CS 600.226: Data Structures Michael Schatz

August 31, 2018 Lecture 2: Introduction to Interfaces



Introductions



Tim Kutcher Head TA

Agenda

- I. Quick Review
- 2. Introduction to Virtual Machines
- 3. Introduction to Java Interfaces

Agenda

I. Quick Review

- 2. Introduction to Virtual Machines
- 3. Introduction to Java Interfaces



Welcome!

Course Webpage: Course Discussions:	https://github.com/schatzlab/datastructures2018 https://piazza.com/jhu/fall2018/600226/home
Office Hours:	Wednesday @ 2:45pm – 4pm, Malone 323 CA office hours throughout the week ©
Programming Language:	Java with Checkstyle and JUnit Virtual Machine (Lubuntu) or CS acct.
Accounts for Majors (CS/C	CE) & Minors:

If you do not already have a personal CS departmental unix account, please complete an account request form ASAP. Check "Linux Undergrad" for account type. (Note - must be declared to be eligible.)

Accounts for Others:

We will need to make accounts. Do people need them?

CS Lab access:

Students must see Steve DiBlasio, with your J-card, in Malone G61A to get CS Lab access. The CS Lab is Malone 122 and that's where course TA/CAs will be available for help.

Course Webpage

https://github.com/schatzlab/datastructures2018

Search or jump to	Pull requests Issues Marketplace	Explore		* +• (
O Code ① Issues 0 □ Pull request	s 0 🍯 Projects 0 💷 Wiki 🕁 Insigi	nts O Settings	⊕ Unwatch + 1 🖈 S	tar 0 ¥Fork 0
faterials for JHU Data Structures 2018 dd topics				Edit
@ 47 commits	1 branch	⊙ o releases	AL 2 cc	intributors
Branch: master + New pull request		Create new file	Upload files Find file	Clone or download -
mschatz Update README.md			Latest commi	t 34a9618 8 minutes ago
In assignments	Update README.md			9 days ago
in lectures	Create README.md			9 days ago
illa policies	Create readme.md			9 days ago
lite resources	Update README.md			7 days ago
gitignore	gitignore			7 days ago
README.md	Update README.md			8 minutes ago
ER README.md				/
JHU EN.600.226:	Data Structures			

The primary goal of the course is for students to be grounded in theory and leave the course empowered to understand and implement a

Show me the photos!



Each sublist has N/8 elements & Balanced in both dimensions

Divide and Conquer

- Brute force is slow because we have to check every single element
 - How can we split up the unsorted list into independent ranges?
 - Lets recursively split up the elements into greater than/less than range based on the current split line (latitude/longitude)



[How many times can we split a list in half?]

How much is a zettabyte?

Unit	Size	~2×
Byte		2 ⁰
Kilobyte	Ι,000	2 ¹⁰
Megabyte	1,000,000	2 ²⁰
Gigabyte	1,000,000,000	2 ³⁰
Terabyte	1,000,000,000,000	240
Petabyte	1,000,000,000,000,000	2 ⁵⁰
Exabyte	1,000,000,000,000,000,000	2 ⁶⁰
Zettabyte	1,000,000,000,000,000,000,000	2 ⁷⁰

How much is a zettabyte?

Unit		Size	~2×
Byte			2 ⁰
Kilobyte		I,000	210
Megabyt			2 ²⁰
Gigabyte	For a	Il practical purposes:	2 ³⁰
Terabyte		lg(X) << 70	2 ⁴⁰
Petabyte			2 ⁵⁰
Exabyte		I,000,000,000,000,000,000	2 ⁶⁰
Zettabyte		1,000,000,000,000,000,000,000	270

Agenda

I. Quick Review

2. Introduction to Virtual Machines

3. Introduction to Java Interfaces



What is a computer? [hardware]



Display Human Interface *Network* Computer Interface Home: 10Mb/s, JHU: 1Gb/s+



Virtual Machines



- Powerful way for 1 physical computer to run >1 operating system
- Run Windows Apps on a Mac host, run Linux within a PC
- Key technology for cloud computing

