

LEC 14

CSE 122

Interfaces

BEFORE WE START

*Chat with neighbors:**What is your favorite fast food chain?*Music: [122 26Wi Lecture Tunes](#) 🐼**Instructor:** Adrian Salguero

<b>TAs:</b>	Ava	Dalton	Neal	Shreyank
	Blake R	Dani	Neha	Sthiti
	Blake P	David	Nicolae	Sushma
	Cady	Diya	Nicole	Suyash
	Caleb	Hanna	Rio	TJ
	Cole	Ivy	Rohan	Wesley
	Colin	Mahima	Saachi	Yang
	Connor	Medha	Shreya	

Questions during Class?

Raise hand or send here

sli.do #cse122



# Lecture Outline

- **Announcements** 
- Interface Review
- More Shapes!
- Comparable

# Announcements

- Creative Project 2 (C2) due Thursday, February 26<sup>th</sup>
- Resubmission Cycle 5 (R5) out Thursday, February 26<sup>th</sup>
  - Eligible: **P1**, P2
- Programming Assignment 3 (P3) out Friday!
  - Due March 5<sup>th</sup> by 11:59 PM
- Quiz 2 Tuesday, March 3<sup>rd</sup>
  - Practice Quiz coming out before then!
- Reminder on Final Exam: **Tuesday, March 17<sup>th</sup> 12:30 PM - 2:20 PM**

