| | Topics |
|--|---|
| CSCI 136: Data Structures and Advanced Programming Lecture 20 Ordered Structures | Ordered structures |
| Instructor: Dan Barowy Williams | |
| Your to-dos | Announcements |
| Lab 6 (partner lab), due Tuesday 4/12 by 10pm. Read before Wed: Bailey, Ch 12-12.5. | ACM TechTalk: "Visual Data Analysis: Why? When? How?" Organized by Prof. Kelly Shaw and CoSSAC. Wednesday, April 13 from 7-7:45pm in TBL 211. "Extra special snacks" provided by CoSSAC afterward in the Eco Cafe. This Friday's colloquium: CS pre-registration info session. |

Announcements

Please **consider being a TA** next semester (especially for this class!)

Applications due Friday, April 22.

https://csci.williams.edu/tatutor-application/

Ordered structures

structure5 Stack implementations



structure in structure5

A **structure** is an interface for a "traversable" collection of objects. In other words, it represents a class that **contains** some number of elements, and those elements can be **iterated**, **added**, and **removed**. **Membership** and **size** can also be checked.

Most of the data structures we discuss in this class implement structure.

structure in structure5

```
public interface Structure<E> extends Iterable<E>
{
    public int size();
    public boolean isEmpty();
    public void clear();
    public boolean contains(E value);
    public void add(E value);
    public E remove(E value);
    public E remove(E value);
    public java.util.Enumeration elements();
    public Iterator<E> iterator();
    public Collection<E> values();
}
```

Question for you

Why is a structure interface a good idea? What benefit do we get from having it?

One reason

Suppose we write a **method** that takes a **structure**. We could give it an instance of **any data structure** that implements the structure interface.

E.g., we could **iterate** over the elements and print them because **all structures** have the **iterator()** method.

What about order?

Does the structure interface require that elements be ordered?

structure in structure5

```
public interface Structure<E> extends Iterable<E>
{
    public int size();
    public boolean isEmpty();
    public void clear();
    public boolean contains(E value);
    public void add(E value);
    public E remove(E value);
    public E remove(E value);
    public java.util.Enumeration elements();
    public Iterator<E> iterator();
    public Collection<E> values();
}
```

What about order?

Does the **structure** interface require that elements be **ordered**?

No.

Is order a property that could be enforced using interfaces?

No. Order is a **data-dependent property**, so there's no way to check whether something is ordered until runtime.

OrderedStructure

Nonetheless, we can signal our intent with an interface.

How would we write an OrderedStructure interface?

Do its elements need to have **any special property**? (i.e., how would we **compare** them?)

Let's try to write this.

(code)

OrderedVector

Let's try implementing an OrderedVector.

(code)











Recap & Next Class

Today:

Ordered structures

Next class:

More iterators

Bitwise operations