

מטרת השיעור

- חזרה על החומר שלמדנו עד כה
- חידוד מונחים תיאורטיים
- יישום בפועל של החומר ע"י פרויקט קטן

2

שיעור חזרה אי

הרצאה מספר 12

1

הפרויקט: אלגברה נמרית למתחילים

כיצד ניגש לביצוע של פרויקט שכזה?
על מה צריך לחשוב קודם?



4

הפרויקט

אני מעוניין שתפתחו עבורי ספרייה ומנשק גרפי התומכים בעבודה באלגברה לינארית: מטריצות, וקטורים, חיבור מטריצות, חיסור מטריצות, הוספת וקטורים, נורמות, הכפלה בסקלרים, מטריצות אלכסוניות, נרמול וקטורים...

הלקוח



אתם (התכניתנים)



3

זיהוי העצמים במערכת

מי המחלקות? מי העצמים? אילו שירותים נגדיר?



6

הפרויקט: אלגברה נמרית למתחילים

- עלינו לממש תוכנית המאפשרת ביצוע פעולות אלגברה נומרית, התכנית תכלול שני חלקים
 - ספריית אלגברה נומרית המגדירה מטריצות, וקטורים ופעולות עליהם
 - מנשק גרפי המאפשר לפתור בעיות באלגברה נומרית
- מטרות התכנית
 - מימוש יעיל וקל לספרייה
 - מנשק גרפי נוח לשימוש ומהיר

5

זיהוי העצמים במערכת

- סקלרים
 - חיבור, חיסור, כפל, חילוק
- וקטורים
 - הכפלה בסקלר, נורמה, הוספה
 - וחיבור וקטורים, מכפלה פנימית ומכפלה חיצונית
- מטריצות
 - כפל מטריצות, כפל בוקטורים, כפל בסקלר

7

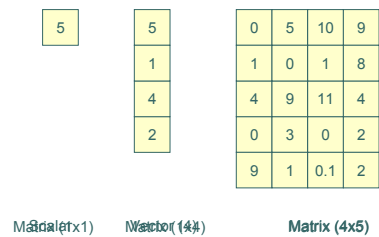
מחלקות, מנשקים, עצמים, שירותים

- סקלרים
 - חיבור, חיסור, כפל, חילוק
- וקטורים
 - הכפלה בסקלר, נורמה, הוספה
 - וחיבור וקטורים, מכפלה פנימית ומכפלה חיצונית
- מטריצות
 - כפל מטריצות, כפל בוקטורים, כפל בסקלר

8

מחלקות, מנשקים, עצמים, שירותים

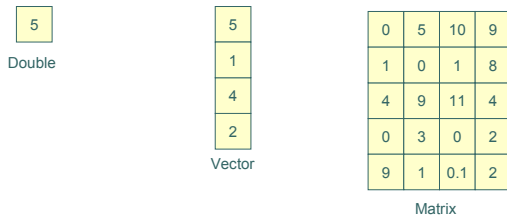
○ אופציה מספר 1 – כל העולם מטריצה?



9

מחלקות, מנשקים, עצמים, שירותים

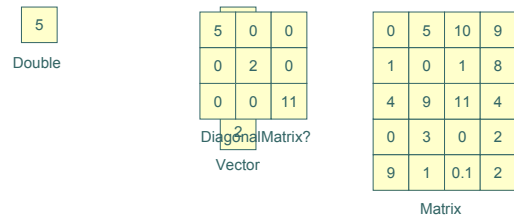
○ אופציה מספר 2 – הפרדה



10

מחלקות, מנשקים, עצמים, שירותים

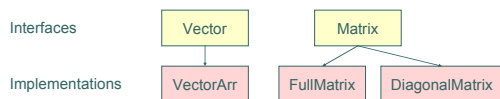
○ אופציה מספר 2 – הפרדה



11

מחלקות, מנשקים, עצמים, שירותים

○ ההצעה שלי



מה היתרונות? מה החסרונות?



12

איך נתמודד עם שגיאות?

- חריגים (exceptions) ?
- שאליות ?
- נסמך על תנאי הקדם והאחר?