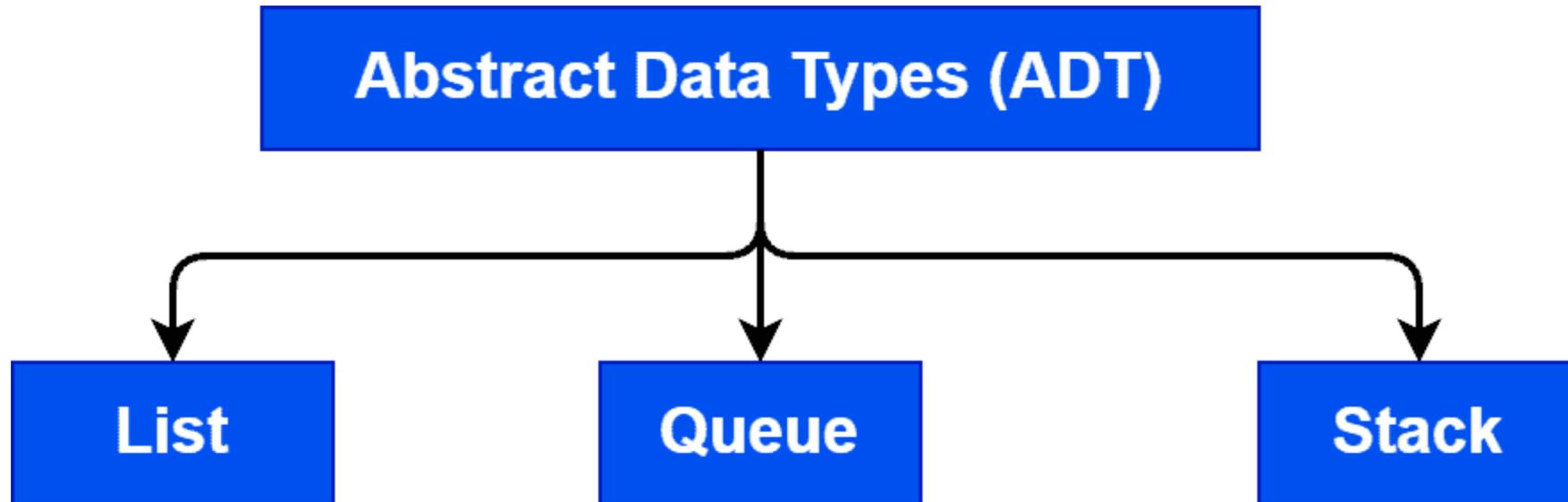
# Lecture Outline

- Announcements
- **Interfaces Review** ◀
- More Shapes!
- Comparable

# Recall: Abstraction

```
J_Scannerclass X
1  /*
2   * Copyright (c) 2003, 2022, Oracle and/or its affiliates. All rights reserved.
3   * DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS FILE HEADER.
4   *
5   * This code is free software; you can redistribute it and/or modify it
6   * under the terms of the GNU General Public License version 2 only, as
7   * published by the Free Software Foundation. Oracle designates this
8   * particular file as subject to the "Classpath" exception as provided
9   * by Oracle in the LICENSE file that accompanied this code.
10  *
11  * This code is distributed in the hope that it will be useful, but WITHOUT
12  * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
13  * FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
14  * version 2 for more details (a copy is included in the LICENSE file that
15  * accompanied this code).
16  *
17  * You should have received a copy of the GNU General Public License version
18  * 2 along with this work; if not, write to the Free Software Foundation,
19  * Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA.
20  *
21  * Please contact Oracle, 500 Oracle Parkway, Redwood Shores, CA 94065 USA
22  * or visit www.oracle.com if you need additional information or have any
23  * questions.
24  */
25
26 package java.util;
27
28 import java.io.*;
29 import java.math.*;
30 import java.nio.*;
31 import java.nio.channels.*;
32 import java.nio.charset.*;
33 import java.nio.file.*;
34 import java.nio.file.Files;
35 import java.text.*;
36 import java.text.spi.NumberFormatProvider;
37 import java.util.function.Consumer;
38 import java.util.regex.*;
39 import java.util.stream.*;
40 import java.util.stream.StreamSupport;
41 import sun.util.locale.provider.LocaleProviderAdapter;
42 import sun.util.locale.provider.ResourceBundleBasedAdapter;
43
44 /**
45  * A simple text scanner which can parse primitive types and strings using
46  * regular expressions.
47  *
48  * Scanner breaks its input into tokens using a
49  * delimiter pattern, which by default matches whitespace. The resulting
50  * tokens may then be converted into values of different types using the
51  * various next methods.
52  *
53  * Scanner allows a user to read a number from
54  * the console.
55  *
56  * Scanner
57  *
58  * Scanner
59  *
60  * Scanner
61  *
62  * Scanner
63  *
64  * Scanner
65  *
66  * Scanner
67  *
68  * Scanner
69  *
70  * Scanner
71  *
72  * Scanner
73  *
74  * Scanner
75  *
76  * Scanner
77  *
78  * Scanner
79  *
80  * Scanner
81  *
82  * Scanner
83  *
84  * Scanner
85  *
86  * Scanner
87  *
88  * Scanner
89  *
90  * Scanner
91  *
92  * Scanner
93  *
94  * Scanner
95  *
96  * Scanner
97  *
98  * Scanner
99  *
100 * Scanner
101 *
102 * Scanner
103 *
104 * Scanner
105 *
106 * Scanner
107 *
108 * Scanner
109 *
110 * Scanner
111 *
112 * Scanner
