CSCI 136: Data Structures and Advanced Programming

Lecture 11

Linked Lists

Instructor: Dan Barowy

Williams

Announcements

- Be sure to review `LinkedList.java` before lab

Outline

1. Interfaces
2. Inheritance
3. Singly Linked List
4. Doubly Linked List
5. Circular List

Interface

An **interface** defines boundary between two systems across which they share information. An interface is a **contract**: calling a method defined in an interface returns the data as promised.

An interface **contains no implementation**!

`Honkable`

"We will encourage you to develop the three great virtues of a programmer: **laziness**, **impatience**, and **hubris**."

—Larry Wall, inventor of the Perl programming language

Laziness. The quality that makes you go to great effort to reduce overall energy expenditure. It makes you write labor-saving programs that other people will find useful, and document what you wrote so you don't have to answer so many questions about it. Hence, the first great virtue of a programmer.

Inheritance (cf. laziness)

Inheritance is a **mechanism** for defining a class in terms of another class. It is a labor-saving device employed to reduce **code duplication**. Inheritance allows programmers to specify a new implementation while :

1. **maintaining the same behavior**,
2. **reusing code**, and
3. **extending the functionality** of existing software.

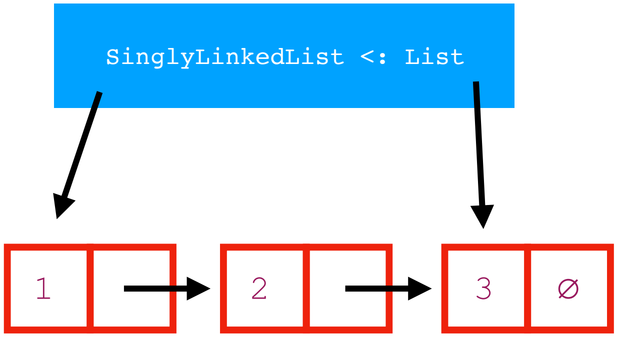
`AbstractHonkable`

code: let's make a linked list!

Problems?

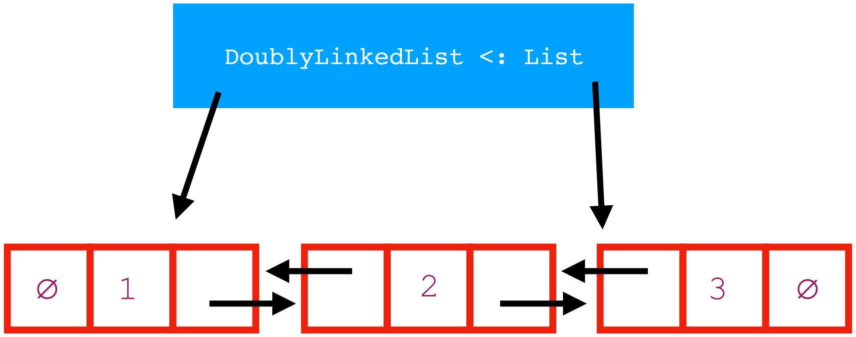
`ListNode` is not a `List`

Abstraction!



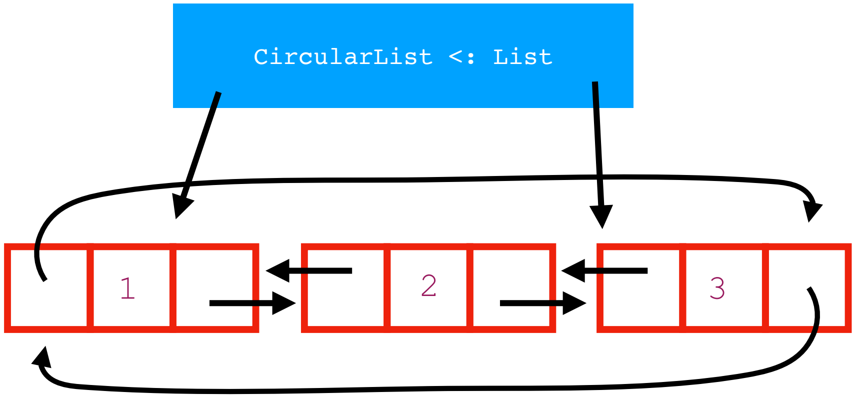
`SinglyLinkedList <: List`

Abstraction!



DoublyLinkedList <: List

Abstraction!



CircularList <: List

Recap & Next Class

Today we learned:

Interfaces and Inhertance in code

Linked Lists

Next class:

Mathematical Induction