14

וקטור

```
public interface Vector {  
    int dimensions();  
    boolean isNormalized();  
    double element(int index);  
    void setElement(int index, double value);  
    void setFromArray(double[] arr);  
    Vector multiply(double scalar);  
    double dotproduct(Vector v);  
    Matrix crossproduct(Vector v);  
    Vector add(Vector v);  
    Vector normalize();  
}
```

13

וקטור

```
public interface Vector {  
    int dimensions();  
    boolean isNormalized();  
    double element(int index) throws DimensionMismatchException;  
    void setElement(int index, double value) throws DimensionMismatchException;  
    void setFromArray(double[] arr) throws DimensionMismatchException;  
    Vector multiply(double scalar);  
    double dotproduct(Vector v) throws DimensionMismatchException;  
    Matrix crossproduct(Vector v) throws DimensionMismatchException;  
    Vector add(Vector v) throws DimensionMismatchException;  
    Vector normalize();  
}
```

16

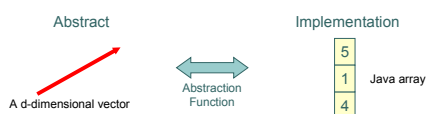
חריגים: DimensionMismatchException

```
public class DimensionMismatchException extends Exception  
{  
    DimensionMismatchException() {  
        super();  
    }  
    DimensionMismatchException(String msg) {  
        super(msg);  
    }  
}
```

15

וכעת בואו נגש למימוש...

```
public class VectorArr implements Vector {  
    private double data[];  
  
    public VectorArr(int dimensions) {  
        data = new double[dimensions];  
    }  
    ...  
}
```



18

מטריצות

```
public interface Matrix {  
    int getWidth();  
    int getHeight();  
    double element(int col, int row) throws DimensionMismatchException;  
    void setElement(int col, int row, double val) throws DimensionMismatchException;  
    void setFromArray(double[][] arr) throws DimensionMismatchException;  
    Vector multiply(Vector v) throws DimensionMismatchException;  
    Matrix multiply(Matrix m) throws DimensionMismatchException;  
    Matrix multiply(double scalar);  
}
```

17

וכעת באו נגש למימוש...

```
public class VectorArr implements Vector {
    private double data[];

    ...

    public Vector add(Vector v) throws DimensionMismatchException {
        if (v.dimensions() != dimensions()) throw new
        DimensionMismatchException();

        VectorArr newv = new VectorArr(data.length);

        for (int i=0; i<data.length; i++) {
            newv.data[i] = data[i] + v.element(i);
        }

        return newv;
    }
    ...
}
```

20

וכעת באו נגש למימוש...

```
public class VectorArr implements Vector {
    private double data[];

    public void setElement(int index, double value) throws DimensionMismatchException {
        if (index<0 || index>=data.length) throw new DimensionMismatchException();

        data[index] = value;
    }

    public void setFromArray(double[] arr) throws DimensionMismatchException {
        if (arr.length != data.length) throw new DimensionMismatchException();

        java.lang.System.arraycopy(arr, 0, data, 0, arr.length);
    }
    ...
}
```

19

ועכשיו למימוש מטריצות



- איך הייתם מממשים את המחלקה 'מטריצה'?
- מה חשוב במימוש של מטריצה?

| | | | |
|---|---|-----|---|
| 0 | 5 | 10 | 9 |
| 1 | 0 | 1 | 8 |
| 4 | 9 | 11 | 4 |
| 0 | 3 | 0 | 2 |
| 9 | 1 | 0.1 | 2 |

יניצול יעיל של הזיכרון
יגישה ישירה לכל תא
יבאו נניח שגודל מטריצה לא משתנה

22

וכעת באו נגש למימוש...

```
...
public Matrix crossproduct(Vector v) throws DimensionMismatchException {
    if (v.dimensions() != dimensions()) throw new
    DimensionMismatchException();

    Matrix m = new FullMatrix(dimensions(), dimensions());
    for (int i=0; i<data.length; i++) {
        for (int j=0; j<data.length; j++) {
            m.setElement(i, j, data[i] * v.element(j));
        }
    }

    return m;
}
...
}
```