113 *
114 * Scanner
115 *
116 * Scanner
117 *
118 * Scanner
119 *
120 * Scanner
121 *
122 * Scanner
123 *
124 * Scanner
125 *
126 * Scanner
127 *
128 * Scanner
129 *
130 * Scanner
131 *
132 * Scanner
133 *
134 * Scanner
135 *
136 * Scanner
137 *
138 * Scanner
139 *
140 * Scanner
141 *
142 * Scanner
143 *
144 * Scanner
145 *
146 * Scanner
147 *
148 * Scanner
149 *
150 * Scanner
151 *
152 * Scanner
153 *
154 * Scanner
155 *
156 * Scanner
157 *
158 * Scanner
159 *
160 * Scanner
161 *
162 * Scanner
163 *
164 * Scanner
165 *
166 * Scanner
167 *
168 * Scanner
169 *
170 * Scanner
171 *
172 * Scanner
173 *
174 * Scanner
175 *
176 * Scanner
177 *
178 * Scanner
179 *
180 * Scanner
181 *
182 * Scanner
183 *
184 * Scanner
185 *
186 * Scanner
187 *
188 * Scanner
189 *
190 * Scanner
191 *
192 * Scanner
193 *
194 * Scanner
195 *
196 * Scanner
197 *
198 * Scanner
199 *
200 * Scanner
201 *
202 * Scanner
203 *
204 * Scanner
205 *
206 * Scanner
207 *
208 * Scanner
209 *
210 * Scanner
211 *
212 * Scanner
213 *
214 * Scanner
215 *
216 * Scanner
217 *
218 * Scanner
219 *
220 * Scanner
221 *
222 * Scanner
223 *
224 * Scanner
225 *
226 * Scanner
227 *
228 * Scanner
229 *
230 * Scanner
231 *
232 * Scanner
233 *
234 * Scanner
235 *
236 * Scanner
237 *
238 * Scanner
239 *
240 * Scanner
241 *
242 * Scanner
243 *
244 * Scanner
245 *
246 * Scanner
247 *
248 * Scanner
249 *
250 * Scanner
251 *
252 * Scanner
253 *
254 * Scanner
255 *
256 * Scanner
257 *
258 * Scanner
259 *
260 * Scanner
261 *
262 * Scanner
263 *
264 * Scanner
265 *
266 * Scanner
267 *
268 * Scanner
269 *
270 * Scanner
271 *
272 * Scanner
273 *
274 * Scanner
275 *
276 * Scanner
277 *
278 * Scanner
279 *
280 * Scanner
281 *
282 * Scanner
283 *
284 * Scanner
285 *
286 * Scanner
287 *
288 * Scanner
289 *
290 * Scanner
291 *
292 * Scanner
293 *
294 * Scanner
295 *
296 * Scanner
297 *
298 * Scanner
299 *
300 * Scanner
301 *
302 * Scanner
303 *
304 * Scanner
305 *
306 * Scanner
307 *
308 * Scanner
309 *
310 * Scanner
311 *
312 * Scanner
313 *
314 * Scanner
315 *
316 * Scanner
317 *
318 * Scanner
319 *
320 * Scanner
321 *
322 * Scanner
323 *
324 * Scanner
325 *
326 * Scanner
327 *
328 * Scanner
329 *
330 * Scanner
331 *
332 * Scanner
333 *
334 * Scanner
335 *
336 * Scanner
337 *
338 * Scanner
339 *
340 * Scanner
341 *
342 * Scanner
343 *
344 * Scanner
345 *
346 * Scanner
347 *
348 * Scanner
349 *
350 * Scanner
351 *
352 * Scanner
353 *
354 * Scanner
355 *
356 * Scanner
357 *
358 * Scanner
359 *
360 * Scanner
361 *
362 * Scanner
363 *
364 * Scanner
365 *
366 * Scanner
367 *
368 * Scanner
369 *
370 * Scanner
371 *
372 * Scanner
373 *
374 * Scanner
375 *
376 * Scanner
377 *
378 * Scanner
379 *
380 * Scanner
381 *
382 * Scanner
383 *
384 * Scanner
385 *
386 * Scanner
387 *
388 * Scanner
389 *
390 * Scanner
391 *
392 * Scanner
393 *
394 * Scanner
395 *
396 * Scanner
397 *
398 * Scanner
399 *
400 * Scanner
401 *
402 * Scanner
403 *
404 * Scanner
405 *
406 * Scanner
407 *
408 * Scanner
409 *
