

תוכנה 1 בשפת Java
שיעור מספר 14:
"מה? כבר נגמר?"

סיון טולדו אוהד ברזילי

בית הספר למדעי המחשב
אוניברסיטת תל אביב

תכנות מונחה עצמים

- למדנו בקורס את עקרונות התכנות מונחה העצמים כפי שהם מתבטאים בשפת Java
- לאחר שסיימנו את לימוד השפה זה הזמן לחזור ולבדוק אילו עקרונות למדנו, ואיך הם מתבטאים בשפת התכנות
- כפי שנראה מיד – יש רכיבים בשפה המשרתים יותר מעקרון אחד וכן יש חפיפה בין העקרונות השונים

הפשטה

- מאפשרת להתעלות מעל לפרטי ייצוג או מימוש:
 - הפשטה ברמת השרות (procedural abstraction)
 - הפשטת נתונים (data abstraction)
 - המחלקה מייצגת טיפוס בעולם הבעיה
 - המנשק הוא רכיב בשפת התכנות
 - הפולימורפיזם ב Java מאפשר הבדל בין הטיפוס הסטטי והדינאמי
 - הפשטת טיפוסים (type abstraction)
 - Generics
 - הפשטת ביצוע סדרתי
 - Iterators

שימוש חוזר

- שכפול קוד <= שרות
- שכפול קוד עם הבדלים קלים <= שרות עם פרמטרים
- כתיבת קוד חדש בעזרת קוד קיים
 - הרחבת מחלקות ע"י הרכבה (קוביות לגו)
 - הרחבת מחלקות ע"י הורשה
- מנשקים – קוד לקוח מותאם לעבודה עם מגוון ספקים
- Generics – קוד ספק אשר הלקוח יכול לבצע לו פרמטריזציה
- תבניות תיכון

מודולריות

■ שמירה על מודולריות תקל בעתיד על שימוש חוזר במודול בשלמותו

■ המחלקה כמודול

■ עקרון הסגור-פתוח

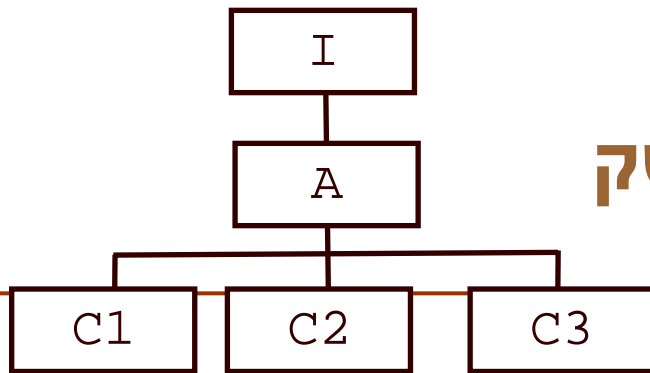
■ סגור – מוכן לשימוש

■ פתוח – קל להרחבה

הכמסה והסתרת מידע

- עקרונות המסייעים להקטנת הצימוד בין מודולים
- Java מספקת 4 דרגות נראות לשרותים ונתונים
 - מוטיבציה אחת: אכיפת החוזה
 - מוטיבציה שניה: גמישות עתידית
- Java מאפשרת למחלקה לממש כמה מנשקים שונים
- הפולימורפיזם ב Java מאפשר הבדל בין הטיפוס הסטטי והדינאמי

