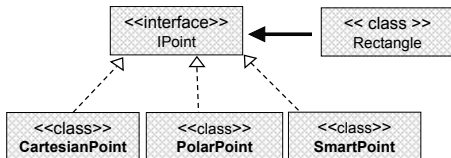


צד הלקוח

- בהרצאה ראינו את הממשק `IPoint`, והצגנו 3 מימושים שונים עבורו
- ראינו כי **לקוחות** התלויים בממשק `IPoint` בלבד, ולא מכירים את המחלקות המממשות **אדישים** לשינויים עתידיים בקוד הספק
- שימוש **בממשקים** חוסך **שכפול קוד לקוח**, בכך שאותו קטע קוד עובד בצורה נכונה עם מגוון ספקים (פולימורפיזם)



2

תוכנה 1 בשפת Java תרגול מספר 7: הורשה

בית הספר למדעי המחשב
אוניברסיטת תל אביב

מחלקות מופשטות



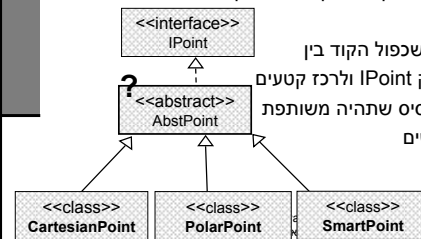
- מחלקה מופשטת מוגדרת ע"י המלה `abstract`
- לא ניתן ליצור מופע של מחלקה מופשטת (בדומה לממשק)
- יכולה לממש ממשק אך לא לממש את כל השירותים המוגדרים בו
- זהו מנגנון מועיל להימנע משכפול קוד במחלקות יורשות

תוכנה 1 בשפת Java
אוניברסיטת תל אביב

4

צד הספק

- בהרצאה האחרונה היכרנו את **מנגנון ההורשה אשר חוסך שכפול קוד בצד הספק**
- ע"י הורשה מקבלת מחלקה את קטע הקוד בירושה במקום לחזור עליו. שני הספקים חולקים אותו הקוד
- ננסה לזהות את שכפול הקוד בין 3 מממשי הממשק `IPoint` ולרכז קטעים אלה במחלקת בסיס שתהיה משותפת לשלושת המימושים



3

CartesianPoint

```

private double x;
private double y;

public CartesianPoint(double x, double y) {
    this.x = x;
    this.y = y;
}

public double x() { return x; }
public double y() { return y; }
public double rho() { return Math.sqrt(x*x + y*y); }
public double theta() { return Math.atan2(y,x); }
    
```

PolarPoint

```

private double r;
private double theta;

public PolarPoint(double r, double theta) {
    this.r = r;
    this.theta = theta;
}

public double x() { return r * Math.cos(theta); }
public double y() { return r * Math.sin(theta); }
public double rho() { return r; }
public double theta() { return theta; }
    
```

קשה לראות דמיון בין מימשי המתודות במקרה זה.
כל 4 המתודות **בסיסיות** יש להן קשר הדוק לייצוג שנבחר **לשדות**

תוכנה 1 בשפת Java
אוניברסיטת תל אביב

6

מחלקות מופשטות - דוגמה

- מחלקה פשוטה:

```

public abstract class A {
    public void f() {
        System.out.println("A.f!");
    }

    public void g();

    A a = new A();

    public class B extends A {
        public void g() {
            System.out.println("B.g!");
        }
    }
}

A a = new B();
    
```



A a = new B();

תוכנה 1 בשפת Java
אוניברסיטת תל אביב

5

CartesianPoint	PolarPoint
<pre>public double distance(IPoint other) { return Math.sqrt((x-other.x()) * (x-other.x()) + (y-other.y())*(y-other.y())); }</pre>	<pre>public double distance(IPoint other) { double deltaX = x()-other.x(); double deltaY = y()-other.y(); return Math.sqrt(deltaX*deltaX + deltaY*deltaY); }</pre>
<p>הקוד דומה אבל לא זהה, נראה מה ניתן לעשות...</p> <p>נוסח לשכתב את CartesianPoint ע"י הוספת משתני העזר ΔX ו-ΔY</p>	
<p>תוכנה 1 בשפת Java אוניברסיטת תל אביב</p>	