410 * Scanner
411 *
412 * Scanner
413 *
414 * Scanner
415 *
416 * Scanner
417 *
418 * Scanner
419 *
420 * Scanner
421 *
422 * Scanner
423 *
424 * Scanner
425 *
426 * Scanner
427 *
428 * Scanner
429 *
430 * Scanner
431 *
432 * Scanner
433 *
434 * Scanner
435 *
436 * Scanner
437 *
438 * Scanner
439 *
440 * Scanner
441 *
442 * Scanner
443 *
444 * Scanner
445 *
446 * Scanner
447 *
448 * Scanner
449 *
450 * Scanner
451 *
452 * Scanner
453 *
454 * Scanner
455 *
456 * Scanner
457 *
458 * Scanner
459 *
460 * Scanner
461 *
462 * Scanner
463 *
464 * Scanner
465 *
466 * Scanner
467 *
468 * Scanner
469 *
470 * Scanner
471 *
472 * Scanner
473 *
474 * Scanner
475 *
476 * Scanner
477 *
478 * Scanner
479 *
480 * Scanner
481 *
482 * Scanner
483 *
484 * Scanner
485 *
486 * Scanner
487 *
488 * Scanner
489 *
490 * Scanner
491 *
492 * Scanner
493 *
494 * Scanner
495 *
496 * Scanner
497 *
498 * Scanner
499 *
500 * Scanner
501 *
502 * Scanner
503 *
504 * Scanner
505 *
506 * Scanner
507 *
508 * Scanner
509 *
510 * Scanner
511 *
512 * Scanner
513 *
514 * Scanner
515 *
516 * Scanner
517 *
518 * Scanner
519 *
520 * Scanner
521 *
522 * Scanner
523 *
524 * Scanner
525 *
526 * Scanner
527 *
528 * Scanner
529 *
530 * Scanner
531 *
532 * Scanner
533 *
534 * Scanner
535 *
536 * Scanner
537 *
538 * Scanner
539 *
540 * Scanner
541 *
542 * Scanner
543 *
544 * Scanner
545 *
546 * Scanner
547 *
548 * Scanner
549 *
550 * Scanner
551 *
552 * Scanner
553 *
554 * Scanner
555 *
556 * Scanner
557 *
558 * Scanner
559 *
560 * Scanner
561 *
562 * Scanner
563 *
564 * Scanner
565 *
566 * Scanner
567 *
568 * Scanner
569 *
570 * Scanner
571 *
572 * Scanner
573 *
574 * Scanner
575 *
576 * Scanner
577 *
578 * Scanner
579 *
580 * Scanner
581 *
582 * Scanner
583 *
584 * Scanner
585 *
586 * Scanner
587 *
588 * Scanner
589 *
590 * Scanner
591 *
592 * Scanner
593 *
594 * Scanner
595 *
596 * Scanner
597 *
598 * Scanner
599 *
600 * Scanner
601 *
602 * Scanner
603 *
604 * Scanner
605 *
606 * Scanner
607 *
608 * Scanner
609 *
610 * Scanner
611 *
612 * Scanner
613 *
614 * Scanner
615 *
616 * Scanner
617 *
618 * Scanner
619 *
620 * Scanner
621 *
622 * Scanner
623 *
624 * Scanner
625 *
626 * Scanner
627 *
628 * Scanner
629 *
630 * Scanner
631 *
632 * Scanner
633 *
634 * Scanner
635 *
636 * Scanner
637 *
638 * Scanner
639 *
640 * Scanner
641 *
642 * Scanner
643 *
644 * Scanner
645 *
646 * Scanner
647 *
648 * Scanner
649 *
650 * Scanner
651 *
652 * Scanner
653 *
654 * Scanner
655 *
656 * Scanner
657 *
658 * Scanner
659 *
660 * Scanner
661 *
662 * Scanner
663 *
664 * Scanner
665 *
666 * Scanner
667 *
668 * Scanner
669 *
670 * Scanner
671 *
672 * Scanner
673 *
674 * Scanner
675 *
676 * Scanner
677 *
678 * Scanner
679 *
680 * Scanner
681 *
682 * Scanner
683 *
684 * Scanner
685 *
686 * Scanner
687 *
688 * Scanner
689 *
690 * Scanner
691 *
692 * Scanner
693 *
694 * Scanner
695 *
696 * Scanner
697 *
698 * Scanner
699 *
700 * Scanner
701 *
702 * Scanner
703 *
704 * Scanner
705 *
706 * Scanner