VirtualBox



- Client application available for Mac, Windows, Linux
- Available to run our reference virtual machine running linux
 - Guaranteed that your development environment matches testing environment
 - Make sure to install the Extension Pack and Guest Additions too

VirtualBox Setup

	 Grundo inc. [G3] units://gindocom/scharzabi/datastructureszono/podmaster/resourcester/inclusion/setab/	0 -		-	-
1	/irtual Machine Instructions				
	o setup your virtual machine, you will first need some software that can run your virtual machine. Our go-to is VirtualBox which can be down on VM terminology, the host is the machine that you have (that will host our Linux Virtual Machine). So download the host for the OS you have	vnloa ve.	aded	here	•
	lext, you will want to download the VM we have set up. To do so, simply follow this google drive link (such a large file it can't be in this report lownload button, then open the cs226.ova file once installed. This should automatically add the machine in VirtualBox with all of the install here are still a few things you will want to do.	sitor ed to	ry), c iols,	lick ti but	he
į	'he default username is cs226 and the default password is password . You can change these if you'd like, as well as the name of the comp	uter			
	irst, you'll want to install the virtual box guest additions for this machine. You can follow instructions here to get the guest additions. This a mproved graphics, clipboard sharing, etc.	llow	s for		
	lext, you will want to configure some settings. Launch VirtualBox and single-click (i.e. do not start) the machine you just installed (cs226). nd change the name if you'd like. Select the "Advanced" tab under the "General" screen and change both of "Shared Clipboard" and "Drag Bidirectional". This allows you to copy and paste from host to the VM and vise versa. If you go to the System screen you could change the nemory used, it defaults to 2048MB. Click "OK" when done to apply these settings. Now when you boot up the VM you should have this st unctionality.	Click 'n'Di amoi arin	c "Se rop" unt c g	tting: to f	5",
	or sharing files between the VM and your computer you have a few options. I would reccommend just using Git, but you can also drag and firectly in the VM, or experiment with shared folders (I haven't dabbled with this so I don't want to give any input).	drog	o, su	bmit	
	ome general notes for using the VM				
	 Check out ~/.bash_aliases to see some of the shortcuts I've included and add your own. 				
	 Customize your terminal by editing ~/.bashrc 				
	 Don't ever just exit out of the VM, make sure you power off and let the machine shut down on it's own. 				
	lools				

Downloading JUnit

To download JUnit, you can run \$ sudo apt-get install junit4 (JUnit 4 is the version we use but it is not as important as far as writing your own tests).

VirtualBox Setup

- 1. Install "host" application to run on your computer
 - Separate app for Mac, Windows or Linux
- 2. Download virtual machine image file cs226.ova from google drive
 - Warning: Big file! (~2.9 GB; at least 5 minutes to download)
- 3. Import image file (double click on cs226.ova)
 - Defaults are okay, but will take ~1m
- 4. Install Oracle VM VirtualBox Extension Pack
 - Improves speed and performance:
 - https://www.virtualbox.org/wiki/Downloads
- 5. Boot virtual machine!
 - You are now running a linux computer inside of your computer
- 6. Install virtual box guest additions
 - Further improves speed and performance
 - <u>https://virtualboxes.org/doc/installing-guest-additions-on-ubuntu/</u>
- 7. Setup development environment, code, and run!



Ubuntu



File Hierarchy

Files are stored in nested directories (folders) that form a tree

- •The top of the tree is called the root, and is spelled '/'
- •Your home directory (on mac) is at /Users/username
- •Command line tools are at /bin/ /usr/bin/ /usr/local/bin/



- •A few special directories have shortcuts
 - ~ = home directory
 - ~bob= bob's home directory
 - = current working directory
 - .. = parent directory
 - = last working directory

Working with the shell

• The shell is interactive and will attempt to complete your command as soon as you press enter

\$ pwd
/Users/mschatz

\$ ls
Desktop/ Library/ Public/ bin/ Documents/ Movies/
Downloads/ Music/ Dropbox/ Pictures/

• Here are a few shortcuts that will make your life easier

Command	Effect
Left/Right arrow	Edit your current command
Up/Down arrow	Scroll back and forth through your command history
Control-r	Search backwards through your command history
Control-c	Cancel the command
Control-u	Clear the current line
Control-a, Control-e	Jump to the beginning and end of the line
Tab	Complete a partial filename or path

