ปฏิบัติการนี้จะเรียนรู้เกี่ยวกับการประยุกต์ใช้ `List` ในการสร้าง

(1) interface `Vector` ที่มีคลาส `DenseVector` และ `SparseVector` เป็นคลาสลูก และ

(2) interface `Matrix` ที่ที่มีคลาส `DenseMatrix` และ `SparseMatrix` เป็นคลาสลูก

สำหรับโจทย์แต่ละข้อ ให้มี package `datastr` สำหรับเก็บคลาสที่เป็น data structure และ package `lab` สำหรับคลาส `Main` ที่มี method `main` ที่เรียกใช้ data structure

โจทย์ปฏิบัติการ ข้อ 1

สร้าง interface `Vector` ที่กำหนด method ต่อไปนี้

- `length()` ที่ส่งมิติ (dimension) ของเวกเตอร์คืน
- `magnitude()` ที่ส่งขนาด (magnitude) ของเวกเตอร์คืน
- `get(int index)` ที่คืนค่า element ที่ `index` ของ implicit parameter
- `set(int index, double value)` ที่แก้ค่า element ที่ `index` ของ implicit parameter เป็น `value`
- `add(Vector v)` ที่คืน `Vector` ที่เป็นผลบวกของ implicit parameter และ explicit parameter โดยจะ throws exception เมื่อไม่สามารถบวกเวกตอร์ได้ (มิติของเวกเตอร์ไม่เท่ากัน)
- `subtract(Vector v)` ที่คืน `Vector` ที่เป็นผลลบของ implicit parameter และ explicit parameter โดยจะ throws exception เมื่อไม่สามารถลบเวกเตอร์ได้ (มิติของเวกเตอร์ไม่เท่ากัน)
- `dot(Vector v)` ที่คืนจำนวนจริงที่เป็น dot product ของ implicit parameter และ explicit parameter โดยจะ throws exception เมื่อไม่สามารถหา dot product ได้ (มิติของเวกเตอร์ไม่เท่ากัน)
- `multiply (double c)` ที่คืน `Vector` ที่เป็นผลคูณของ implicit parameter กับค่าคงที่ `c`

จากนั้นสร้างคลาส `DenseVector` และ `SparseVector` ที่ implement interface `Vector` แล้วเขียนคลาส `Main` เพื่อทดสอบการทำงานของคลาส `DenseVector` และ `SparseVector`

ตัวอย่างการทำงาน

```
v1     = [ 5.0 2.0 9.0 4.0 4.0 1.0 5.0 4.0 1.0 0.0 ]
0.5*v1 = [ 2.5 1.0 4.5 2.0 2.0 0.5 2.5 2.0 0.5 0.0 ]
v2     = [ 1.0 0.0 0.0 0.0 9.0 6.0 5.0 0.0 0.0 9.0 ]
0.5*v2 = [ 0.5 0.0 0.0 0.0 4.5 3.0 2.5 0.0 0.0 4.5 ]
v2+v1  = [ 6.0 2.0 9.0 4.0 13.0 7.0 10.0 4.0 1.0 9.0 ]
v1-v2  = [ 4.0 2.0 9.0 4.0 -5.0 -5.0 0.0 4.0 1.0 -9.0 ]
v1.v2  = 72.0
v2.v1  = 72.0
```

โจทย์ปฏิบัติการ ข้อ 2

เพิ่ม method ต่อไปนี้ใน interface `Vector` และคลาส `DenseVector` และ `SparseVector` ในข้อ 1

- `multiply (Matrix v)` ที่คืน `Vector` ที่เป็นผลคูณของ implicit parameter (`Vector`) ด้วย explicit parameter (`Matrix`) โดยจะ throws exception เมื่อไม่สามารถคูณได้ (incomnpatibility for multiplication)

สร้าง interface `Matrix` ที่กำหนด method ต่อไปนี้

- `numRows()` และ `numCols()` ที่คืนค่าเป็นจำนวน row และ จำนวน column ของเมทริกซ์ ตามลำดับ
- `get(int r, int c)` ที่คืนค่า element ที่ row `r` , column `c` ในเมทริกซ์
- `set(int r, int c, double v)` ที่แก้ค่า element ที่ ที่ row `r` , column `c` ในเมทริกซ์ เป็น `v`
- `add(Matrix v)` ที่คืน `Matrix` ที่เป็นผลบวกของ implicit parameter และ explicit parameter โดยจะ throws exception เมื่อไม่สามารถบวกเมทริกซ์ได้ (มิติของเมทริกซ์ไม่เท่ากัน)
- `multiply (Matrix v)` ที่คืน `Matrix` ที่เป็นผลคูณของ implicit parameter และ explicit parameter โดยจะ throws exception เมื่อไม่สามารถคูณเมทริกซ์ได้ (incomnpatibility for multiplication)

จากนั้นสร้างคลาส `DenseMatrix` และ `SparseMatrix` ที่ใช้ `Vector` ในข้อ 1 และ implement interface `Matrix` , คลาส `DenseMatrix` และคลาส `SparseMatrix` แล้วเขียนคลาส `Main` เพื่อทดสอบการทำงานของคลาส `DenseMatrix` และ `SparseMatrix` โดยสร้างเมทริกซ์โดยสุ่มขนาดเมทริกซ์และค่าในเมทริกซ์ แล้วหาผลบวกและผลคูณของเมทริกซ์

ตัวอย่างการทำงาน

```
matrix m1 =
9.0    1.0    4.0    2.0    7.0    4.0    8.0    0.0    2.0    1.0
5.0    1.0    4.0    0.0    9.0    7.0    3.0    9.0    6.0    0.0
7.0    9.0    6.0    2.0    6.0    8.0    6.0    9.0    7.0    0.0
7.0    5.0    7.0    0.0    5.0    2.0    4.0    5.0    8.0    7.0
9.0    9.0    0.0    5.0    4.0    1.0    0.0    6.0    7.0    4.0
============================
matrix m2 =
7.0    0.0
0.0    0.0
0.0    0.0
0.0    0.0
2.0    0.0
4.0    1.0
2.0    0.0
0.0    3.0
0.0    0.0
0.0    0.0
============================
matrix m3 =
0.0    7.0    0.0    0.0    0.0    3.0    0.0    0.0    0.0    0.0
0.0    0.0    0.0    3.0    7.0    0.0    0.0    4.0    0.0    0.0
7.0    9.0    0.0    1.0    9.0    0.0    9.0    0.0    0.0    0.0
0.0    8.0    0.0    0.0    7.0    0.0    0.0    0.0    0.0    0.0
0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
============================
m1*m2 =
109.0  4.0
87.0   34.0
105.0  35.0
75.0   17.0
75.0   19.0
============================
m1+m3 =
9.0    8.0    4.0    2.0    7.0    7.0    8.0    0.0    2.0    1.0
5.0    1.0    4.0    3.0    16.0   7.0    3.0    13.0   6.0    0.0
14.0   18.0   6.0    3.0    15.0   8.0    15.0   9.0    7.0    0.0
7.0    13.0   7.0    0.0    12.0   2.0    4.0    5.0    8.0    7.0
9.0    9.0    0.0    5.0    4.0    1.0    0.0    6.0    7.0    4.0
```