ניתן לממש עכשיו למרות שעוד לא מימשנו את FullMatrix

21

מטריצה מלאה - כפל

- Vector multiply(Vector v) throws DimensionMismatchException;
- Matrix multiply(Matrix m) throws DimensionMismatchException;
- Matrix multiply(double scalar);

```
public Matrix multiply(double scalar) {
    FullMatrix newm = new FullMatrix(width, height);
    for (int i=0; i<width; i++) {
        for (int j=0; j<height; j++) {
            newm.data[i][j] = data[i][j] * scalar;
        }
    }
    return newm;
}
```

24

מטריצה מלאה

```
public class FullMatrix implements Matrix {

    double data[][];
    int width;
    int height;

    private boolean dimensionsValid(int col, int row) {
        return (col>=0 && col<data.length && row>=0 && row <data[0].length);
    }

    public FullMatrix(int width, int height) {
        data = new double[width][height];
        this.width = width;
        this.height = height;
    }

    public double element(int col, int row) throws DimensionMismatchException {
        if (!dimensionsValid(col, row)) throw new DimensionMismatchException();

        return data[col][row];
    }
    ...
}
```

23

מטריצה מלאה - כפל

- Vector multiply(Vector v) throws DimensionMismatchException;
- Matrix multiply(Matrix m) throws DimensionMismatchException;
- Matrix multiply(double scalar);

```
public Matrix multiply(Matrix m) throws DimensionMismatchException {
    if (m.getHeight() != width) throw new DimensionMismatchException();

    FullMatrix newm = new FullMatrix(m.getWidth(), height);
    for (int row=0; row<height; row++) {
        for (int mcol = 0; mcol<m.getWidth(); mcol++) {
            double val = 0.0;
            for (int col=0; col<width; col++) {
                val += data[col][row] * m.element(mcol, col);
            }
            newm.data[mcol][row] = val;
        }
    }
    return newm;
}
```

26

מטריצה מלאה - כפל

- Vector multiply(Vector v) throws DimensionMismatchException;
- Matrix multiply(Matrix m) throws DimensionMismatchException;
- Matrix multiply(double scalar);

```
public Vector multiply(Vector v) throws DimensionMismatchException {
    if (v.dimensions() != width) throw new DimensionMismatchException();

    Vector newv = new VectorArr(height);
    for (int i=0; i<height; i++) {
        double val = 0.0;
        for (int j=0; j<width; j++) {
            val += data[j][i] * v.element(j);
        }
        newv.setElement(i, val);
    }
    return newv;
}
```

25

מטריצה אלכסונית – שירותי מחלקה

- הרבה פעמים אנחנו זקוקים למטריצת היחידה

```
public class DiagonalMatrix implements Matrix {
    double data[];
    int size;
    ...

    public static DiagonalMatrix eye(int size) {
        DiagonalMatrix eyeMat = new DiagonalMatrix(size);
        for (int i=0; i<size; i++)
            eyeMat.data[i] = 1.0;
        return eyeMat;
    }
    ...
}
```

28

מטריצה אלכסונית

- שווה לנו לעשות מטריצה אלכסונית? מה נחסוך?



ימקום אחסון

יסדר גודל זמן באלגוריתמים שונים (כמו כפל מטריצות)

27

שימוש בספרייה

- כתבנו את הספרייה אך עוד אין ממשק
- רצוי לבדוק את הספרייה כבר בשלב הזה

```
Vector v1 = new VectorArr(3);
v1.setFromArray(new double[] {1, 0, 0});

Vector v2 = new VectorArr(3);
v2.setFromArray(new double[] {0, 1, 0.5});

System.out.println("v1=" + v1);
System.out.println("v2=" + v2);
System.out.println("v1+v2=" + v1.add(v2));
System.out.println("v1*v2=" + v1.dotProduct(v2));
System.out.println("v1*v2=" + v1.crossProduct(v2));
System.out.println("normalized v2=" + v2.normalize());
```

30

```
v1=[1.0, 0.0, 0.0]
v2=[0.0, 1.0, 0.5]
v1+v2=[1.0, 1.0, 0.5]
v1*v2=0.0
v1*v2=[0.0, 1.0, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
normalized v2=[0.0, 0.8944271909999159,
0.4472135954999579]
```

שתי שאלות מעניינות

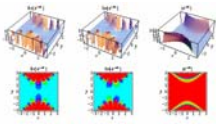
- תוצאת הכפלה של מטריצה מלאה באלכסונית היא מטריצה מלאה... האם המימושים השונים מודעים אחד לשני?