Unix essentials

Command	Output
man	Look up something in the manual (also try Google)
ls	List the files in the current directory
cd	Change to a different directory
pwd	Print the working directory
mv, cp, rm	Move, copy, remove files
mkdir, rmdir	Make or remove directories
cat, less, head, tail, cat	Display (parts) of a text file
echo	Print a string
sort, uniq	Sort a file, get the unique lines
grep	Find files containing a search string
chmod	Change permissions on a file
WC	Count lines in a file
history	What commands did I just run?
(pipe), > (redirect)	Send output to a different program, different file

CAUTION!

Delete one directory:

\$ rm -rf ~/Documents/cs226/datastructure2018/asn1/

Delete everything in your home directory 😕

\$ rm -rf ~ /Documents/cs226/datastructure2018/asn1/

Unix is *very* unforgiving

A single space can ruin your day/week/year/career!

Use github and dropbox to make frequent checkpoints of your code

Syncing the repo

0.0.0		cs226 [Running]	
Activities	🖾 Terminal 🔫	Thu 21:52	
6	Trash	cr226@cr226.VirtualBox: - /Documents/cr226.Matastructures/2019/Jectures/02.Interfaces	
S		CS220@CS220-VirtualBox: ~/Documents/cS220/datastructures201a/iectures/02.interraces	
	test test Terminal (Shortcut) cs226 (Shortcut)	<pre>File Edit View Search Terminal Help cs226@cs226-VirtualBox:-/Documents/cs226 \$ ls datastructures2018 README update_repo.sh cs226@cs226-VirtualBox:-/Documents/cs226 \$ ls datastructures2018/ cs226@cs226-VirtualBox:-/Documents/cs226 \$ sh update_repo.sh Cloning into 'datastructures2018' remote: Counting objects: 200, done. remote: Compressing objects: 100% (164/164), done. remote: Total 200 (delta 82), reused 57 (delta 17), pack-reused 0 Receiving objects: 100% (200/200), 4.19 MiB 1.86 MiB/s, done. Resolving deltas: 100% (82/82), done. cs226@cs226-VirtualBox:-/Documents/cs226 \$ ls datastructures2018 README update_repo.sh cs226@cs226-VirtualBox:-/Documents/cs226 \$ ls datastructures2018/ assignments lectures policies README.nd resources cs226@cs226-VirtualBox:-/Documents/cs226 \$ cd datastructures2018/lectures/02.Interfaces/ cs226@cs226-VirtualBox:-/Documents/cs226 \$ ls HelloAll.java HelloWorld.java cs226@cs226-VirtualBox:-/Documents/cs226/datastructures2018/lectures/02.Interfaces \$ ls</pre>	
			🕅 🚳 🛡 Left X

After running 'update_repo.sh' just do 'git pull' within the datastructures2018 directory

Our first java program inside VirtualBox



Our first java program inside VirtualBox

0.0.0		cs226 [Running]	
Activities	🖸 Terminal 🔻	Thu 22:06	∴ 🕫 🖛 -
6	-		
	cs226g	pcs226-VirtualBox: ~/Documents/cs226/datastructures2018/lectures/02.interfaces	000
	File Edit View Search Terminal He cs226@cs226-VirtualBox:~/Docur S cat HelloWorld.java	Hp ments/cs226/datastructures2018/lectures/02.Interfaces	
	public class HelloWorld {		
	public static void main {	n(String [] argv)	
0	System.out.pri } }	ntln("Hello World!");	
	cs226@cs226-VirtualBox:~/Docur \$ javac -Xlint:all HelloWorld.	ents/cs226/datastructures2018/lectures/02.Interfaces java	
-	cs226@cs226-VirtualBox:~/Docur \$ java HelloWorld	ments/cs226/datastructures2018/lectures/02.Interfaces	
A	Hello World! cs226@cs226-VirtualBox:-/Docum	ments/cs226/datastructures2018/lectures/02.Interfaces	
?	s -		
<u>a</u>			
2-			