CartesianPoint	PolarPoint
<pre>public void rotate(double angle) { double currentTheta = Math.atan2(y,x); double currentRho = rho(); x = currentRho * Math.cos(currentTheta+angle); y = currentRho * Math.sin(currentTheta+angle); }</pre>	<pre>public void rotate(double angle) { theta += angle; }</pre>
<pre>public void translate(double dx, double dy) { x += dx; y += dy; }</pre>	<pre>public void translate(double dx, double dy) { double newX = x() + dx; double newY = y() + dy; r = Math.sqrt(newX*newX + newY*newY); theta = Math.atan2(newY, newX); }</pre>
<p>גם כאן קשה לראות דמיון בין מימושי המתודות, למימושי קשר הדוק לייצוג שנבחר לשדות</p>	
<p>תוכנה 1 בשפת Java אוניברסיטת תל אביב</p>	

CartesianPoint	PolarPoint
<pre>public double distance(IPoint other) { double deltaX = x()-other.x(); double deltaY = y()-other.y(); return Math.sqrt(deltaX * deltaX + (deltaY * deltaY)); }</pre>	<pre>public double distance(IPoint other) { double deltaX = x()-other.x(); double deltaY = y()-other.y(); return Math.sqrt(deltaX*deltaX + deltaY*deltaY); }</pre>
<p>נשאר הבדל אחד</p> <p>נחליף את x להיות $-x$ במאזן ביצועים לעומת כלליות נעדיף תמיד את הכלליות</p>	
<p>תוכנה 1 בשפת Java אוניברסיטת תל אביב</p>	

CartesianPoint	PolarPoint
<pre>public double distance(IPoint other) { double deltaX = x()-other.x(); double deltaY = y()-other.y(); return Math.sqrt((x-other.x()) * (x-other.x())+ (y-other.y())*(y-other.y())); }</pre>	<pre>public double distance(IPoint other) { double deltaX = x()-other.x(); double deltaY = y()-other.y(); return Math.sqrt(deltaX*deltaX + deltaY*deltaY); }</pre>
<p>תוכנה 1 בשפת Java אוניברסיטת תל אביב</p>	

CartesianPoint	PolarPoint
<pre>public String toString(){ return "(x=" + x() + ", y=" + y() + ", r=" + rho() + ", theta=" + theta() + ")"; }</pre>	<pre>public String toString() { return "(x=" + x() + ", y=" + y() + ", r=" + r + ", theta=" + theta + ")"; }</pre>
<p>תהליך דומה ניתן גם לבצע עבור toString</p>	
<p>תוכנה 1 בשפת Java אוניברסיטת תל אביב</p>	

CartesianPoint	PolarPoint
<pre>public double distance(IPoint other) { double deltaX = x()-other.x(); double deltaY = y()-other.y(); return Math.sqrt(deltaX * deltaX + (deltaY * deltaY)); }</pre>	<pre>public double distance(IPoint other) { double deltaX = x()-other.x(); double deltaY = y()-other.y(); return Math.sqrt(deltaX*deltaX + deltaY*deltaY); }</pre>
<p>שתי המתודות זהות לחלוטין. ניתן להעביר את המתודה למחלקה AbstPoint וילחוק אותה מהמחלקות CartesianPoint ו-PolarPoint</p>	
<p>תוכנה 1 בשפת Java אוניברסיטת תל אביב</p>	

אתחולים ובנאים

Extract Superclass Refactoring



ניתן לבצע תהליך זה בצורה אוטומטית ע"י שכתוב מבני (Refactoring) שנקרא: Extract Superclass

הגרסה ב-Eclipse עוד לא "מושלמת"

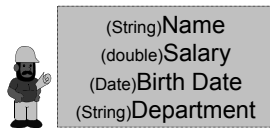
תוכנה 1 בשפת Java
אוניברסיטת תל אביב

13

מה הסדר ביצירת מופע של מחלקה?

שלב ראשון: הקצאת זיכרון לשדות העצם והצבת ערכי ברירת מחדל

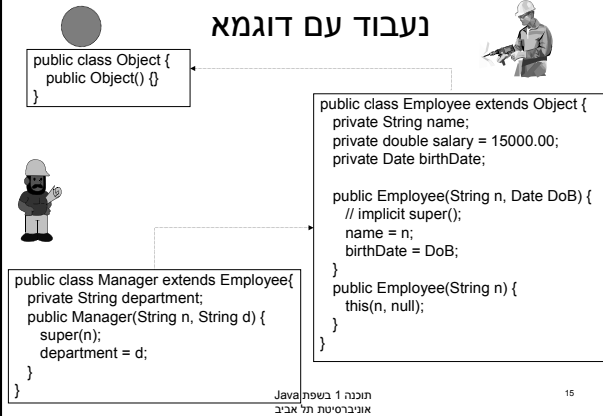
Manager m = new Manager("Joe Smith", "Sales");