- יש לנו ממשק אחד למטריצה, ושני מימושים שונים. האם אנחנו צריכים בית חרושת?



29

מנעק גרפי



מה? מיז איך? למה?

m-by-n matrix

$a_{i,j}$

$\begin{matrix} \text{rows} \\ \downarrow \\ \begin{matrix} \alpha_{1,1} & \alpha_{1,2} & \alpha_{1,3} & \dots \\ \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} & \dots \\ \alpha_{3,1} & \alpha_{3,2} & \alpha_{3,3} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{matrix} \end{matrix}$

$\begin{matrix} \text{columns} \\ \rightarrow \\ \end{matrix}$

$\begin{matrix} \text{changes} \\ \rightarrow \\ \end{matrix}$



אתר: www.mathworks.com/help/matlab/creating_plots.html

שימוש בספרייה

וקצת בדיקות על המטריצות...

```

Matrix m1 = new FullMatrix(3,3);
m1.setFromArray(new double[][] {
    {1, 1, 0},
    {0.5, 1, 0},
    {0.2, 0.8, 1}});

System.out.println("m1=" + m1);
System.out.println("m1*5=" + m1.multiply(5));
System.out.println("m1*m1=" + m1.multiply(m1));

Matrix m2 = new DiagonalMatrix(3);
m2.setFromArray(new double[][] {
    {2, 0, 0},
    {0, 3, 0},
    {0, 0, 4}});

System.out.println("m2=" + m2);
System.out.println("m2*5=" + m2.multiply(5));
System.out.println("m2*m1=" + m2.multiply(m1));
    
```

```

m1=[1.0 0.5 0.2; 1.0 1.0 0.8; 0.0 0.0 1.0];
m1*5=[5.0 2.5 1.0; 5.0 5.0 4.0; 0.0 0.0 5.0];
m1*m1=[1.5 1.0 0.8; 2.0 1.5 1.8; 0.0 0.0 1.0];
m2=D[2.0 3.0 4.0];
m2*5=[10.0 15.0 20.0];
m2*m1=[2.0 1.0 0.4; 3.0 3.0; 2.4000000000000004 0.0 0.0 4.0];
    
```

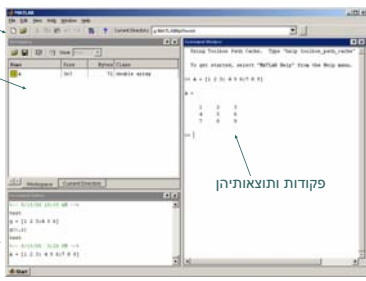
עיצוב המנעק הגראפי

ללמוד מהטובים ביותר - Matlab

תפריטים וסרגלים

אזור העבודה


הסטוריה



פקודות ותוצאותיהן

עיצוב המנעק הגראפי

- כמו שלמדנו בד"כ התכניתן (אנחנו) מקבל בפירוט איך אמור להיראות ולהתנהג המנעק הגראפי
- בוא נניח לרגע שלא קיבלנו... איך הייתם אתם מעצבים מנעק גרפי לפעולות של אלגברה ליניארית?

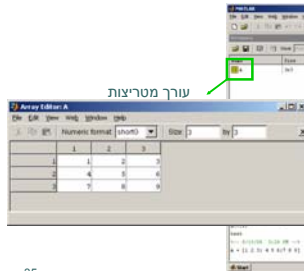


מטרות צעוות

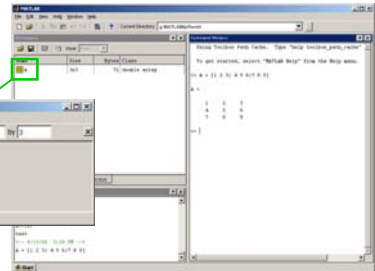
- בואו "נניח" שנמשך משהו בסגנון Matlab, שזו החלטה קצת שאפתנית
- נפרק את העבודה לחלקים
 - ספריית אלגברה ליניארית
 - זיכרון (Workspace)
 - כלי עריכה למטריצות
 - מחולל פקודות (Parser)