Working the command line



Working the command line



💁 💮 🍋 🚅 🥔 🚍 🌌 🔐 🔘 🚳 🖲 Left 🗶

Introduction to Checkstyle http://checkstyle.sourceforge.net/



Introduction to Checkstyle

O Not Secure checkstyle.sourceforge.net/reports/google-java-	style-20170228.html	Ŷ	0	ų	M:	0 1	0	P 0	Ξ	+
	Google Java Style Guide									
Table of Contents										
1 Introduction	4.6 Whitespace									
1.1 Terminology notes	4.7 Grouping parentheses: recommended									
1.2 Guide notes	4.8 Specific constructs									
2.Source file basics	5 Naming									
2.1 File name	5.1 Rules common to all identifiers									
2.2 File encoding: UTF-8	5.2 Rules by identifier type									
2.3 Special characters	5.3 Camel case: defined									
3 Source file structure	6 Programming Practices									
3.1 License or copyright information. if present	6.1.@Override: always used									
3.2 Package statement	6.2 Caught exceptions: not ignored									
3.3 Import statements	6.3 Static members: qualified using class									
3.4 Class declaration	6.4 Finalizers: not used									
4 Formatting	7.Javadoc									
4.1 Braces	7.1 Formatting									
4.2 Block indentation: +2 spaces	7.2 The summary fragment									
4.3 One statement per line	7.3 Where Javadoc is used									
4.4 Column limit: 100										
4.5 Line-wrapping										

This document serves as the complete definition of Google's coding standards for source code in the Java M Programming Language. A Java source file is described as being in Google Style if and only if it adheres to the rules herein.

Introduction to Checkstyle

000	cs226 [Running]	
Activities	Terminal - Thu 21:58	A ≪ @ -
(٢)		
	cs226@cs226-VirtualBox: ~/Documents/cs226/datastructures2018/lectures/02.Interfaces	© © ©
1	File Edit View Search Terminal Help	
2	<pre>cs226@cs226-VirtualBox:~/Documents/cs226/datastructures2018/lectures/02.Interfaces \$ java -jar ~/Documents/cs226/datastructures2018/resources/checkstyle-8.12-all.jar -c ~/Documents/cs22 s2018/resources/cs226_checks.xml HelloWorld.java</pre>	26/datastructure
	<pre>Starting audit [ERROR] /home/cs226/Documents/cs226/datastructures2018/lectures/02.Interfaces/HelloWorld.java:1: Miss' mment. [JavadocType]</pre>	lng a Javadoc co
	<pre>[ERROR] /home/cs226/Documents/cs226/datastructures2018/lectures/02.Interfaces/HelloWorld.java:1:1: Utionuld not have a public or default constructor. [HideUtilityClassConstructor] [ERROR] /home/cs226/Documents/cs226/datastructures2018/lectures/02.Interfaces/HelloWorld.java:1:1: Utionuld.java:1:1: Utionuld.java:2:1: //</pre>	llity classes sh
	ould be on the previous line. [LeftCurly]	
	<pre>[ERROR] /home/cs226/Documents/cs226/datastructures2018/lectures/02.Interfaces/HelloWorld.java:3: 'meth ' has incorrect indentation level 8, expected level should be 4. [Indentation]</pre>	hod def modifier
-0-	[ERROR] /home/cs226/Documents/cs226/datastructures2018/lectures/02.Interfaces/HelloWorld.java:3:1: Fi characters (this is the first instance). [FileTabCharacter]	le contains tab
A	[ERROR] /home/cs226/Documents/cs226/datastructures2018/lectures/02.Interfaces/HelloWorld.java:3:9: Mis comment. [JavadocMethod]	ssing a Javadoc
2	<pre>[ERROR] /home/cs226/Documents/cs226/datastructures2018/lectures/02.Interfaces/HelloWorld.java:3:39: 's wed by whitespace. [NoWhitespaceAfter]</pre>	string' is follo
	[ERROR] /home/cs226/Documents/cs226/datastructures2018/lectures/02.Interfaces/HelloWorld.java:4: 'meti has incorrect indentation level 8, expected level should be 4. [Indentation]	hod def lcurly'
a	[ERROR] /home/cs226/Documents/cs226/datastructures2018/lectures/02.Interfaces/HelloWorld.java:4:9: '{ ould be on the previous line. [LeftCurly]	at column 2 sh
	[ERROR] /home/cs226/Documents/cs226/datastructures2018/lectures/02.Interfaces/HelloWorld.java:5: 'meth	hod def' child h
2-	[ERROR] /home/cs226/Documents/cs226/datastructures2018/lectures/02.Interfaces/HelloWorld.java:6: 'meth has incorrect indentation level 8, expected level should be 4. [Indentation] Audit done.	nod def rcurly'
	Checkstyle ends with 11 errors.	
	<pre>cs226gcs226-VirtualBox:~/Documents/cs226/datastructures2018/lectures/02.interfaces \$ </pre>	
::::		
	월 2 월 2 4	🌶 🚍 🚅 🔛 🥥 🔅 🖲 Left 36

