ממשקים interface

- מבנה תחבירי ב Java המאפשר לחסוך בקוד לקוח
- קוד אשר משמש בממשק יוכל בזמן ריצה לעבוד עם מגוון מחלקות המממשות את הממשק הזה (ללא צורך בשכפול הקוד עבור כל מחלקה)
- דוגמא: נגן מוזיקה אשר מותאם לעבודה עם קובצי מוזיקה (mp3) ועם קובצי וידאו (mp4)

תחביר

```
Declaration { modifiers public interface Colors {  
  Body {  
    public static final int RED = 1;  
    int GREEN = 2;  
    public int foo();  
    String bar(int val);  
  }  
}
```

Constants
public final static
by default

Methods
No implementation (abstract)
public by default

- ממשק מגדיר טיפוס הפניה חדש (reference type)
- בזמן ריצה האובייקט המוצע הוא ממחלקה המממשת את הממשק

4

תחביר ספק/לקוח

```
public class MyClass implements Colors {  
  public int foo() { ... }  
  public String bar(int val) { ... }  
  public void moreFunc() { ... }  
}  
  
public static void main(String[] args) {  
  MyClass cls = new MyClass();  
  Colors col = cls;  
}
```

ספק

הספק מחייב לממש את כל הפונקציות שהוגדרו בממשק. הוא יכול להגדיר מתודות נוספות.

לקוח

הלקוח יכול לפנות לאובייקטים בעזרת טיפוס המחלקה או בעזרת טיפוס הממשק.

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);  
  }  
}
```

5

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);
}
```

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

```
public void playVideos(VideoClip[] clips) {
    do {
        if (shuffle)
            Collections.shuffle(Arrays.asList(clips));
        for (VideoClip videoClip : clips)
            videoClip.play();
    } while (repeat);
}
```

נרצה למזג את שני קטעי הקוד

הגדרת הממשק

שני קטעי הקוד עושים שימוש ב"משהו שאפשר לנגן אותו"

נגדיר ממשק `Playable`

כל מחלקה שממשת טיפוס שאפשר לנגן אותו תממש את הממשק

דומה להגדרת מחלקה

נגדיר מתודה יחידה

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

שימוש בממשק (Player)

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

        for (Playable item : items)
            item.play();

    } while (repeat);
}
```

9

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

פעולות על סיביות

אופרטורים לביצוע פעולות על ביטים

- רק על טיפוסים איטגרליים (int, short, byte, char)

~	Unary bitwise complement
<<	Signed left shift
>>	Signed right shift
>>>	Unsigned right shift
&	Bitwise AND
^	Bitwise XOR
	Bitwise OR

13

פעולות על סיביות - דוגמאות

32 int ביטים ■

י י צ ו ג ב י נ א ר י

```

      3       0000000000000000000000000000000000000011
     ~3      1111111111111111111111111111111111111100
    -3      1111111111111111111111111111111111111101
   3 << 2    0000000000000000000000000000000000000100
  -3 >> 1    1111111111111111111111111111111111111110
-3 >>> 1    0111111111111111111111111111111111111110
```

מה נקבל מ 3 & i?

■ שני הביטים הימניים של i

■ ומה נקבל מ `0xF0 & i` ?

14

שימוש

■ נתחיל במספר (או מספרים) שעבורו אנו יודעים את הייצוג הבינארי

[illegible][illegible]

■ בעזרת הפעולות שתוארו נוכל להגיע לכל סידור ביטים רצוי

■ בהקשר לתרגיל 6, איך לייצר IPAddress באמצעות int אחד?

15

Casting (המרה)

- Sometimes it is desirable for a variable of one type to be considered as belonging to another in an operation
- We say the variable is *cast* to the new type
- Syntax: *(type)operation/variable*

16

Example – find what's wrong

```
int a = 10;
int b = 20;
System.out.println("average=" + (a+b)*(1/2));
```

17

Alternatives

```
int a = 10;
int b = 20;

System.out.println("average = " + (a+b)*((double)1/2));
System.out.println("average = " + (a+b)*(1.0/2));

// will the following work?
System.out.println("average = " + (a+b)*(double) (1/2));
```

18

Implicit Cast

- Automatic conversion between data types
- Occurs when data types can not be compared as is

```
int x, y;  
double d;  
x = 3; y = 5;  
d = x + y; // d gets 8.0000
```

19

Implicit Cast (Cont.)

```
int a = 4.5; // implicit cast from double (4.5) to int (4),  
             compilation error
```

```
double x = 4; // implicit cast from int to double, "int is a  
              double" relation hold
```

20

static field

- שדה המוגדר static (class variable) אינו משויך לאובייקט מסוים
- קיימת בדיוק התגלמות אחת שלו
- אם שדה אינו מוגדר static (instance variable) בכל פעם שנוצר אובייקט חדש נוצר משתנה חדש המשוך לאובייקט זה.

21

static method

- מתודה סטטית (class method) אינה משויכת לאובייקט
- אין אובייקט נוכחי (this)
- מתודה שאינה סטטית (instance method) נקראת תמיד תוך התייחסות לאובייקט מסוים. אובייקט זה הוא האובייקט הנוכחי (this) במתודה
- נוכל לגשת רק לשדות שהוגדרו סטטיים

22

static example

```
class Point {  
    int x, y, useCount;  
    Point(int x, int y) { this.x = x; this.y = y; }  
    static Point origin = new Point(0, 0);  
}  
  
class Test {  
    public static void main(String[] args) {  
        Point p = new Point(1,1);  
        Point q = new Point(2,2);  
  
        p.x = 3; p.y = 3; p.useCount++;  
        p.origin.useCount++;  
  
        System.out.println("(" + q.x + ", " + q.y + "  
        System.out.println(q.useCount);  
        System.out.println(q.origin == Point.origin);  
        System.out.println(q.origin.useCount);  
    }  
}
```

גישה לשדה סטטי
יש להשתמש בשם מחלקה ולא
באובייקט

גישה לאותה הפניה – true

23

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 (or the forum) 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 grader
 - odedelba@post.tau.ac.il