עיצוב המנעק הגראפי

ללמוד מהטובים ביותר - Matlab



עורך מטריצות



DialogMatrix

○ SWT מגדיר מחלקה Dialog, מחלקה מופשטת ממנה נוריש כל "דיאלוג" שנרצה

```
public class DialogMatrix extends Dialog {
    private int width, height;
    private Matrix result;

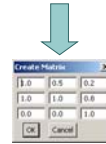
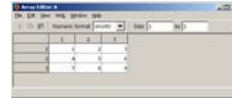
    public DialogMatrix(int width, int height, Shell parent, int style) {
        super(parent, style);

        this.width = width;
        this.height = height;

        result = new FullMatrix(width, height);
        setText("Create Matrix");
    }
    ...
}
```

38

כלי עריכה למטריצות



- בואו נעשה משהו "דומה" למה ש-Matlab מציע
- ניצור Dialog
 - מקבל כקלט מטריצה מגודל מסוים
 - מאפשר לערוך כל תא
 - ניתן לקבל/לדחות השינויים

אך אתם מציעים לממש זאת?

37

DialogMatrix

```
...
shell.setEnabledButton(okButton);
shell.pack();
Dialog.open();
Display display = getParent().getDisplay();
while (!shell.isDisposed()) {
    if (!display.readAndDispatch()) display.sleep();
}
return result;
}
```



| | | | |
|---|---|----|---|
| 0 | 5 | 10 | 9 |
| 1 | 0 | 1 | 8 |
| 4 | 9 | 11 | 4 |
| 0 | 3 | 0 | 2 |
| 9 | 1 | 0 | 1 |

40

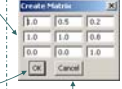
DialogMatrix

```
public Matrix open() {
    Shell parent = getParent();
    Shell shell = new Shell(parent, SWT.DIALOG_TRIM | SWT.APPLICATION_MODAL);
    shell.setText(getText());
    // Your code goes here (widget creation, set result, etc).
    shell.setLayout(new GridLayout(width, true));

    Text cells[] = new Text[width * height];
    for (int i=0; i<height; i++) {
        for (int j=0; j<width; j++) {
            cells[i * width + j] = new Text(shell, SWT.BORDER);
            Text cell = cells[i * width + j];
            cell.setLayoutData(new GridData(SWT.FILL, SWT.CENTER, false, false));
            try {
                cell.setText(String.valueOf(result.elementAt(j, i)));
            } catch (DimensionMismatchException e) {}
        }
    }

    Button okButton = new Button(shell, SWT.NONE);
    okButton.setText("OK");
    okButton.setLayoutData(new GridData(SWT.CENTER, SWT.CENTER, false, false));
    okButton.addSelectionListener(new OKSelectionListener(shell, cells));

    Button cancelButton = new Button(shell, SWT.NONE);
    cancelButton.setText("Cancel");
    cancelButton.setLayoutData(new GridData(SWT.CENTER, SWT.CENTER, false, false));
    cancelButton.addSelectionListener(new CancelSelectionListener(shell));
    ...
}
```



39

ומה הלאה?

- יש עוד הרבה מה לעשות
- מימוש זיכרון (Workspace)
- מימוש מחולל הפקודות
- חיבור כל הרכיבים יחדיו



42

DialogMatrix – טיפול באירועים

מחלקה פנימית:

```
private class OKSelectionListener extends SelectionAdapter {
    private Shell shell;
    private Text[] cells;

    public OKSelectionListener(Shell shell, Text[] cells) {
        super();
        this.shell = shell;
        this.cells = cells;
    }

    public void widgetSelected(SelectionEvent e) {
        for (int i=0; i<height; i++) {
            for (int j=0; j<width; j++) {
                Text cell = cells[i * width + j];

                try {
                    result.elementAt(j, i, Double.valueOf(cell.getText()).doubleValue());
                } catch (DimensionMismatchException dme) {}
            }
        }
        shell.close();
    }
}
```

41