Introduction to Checkstyle



\$ java -jar ~/Documents/cs226/datastructures2018/resources/checkstyle-8.12-all.jar -c ~/Documents/cs226/datastructures2018/resources/cs226_checks.xml HelloWorld.java

Checkstyle Eclipse Plugin

<text></text>	r Bookmark
<page-header><page-header><text><section-header><section-header><section-header></section-header></section-header></section-header></text></page-header></page-header>	
<section-header><section-header><section-header><section-header><text><text></text></text></section-header></section-header></section-header></section-header>	
Eclipse Checkstyle Plugin Checkstyle integration into the Eclipse IDE. Coding standards made easy. If Install 2 Code 1 Install 2 Code 1 Install via Eclipse Marketplace. Drag and drop this link into a running Eclipse Indigo/Juno/Kepler/Luna/Mars workspace 2 Latest release 6.19.1, based on Checkstyle 6.19, see release notes	
Plugin Checkstyle integration into the Eclipse IDE. Coding standards made easy. Install Download 2 Install via Eclipse Marketplace. Drag and drop this link into a running Eclipse Indigo/Juno/Kepler/Luna/Mars workspace Latest release 6.19.1, based on Checkstyle 6.19, see release notes	
Checkstyle integration into the Eclipse IDE. Coding standards made easy. Install 1 Download 2 Download 2 Install via Eclipse Marketplace. Drag and drop this link into a running Eclipse Indigo/Juno/Kepler/Luna/Mars workspace Latest release 6.19.1, based on Checkstyle 6.19, see release notes	
Install 1 Download 2 Code ¹ Install via Eclipse Marketplace. Drag and drop this link into a running Eclipse Indigo/Juno/Kepler/Luna/Mars workspace ² Latest release 6.19.1, based on Checkstyle 6.19, see release notes	
¹ Install via Eclipse Marketplace. Drag and drop this link into a running Eclipse Indigo/Juno/Kepler/Luna/Mars workspace ² Latest release 6.19.1, based on Checkstyle 6.19, see release notes	
echose	
COMMUNITY AWARDS 2007 WINNER	

Agenda

- I. Quick Review
- 2. Introduction to Virtual Machines
- 3. Introduction to Java Interfaces

Interfaces



Interfaces



What other interfaces do you know?

Cars Elevators Pedestrian crossings & street lights



What other interfaces do you know?

Cars Elevators Pedestrian crossings & street lights

Java API

- Description of how objects, classes, keywords behave
- Implementations available for many many different platforms

The internet:

- TCP/IP stack defines the protocols for computers to talk to each other
- Implementations available for different platforms

Fortnight

- Walking around, hitting/shooting items causes some reaction
- Closed implementation and interface

How would you write a Fortnight bot?