707 *
708 * Scanner
709 *
710 * Scanner
711 *
712 * Scanner
713 *
714 * Scanner
715 *
716 * Scanner
717 *
718 * Scanner
719 *
720 * Scanner
721 *
722 * Scanner
723 *
724 * Scanner
725 *
726 * Scanner
727 *
728 * Scanner
729 *
730 * Scanner
731 *
732 * Scanner
733 *
734 * Scanner
735 *
736 * Scanner
737 *
738 * Scanner
739 *
740 * Scanner
741 *
742 * Scanner
743 *
744 * Scanner
745 *
746 * Scanner
747 *
748 * Scanner
749 *
750 * Scanner
751 *
752 * Scanner
753 *
754 * Scanner
755 *
756 * Scanner
757 *
758 * Scanner
759 *
760 * Scanner
761 *
762 * Scanner
763 *
764 * Scanner
765 *
766 * Scanner
767 *
768 * Scanner
769 *
770 * Scanner
771 *
772 * Scanner
773 *
774 * Scanner
775 *
776 * Scanner
777 *
778 * Scanner
779 *
780 * Scanner
781 *
782 * Scanner
783 *
784 * Scanner
785 *
786 * Scanner
787 *
788 * Scanner
789 *
790 * Scanner
791 *
792 * Scanner
793 *
794 * Scanner
795 *
796 * Scanner
797 *
798 * Scanner
799 *
800 * Scanner
801 *
802 * Scanner
803 *
804 * Scanner
805 *
806 * Scanner
807 *
808 * Scanner
809 *
810 * Scanner
811 *
812 * Scanner
813 *
814 * Scanner
815 *
816 * Scanner
817 *
818 * Scanner
819 *
820 * Scanner
821 *
822 * Scanner
823 *
824 * Scanner
825 *
826 * Scanner
827 *
828 * Scanner
829 *
830 * Scanner
831 *
832 * Scanner
833 *
834 * Scanner
835 *
836 * Scanner
837 *
838 * Scanner
839 *
840 * Scanner
841 *
842 * Scanner
843 *
844 * Scanner
845 *
846 * Scanner
847 *
848 * Scanner
849 *
850 * Scanner
851 *
852 * Scanner
853 *
854 * Scanner
855 *
856 * Scanner
857 *
858 * Scanner
859 *
860 * Scanner
861 *
862 * Scanner
863 *
864 * Scanner
865 *
866 * Scanner
867 *
868 * Scanner
869 *
870 * Scanner
871 *
872 * Scanner
873 *
874 * Scanner
875 *
876 * Scanner
877 *
878 * Scanner
879 *
880 * Scanner
881 *
882 * Scanner
883 *
884 * Scanner
885 *
886 * Scanner
887 *
888 * Scanner
889 *
890 * Scanner
891 *
892 * Scanner
893 *
894 * Scanner
895 *
896 * Scanner
897 *
898 * Scanner
899 *
900 * Scanner
901 *
902 * Scanner
903 *
904 * Scanner
905 *
906 * Scanner
907 *
908 * Scanner
909 *
910 * Scanner
911 *
912 * Scanner
913 *
914 * Scanner
915 *
916 * Scanner
917 *
918 * Scanner
919 *
920 * Scanner
921 *
922 * Scanner
923 *
924 * Scanner
925 *
926 * Scanner
927 *
928 * Scanner
929 *
930 * Scanner
931 *
932 * Scanner
933 *
934 * Scanner
935 *
936 * Scanner
937 *
938 * Scanner
939 *
940 * Scanner
941 *
942 * Scanner
943 *
944 * Scanner
945 *
946 * Scanner
947 *
948 * Scanner
949 *
950 * Scanner
951 *
952 * Scanner
953 *
954 * Scanner
955 *
956 * Scanner
957 *
958 * Scanner
959 *
960 * Scanner
961 *
962 * Scanner
963 *
964 * Scanner
965 *
966 * Scanner
967 *
968 * Scanner
969 *
970 * Scanner
971 *
972 * Scanner
973 *
974 * Scanner
975 *
976 * Scanner
977 *
978 * Scanner
979 *
980 * Scanner
981 *
982 * Scanner
983 *
984 * Scanner
985 *
986 * Scanner
987 *
988 * Scanner
989 *
990 * Scanner
991 *
992 * Scanner
993 *
994 * Scanner
995 *
996 * Scanner
997 *
998 * Scanner
999 *
1000 * Scanner
1001 *
1002 * Scanner
1003 *
