# **Data Structures: Linked Lists**

# **Objects and classes**

# **Building a social network**

#### Suppose you're in the year 2004 and had this great idea...

- of a website where people could spend countless hours
- uploading personal information, writing messages to people that you see daily
- and you realise that you could sell this information!

#### You decide you want to build the basic data infrastructure of your website first

- A social network consists of people, each individual being a Person
- Each Person has a name, location, age and a list of friends

#### The first step on your road to success, fame and \$\$\$

#### You decide you want to build the basic data infrastructure of your website first

- A social network consists of people, each individual being a Person
- Each Person has a name, location, age and a list of friends



#### You want to test your software with randomly generated people

# To do this, you would like to generate a social network with randomly generated Person objects, each one with randomly assigned firends

- When constructing a SocialNetwork you would decide on the size of the network and the maximum number of friends
- Take a look at the NameGenerator class
- We want to add two methods: getRandomFirstName, and getRandomLastName

#### NameGenerator

#### **Attributes**

- List of first names
- List of last names

#### Methods

- NameGenerator()
- loadNamesFile(String file\_location)
- getRandomFirstName()
- getRandomLastName()

# Now that you have a simulated social network, you are ready for pitching your idea

You want to generate silly statistics that would attract investors to your new startup

- Find the person with the largest number of friends
- Find the 2nd person with the largest number of friends
- Find the N-th person with the largest number of friends
- Find the total number of connections in your social network
- Find if you have isolated groups in your social network (see a4)
- Find the maximum degree of separation

Data structures are a way of organizing collections of data in the computer's memory

#### Data structures are a way of organizing collections of data in the computer's memory

Examples of data structures we have seen so far

• Arrays

#### Data structures are a way of organizing collections of data in the computer's memory

Examples of data structures we have seen so far



• Objects

#### Data structures are a way of organizing collections of data in the computer's memory

Examples of data structures we have seen so far



• Networks of cities (Assignment 4), a social network (last class)

#### Data structures are a way of organizing collections of data in the computer's memory

Examples of data structures we have seen so far

- Arrays
- Objects
- Networks of cities (Assignment 4), a social network (last class)
- Linked Lists (today's class)















This is a visual representation of a linked list using int values



# Going through the components of a LinkedList

#### We need to define a class for each element in the list: the Node class

The Node class should have:

- A pointer to some data. We will call it its value. For simplicity, we will hold int numbers in our LinkedList
- A pointer to the next element in the list. Unsurprisingly, we will call it next

# Going through the components of a LinkedList

#### We need to define a class for doing operations on the list: the LinkedList class

The LinkedLsit class should have:

- A pointer to the **first** element in the list. We will call it start.
- All the methods with the operations we want to make on lists

```
1 public class LinkedList{
2    private Node start;
3 
4    // add a constructor here
5 
6    // add Insertion, search and deletion methods here
7  }
8
```

# Inserting elements in a linked list

- Inserting at the beginning of the LinkedList
- Inserting at the end of the LinkedList
- Inserting somewhere in the middle of the LinkedList
- Deleting a Node from the List

#### Resources

- Classes and Objects: http://docs.oracle.com/javase/tutorial/java/javaOO/
- The Shoelace Algorithm: http://en.wikipedia.org/wiki/Shoelace\_formula
- Suggested reading: How to think like a Computer Scientist, Chapter 11