Lecture 27: Matrices

CS 51G
Spring 2018
Kim Bruce

Announcements

- Exercise 13.8.3
- No official lab or quiz this week
  - Lablet instead
    - Several short methods using lists, due Monday night

Laundry — Again?

- Can use lists and inheritance to make laundry easier.
  - Clothes objects have two lists: solid and framed items.
  - Abstract class can define all methods
  - Pants and T-shirt classes just construct objects

Lists of Lists of ...

Multi-dimensional Lists

```
def d: List[List[Number]] = emptyList[List[Number]]
      or
def d': List[List[Number]] = list[List[Number]] []

d.add (list[Number][2,4,6])
d.add (list[Number][1, 3, 5, 7])

print (d)
for (d) do {val: List[Number] → print("next is: {val}")}
```

If that makes your head hurt

```
type NumList = List[Number]
def d': List[NumList] = emptyList[NumList]

d’.add (list[Number][2,4,6])
d’.add (list[Number][1, 3, 5, 7])

print (d’)
for (d’) do {val: List[Number] → print("next is: {val}")}
```

Matrix

- Define type and class to represent square matrices
  - Formed as a list of rows
- Example of 5 x 3 matrix:

	1	2	3
1	a.at(1).at(1)	a.at(1).at(2)	a.at(1).at(3)
2	a.at(2).at(1)	a.at(2).at(2)	a.at(2).at(3)
3	a.at(3).at(1)	a.at(3).at(2)	a.at(3).at(3)
4	a.at(4).at(1)	a.at(4).at(2)	a.at(4).at(3)
5	a.at(5).at(1)	a.at(5).at(2)	a.at(5).at(3)

Defining Matrix

```
type Matrix⟦T⟧ = {
    at (r: Number, c: Number) -> T
    at (r: Number, c: Number) put (value:T) -> Done
}
```

- See class defined at
  - http://www.cs.pomona.edu/classes/cs051G/demos/Matrix/Matrix.grace
- Allows you to ignore implementation as list of lists.
- Must provide default value for each slot

Puzzle

- Suppose define:
  - def a:Matrix[Number] =
    matrixSize[Number](5,3) defaultValue (0)
  - What is result of executing:
    - for (1..5) do {row: Number ->
      for (1..3) do { col: Number ->
      a.at(row,col)put(4*(row-1) + col)
      }
      }

Tic-Tac-Toe

- See program

Questions?