תוכנה 1 בשפת Java
אוניברסיטת תל אביב

16

נעבוד עם דוגמא



תוכנה 1 בשפת Java
אוניברסיטת תל אביב

15

בואו נעקוב אחרי יצירת האובייקט

- Basic initialization
 - Allocate memory for the complete Manager object
 - Initialize all instance variables to their default values
- Call constructor: Manager("Joe Smith", "Sales")
 - Bind constructor parameters: n="Joe Smith", d="Sales"
 - No explicit this() call
 - Call super(n) for Employee(String)
 - Bind constructor parameters: n="Joe Smith", DoB=null
 - No explicit this() call
 - Call super() for Object()
 - No binding necessary
 - No this() call
 - No super() call (Object is the root)
 - No explicit variable initialization for Object
 - Initialize explicit Employee variables: salary=15000.00;
 - Execute body: name="Joe Smith", date=null;
 - Steps skipped
 - Execute body: No body in Employee(String)
 - No explicit initializers for Manager
 - Execute body: department="Sales"

```
public class Manager extends Employee {
    private String department;
    public Manager(String n, String d) {
        super(n);
        department = d;
    }
}

public class Employee extends Object {
    private String name;
    private double salary = 15000.00;
    private Date birthDate;

    public Employee(String n, Date DoB) {
        // implicit super();
        name = n;
        birthDate = DoB;
    }

    public Employee(String n) {
        this(n, null);
    }
}
```

תוכנה 1 בשפת Java
אוניברסיטת תל אביב

18

מה הסדר ביצירת מופע של מחלקה?

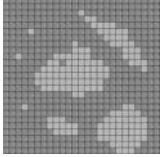
- שלב שני: נקרא הבנאי הנוכחי והאלגוריתם הבא מופעל:
- Bind constructor parameters.
 - If explicit this(), call recursively, and then skip to Step 5.
 - Call recursively the implicit or explicit super(...), except for Object because Object has no parent class.
 - Execute the explicit instance variable initializers.
 - Execute the body of the current constructor.

תוכנה 1 בשפת Java
אוניברסיטת תל אביב

17

הפרויקט בגדול

■ שמות עצם בהגדרה



העולם



חיה – אוכלי עשב וטורפים

תוכנה 1 בשפת Java
אוניברסיטת תל אביב

20

Exercise 7

טורפים ואוכלי עשב



החלפת קבצים בינאריים

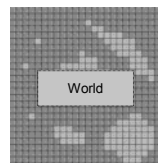
1. השלימו את הפרויקט לפי ההוראות עד סעיף 4
2. בחרו ב-Package Explorer את המחלקות שאתם רוצים להעביר לסטודנט אחר



תוכנה 1 בשפת Java
אוניברסיטת תל אביב

22

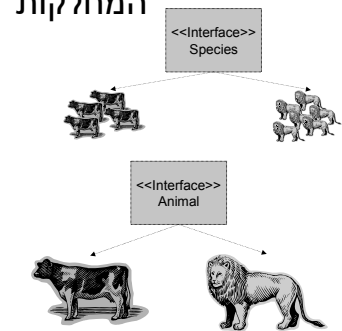
המחלקות



World

PatchState

Action

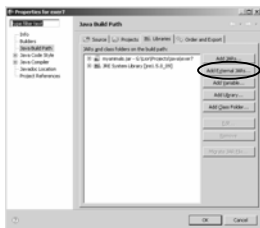


תוכנה 1 בשפת Java
אוניברסיטת תל אביב

21

החלפת קבצים בינאריים

- העתיקו את קובץ ה-jar שקיבלתם מסטודנט אחר אל ספריית הפרויקט שלכם
- בחרו ב-eclipse Project → properties
- הוסיפו את קובץ ה-jar המתאים
- הוסיפו קריאות ליצירת המיין החדש
- הריצו את הפרויקט

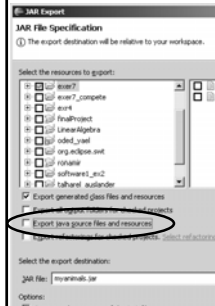


תוכנה 1 בשפת Java
אוניברסיטת תל אביב

24

החלפת קבצים בינאריים

1. השלימו את הפרויקט לפי ההוראות עד סעיף 4
2. בחרו ב-Package Explorer את המחלקות שאתם רוצים להעביר לסטודנט אחר
3. לחצו כפתור ימני ו-export (או מתפריט file)
4. ייצאו את הקבצים (ללא קבצי מקור)



תוכנה 1 בשפת Java
אוניברסיטת תל אביב

23