Data Structures



Building, searching, traversing, analyzing Make you big-data superheros ©

Introduction to Java Interfaces

Objects define their interaction with the outside world through the methods that they expose. Methods form the object's interface with the outside world; the buttons on the front of your television set, for example, are the interface between you and the electrical wiring on the other side of its plastic casing. You press the "power" button to turn the television on and off. [...] **An interface is a group of related methods with empty bodies.**

https://docs.oracle.com/javase/tutorial/java/concepts/interface.html



```
interface Counter {
    int value();
    void up();
    void down();
}
```

specification: Counter has an integer value '+' button increments by 1 '-' button decrements by 1

Specification can be a separate documents or in the javadoc comments

One Possible Implementation: MyCounter

```
/**
    The most basic counter.
*/
public class MyCounter implements Counter {
    // Current value of the counter.
    private int value;
    /**
        A counter that starts at 0 and increments/decrements by 1.
    */
    public MyCounter() {}
    @Override
    public int value() {
        return this.value;
    }
    @Override
    public void up() {
        this.value += 1;
    }
    @Override
    public void down() {
        this.value -= 1;
    }
<continues>
```

One Possible Implementation: MyCounter

<continues>

}

```
/**
    Simple assert-based unit tests for this counter.
    Make sure you run MyCounter with -enableassertions!
    We'll learn a much better approach to unit testing later.
    @param args Command line arguments.
*/
public static void main(String[] args) {
    Counter c = new MyCounter();
    assert c.value() == 0;
    c.up();
    assert c.value() == 1;
    c.down();
    assert c.value() == 0;
    c.down();
    c.up();
    c.up();
    c.up();
    assert c.value() == 2;
}
```

Why would we ever use this???

Using Interfaces

```
public class CounterDriver {
    public static void main(String[] args) {
        Counter c = null;
        if ((args.length > 0) && (args[0].equals("Weird"))) {
          c = new WeirdCounter();
        } else {
          c = new MyCounter();
        }
        System.out.println("Counter is " + c.value());
        c.up();
        System.out.println("Counter is " + c.value());
        c.down();
        System.out.println("Counter is " + c.value());
    }
```

}

Because MyCounter and WeirdCounter both implement Counter, we can determine at runtime which flavor to use for c, potentially leading to very different results

CounterDriver Results

0.0.0	cs226 [Running]		
Activities	Terminal Thu 23:11	A 40	۰ ا
٩	cs226@cs226-VirtualBox: ~/Documents/cs226/datastructures2018/lectures/02.Interfaces File Edit View Search Terminal Help	000	
9	<pre>\$ cat CounterDriver.java public class CounterDriver { public static void main(String[] args) {</pre>		
	<pre>Counter c = null; if ((args.length > 0) && (args[0].equals("Weird"))) { c = new WeirdCounter(); </pre>		
0	else {		
	<pre>System.out.println("Counter is " + c.value());</pre>		
	c.up(); System.out.println("Counter is " + c.value());		
?	<pre>c.down(); System.out.println("Counter is " + c.value());</pre>		
a	}		
2	<pre>cs226@cs226-VirtualBox:~/Documents/cs226/datastructures2018/lectures/02.Interfaces \$ java CounterDriver foo Counter is 0 Counter is 1</pre>		
	Counter is 0 cs226@cs226-VirtualBox:~/Documents/cs226/datastructures2018/lectures/02.Interfaces \$ java CounterDriver Weird Counter is 0 Counter is 42		
	Counter ts 41	0 h 2 / 2 2 2 0 0 0	Left 36

Power of Interfaces

Interfaces form a contract between the class and the outside world, and this contract is enforced at build time by the compiler. If your class claims to implement an interface, all methods defined by that interface must appear in its source code before the class will successfully compile.

https://docs.oracle.com/javase/tutorial/java/concepts/interface.html

The power of interfaces is that there may be multiple implementations of the same abstract idea.

- Toyotas and Teslas have totally different designs but knowing the interface of one lets you know how to use the other
- These variants are formally described through an abstract data type the specifies the contract that the interface will follow.
- Importantly, the interface only controls the syntax of what is allowed, the ADT and implementation define the semantics (meaning) of the code.
- In the upcoming homework you will be asked to implement several different versions of the counter that do slightly different things: multiple/divide rather than add/subtract, etc.

Next Steps

- I. Reflect on the magic and power of interfaces ③
- 2. Check on Piazza
- 3. Download class virtual machine, get CS account and/or set up Linux!
- 4. Get comfortable with a editor (VI rules!) and/or an IDE (Eclipse for Java)
- 5. Get comfortable with checkstyle



Questions?