מחלקה מופשטת לעומת מנשק



- מנשק יוצר חסכון בקוד לקוח
 - הקוד נכתב פעם אחת אבל יכול להשתמש במגוון ספקים (כולל כאלה שעוד לא נכתבו!)
- מחלקה מופשטת יוצרת חסכון בקוד ספק
 - שתי מחלקות אשר מכילות קוד דומה יכולות להעביר את הקוד הדומה למחלקת בסיס משותפת, גם אם מחלקת הבסיס לא מייצגת טיפוס שלם
- מכיוון שגם ירושת מחלקות וגם מימוש מנשקים משתמשים באותם מנגנוני שפה (פולימורפיזם) ניתן להשתמש במחלקה מופשטת גם כדי ליצור חסכון אצל לקוחות המחלקה
- **שימוש זה בעייתי** מכיוון מחלקה יורשת ממחלקה אחת בדיוק – ולכן לא ניתן להשתמש בכך עבור יותר מ"היבט" אחד של המחלקה

Polymorphism לעומת Generics

■ אם ניתן להגדיר מבנה נתונים של Object למה צריך טיפוסים מוכללים?

■ מאחורי הקלעים זה מה שנוצר

■ פרט לחשיבות המתודולוגית של הפשטת טיפוסים, יש חשיבות לגילוי השגיאות בזמן קומפילציה ולא בזמן ריצה

מהלך הקורס

- העקרונות נלמדו ביחד עם השפה:
- תכנות ב Java + תכנות מונחה עצמים
- מוטיב מארגן – ספק, לקוח והיחסים ביניהם

מהלך הקורס

- שיעור 1 – משתנים, הטיפוסים היסודיים, ביטויים ואופרטורים, הכרת טיפוס המחרוזת והמערך



מהלך הקורס

■ שיעור 2 – מבני בקרה, זימון והגדרת שרותים, העמסת שרותים

■ שיעור 3 –

■ מודל הזיכרון של Java – Heap and Stack

■ המחלקה כמודול

■ תיכון בעזרת חוזה

■ שיעור 4 –

■ הוכחת נכונות – של שרות, של מחלקה

■ המחלקה כטיפוס נתונים

■ מצב מופשט

מהלך הקורס

■ שיעור 5 –

■ מודל הזיכרון של Java – זימון שרותי מופע

■ מנשקים

■ שיעור 6 –

■ פולימורפיזם, מפעלים

■ טיפוסים מוכללים, מחלקות פנימיות ואיטרטורים

■ שיעור 7 –

■ יחסים בין מחלקות והורשה

מהלך הקורס

■ שיעור 8 – חריגים

■ שיעור 9 – הורשה וחוזה, הורשה ו Generics, הורשה מרובה

■ שיעור 10 – מנשקי משתמש גרפים

■ שיעור 11 – מנגנוני השפה:

■ שיגור דינאמי

■ איסוף זבל

■ טעינת מחלקות והשתקפות

מהלך הקורס

■ שיעור 12 – נושאים בהנדסת תוכנה:

- בדיקות תוכנה
- מחזור החיים של תוכנה
- סקירה (חלקית) של תבניות עיצוב קלאסיות
- בעיית חתכי רוחב במערכות תוכנה
- שכתוב מבני בהקשר רחב

■ שיעור 13 –

- תקשורת ב Java
- התקשורת כחתך רוחב
- שרתי אינטרנט כ framework

■ שיעור 14 – סיכום

לאן מכאן?

הבחינה

14:00 , 11/02/2007 מועד א' ■

09:00 , 11/09/2007 מועד ב' ■

3 שעות ■

חומר סגור ■

נושאי הבחינה – כל חומר הקורס ■

הרצאות – לא כולל Threads ■

תרגולים ■

תרגילים ■

חומר עזר מאתר הקורס ■

הבחינה

■ 4 שאלות (פלוס-מינוס) – מרובות סעיפים

■ סוגי שאלות:

■ שאלות תכנות

■ עיצוב בעזרת חוזה

■ מצב מופשט

■ בקיאות בשפת התכנות

■ עיצוב – שכתוב של קוד, עמידה על יתרונות/חסרונות מימושים

■ אתר הכנה לבחינה כולל שאלות דוגמא מהסוגים השונים ואוסף בחינות קודמות (+ פתרונות יתווספו מאוחר יותר)



בהצלחה