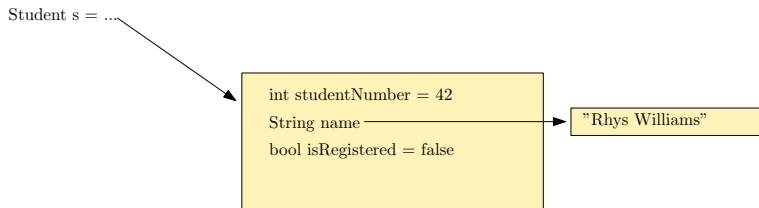


More ADTs and Exceptions

Daniel Archambault

Previously in CS-115



Remember the Class Variables!

Previously in CS-115

- How are variables of simple types represented?

Previously in CS-115

- How are variables of simple types represented?
- How are variables of class types represented?

Previously in CS-115

- How are variables of simple types represented?
- How are variables of class types represented?
- How does assignment work with simple types?

Previously in CS-115

- How are variables of simple types represented?
- How are variables of class types represented?
- How does assignment work with simple types?
- How does assignment work with class types?

Previously in CS-115

- How are variables of simple types represented?
- How are variables of class types represented?
- How does assignment work with simple types?
- How does assignment work with class types?
- What type do arrays behave like?

Previously in CS-115

- How are variables of simple types represented?
- How are variables of class types represented?
- How does assignment work with simple types?
- How does assignment work with class types?
- What type do arrays behave like?
- What is the difference between a static and non-static attribute?

Previously in CS-115

- This lecture is only about ADTs
- Error! Error! This lecture is **also** about exceptions

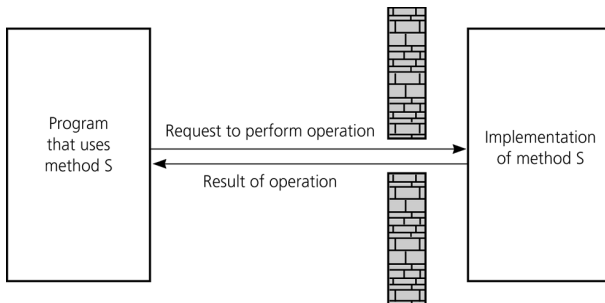
ADTs and Exceptions

Abstract Data Type

- Separates the behaviour of a data structure from its implementation
 - ▶ Detail how the module behaves
 - ▶ Describe the type of data stored
- They should not supply implementation information

Specification of ADT

- Types of operations on data, typically:
 - ▶ Add data to the data
 - ▶ Remove data from the data
 - ▶ Query the data
- Before representation design, think about operations
- Lead to better data representations
 - ▶ If constrained to read only the first element of the list...



Data Abstraction and Problem Solving in Java, p 173

Abstract Data Type vs Data Structure

- ADTs and data structures are not the same thing
- Data structures
 - ▶ Specifies exactly how we store data
 - ▶ Specifies exact implementations of methods
 - ▶ These are composed of classes, attributes, and methods in Java
- ADTs closer to design
- Data structures close to implementation

Attributes of List ADT

- Items in a list have
 - ▶ A unique predecessor
 - ▶ A unique successor
- Front does not have a predecessor
- End does not have a successor
- Doesn't tell you *how* it's done. Just what is supported.

Operations Supported by List ADT

- Create an empty list
- Determine whether a list is empty
- Determine the number of items in a list
- Add an item at a given position in the list
- Remove the item at a given position in the list
- Remove all the items from the list
- Retrieve (get) the item at a given position in the list
- Doesn't tell you *how* it's done. Just what is supported.

In Java

```
public Interface List{  
    public boolean isEmpty ();  
    public int numberOfElements ();  
    public void addItem (ListItem li, int pos);  
    public ListItem getItem (int pos);  
}
```

- The closest thing to ADTs in Java are interfaces
- Specify methods without implementation
- Specify constants
- Not allowed attributes or any implementation
- You can declare interfaces in Java (we will get to this)

Exceptions

A problem has been detected and windows has been shut down to prevent damage to your computer.

The problem seems to be caused by the following file: SPCMDCON.SYS

PAGE_FAULT_IN_NONPAGED_AREA

If this is the first time you've seen this Stop error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

Technical information:

*** STOP: 0x00000050 (0xFD3094C2, 0x00000001, 0xFBFE7617, 0x00000000)

*** SPCMDCON.SYS - Address FBFE7617 base at FBFE5000, DateStamp 3d6dd67c

https://upload.wikimedia.org/wikipedia/commons/a/a8/Windows_XP_BSOD.png

● Well, that wasn't handled properly...

Why Exception Handling?

- Exception Handling is about handling errors.
 - ▶ To err is human...
 - ▶ It's about anticipating very common errors
 - ▶ We aim to build robust and fault tolerant programs.
 - ▶ Often associated with Input and Output (I/O), e.g., file I/O, user input
 - ▶ More examples
 - ★ out-of-bounds array index
 - ★ arithmetic overflow (i.e. value outside range of values),
 - ★ division by zero
 - ▶ Basically, recover now before things go really wrong

Errors in Java

- In Java, errors are *thrown* and *caught*
 - ▶ the code that caused the error throws an exception
 - ★ the exception causes an immediate return out of the method
 - ▶ the code that called the method has a catch clause
 - ★ this code receives the error and handles it
- The error itself is a class that inherits from `Throwable`
- `Throwable` is a superclass of all errors in Java

How to Handle Errors in Java

```
try {  
    //Do some risky method calls...  
}  
catch (IOException e1) {  
    //if an IOException recover here  
}  
catch (LemonException e2) {  
    //if Neal Harman threw an exception recover here  
}  
catch (Exception e3) {  
    //matches everything that inherits from exception  
}
```

- Appropriate catches called depending on error type

What about finally?

```
try {  
    //Do some risky method calls...  
}  
....  
finally {  
    //stuff happens here  
}
```

- No matter what happens, finally gets executed
 - ▶ any error, finally gets executed
 - ▶ no errors at all, finally gets executed
- Finally usually releases resources, closes files/database connections, and network connections

But I want to throw my own errors!

- You can write code that doesn't work... (not good)
- But you can also write code that throws good errors
 - ▶ in method definition declare method that throws an error
 - ▶ in the code, detect the error with if statements
 - ▶ use the **throw** keyword

Example of Throwing Exceptions

```
int div (int a, int b) throws ArithmeticException {  
    if (b == 0)  
        throw new ArithmeticException ("Div by 0");  
    else  
        return a/b;  
}
```

- Outside this method, you can catch the `ArithmeticException`

Review

- What is an ADT?

Review

- What is an ADT?
- How does it differ from a data structure?

Review

- What is an ADT?
- How does it differ from a data structure?
- How do we handle code that could throw an error?

Review

- What is an ADT?
- How does it differ from a data structure?
- How do we handle code that could throw an error?
- How do we get a method to throw an error?