1004 * Scanner
1005 *
1006 * Scanner
1007 *
1008 * Scanner
1009 *
1010 * Scanner
1011 *
1012 * Scanner
1013 *
1014 * Scanner
1015 *
1016 * Scanner
1017 *
1018 * Scanner
1019 *
1020 * Scanner
1021 *
1022 * Scanner
1023 *
1024 * Scanner
1025 *
1026 * Scanner
1027 *
1028 * Scanner
1029 *
1030 * Scanner
1031 *
1032 * Scanner
1033 *
1034 * Scanner
1035 *
1036 * Scanner
1037 *
1038 * Scanner
1039 *
1040 * Scanner
1041 *
1042 * Scanner
1043 *
1044 * Scanner
1045 *
1046 * Scanner
1047 *
1048 * Scanner
1049 *
1050 * Scanner
1051 *
1052 * Scanner
1053 *
1054 * Scanner
1055 *
1056 * Scanner
1057 *
1058 * Scanner
1059 *
1060 * Scanner
1061 *
1062 * Scanner
1063 *
1064 * Scanner
1065 *
1066 * Scanner
1067 *
1068 * Scanner
1069 *
1070 * Scanner
1071 *
1072 * Scanner
1073 *
1074 * Scanner
1075 *
1076 * Scanner
1077 *
1078 * Scanner
1079 *
1080 * Scanner
1081 *
1082 * Scanner
1083 *
1084 * Scanner
1085 *
1086 * Scanner
1087 *
1088 * Scanner
1089 *
1090 * Scanner
1091 *
1092 * Scanner
1093 *
1094 * Scanner
1095 *
1096 * Scanner
1097 *
1098 * Scanner
1099 *
1100 * Scanner
1101 *
1102 * Scanner
1103 *
1104 * Scanner
1105 *
1106 * Scanner
1107 *
1108 * Scanner
1109 *
1110 * Scanner
1111 *
1112 * Scanner
1113 *
1114 * Scanner
1115 *
1116 * Scanner
1117 *
1118 * Scanner
1119 *
1120 * Scanner
1121 *
1122 * Scanner
1123 *
1124 * Scanner
1125 *
1126 * Scanner
1127 *
1128 * Scanner
1129 *
1130 * Scanner
1131 *
1132 * Scanner
1133 *
1134 * Scanner
1135 *
1136 * Scanner
1137 *
1138 * Scanner
1139 *
1140 * Scanner
1141 *
1142 * Scanner
1143 *
1144 * Scanner
1145 *
1146 * Scanner
1147 *
1148 * Scanner
1149 *
1150 * Scanner
1151 *
1152 * Scanner
1153 *
1154 * Scanner
1155 *
1156 * Scanner
1157 *
1158 * Scanner
1159 *
1160 * Scanner
1161 *
1162 * Scanner
1163 *
1164 * Scanner
1165 *
1166 * Scanner
1167 *
1168 * Scanner
1169 *
1170 * Scanner
1171 *
1172 * Scanner
1173 *
1174 * Scanner
1175 *
1176 * Scanner
1177 *
1178 * Scanner
1179 *
1180 * Scanner
1181 *
1182 * Scanner
1183 *
1184 * Scanner
1185 *
1186 * Scanner
1187 *
1188 * Scanner
1189 *
1190 * Scanner
1191 *
1192 * Scanner
1193 *
1194 * Scanner
1195 *
1196 * Scanner
1197 *
1198 * Scanner
1199 *
1200 * Scanner
1201 *
1202 * Scanner
1203 *
1204 * Scanner
1205 *
1206 * Scanner
1207 *
1208 * Scanner
1209 *
1210 * Scanner
1211 *
1212 * Scanner
1213 *
1214 * Scanner
1215 *
1216 * Scanner
1217 *
1218 * Scanner
1219 *
1220 * Scanner
1221 *
1222 * Scanner
1223 *
1224 * Scanner
1225 *
1226 * Scanner
1227 *
1228 * Scanner
1229 *
1230 * Scanner
1231 *
1232 * Scanner
1233 *
1234 * Scanner
1235 *
1236 * Scanner
1237 *
1238 * Scanner
1239 *
1240 * Scanner
1241 *
1242 * Scanner
1243 *
1244 * Scanner
1245 *
1246 * Scanner
1247 *
1248 * Scanner
1249 *
1250 * Scanner
1251 *
1252 * Scanner
1253 *
1254 * Scanner
1255 *
1256 * Scanner
1257 *
1258 * Scanner
1259 *
1260 * Scanner
1261 *
1262 * Scanner
1263 *
1264 * Scanner
1265 *
1266 * Scanner
1267 *
1268 * Scanner
1269 *
1270 * Scanner
1271 *
1272 * Scanner
1273 *
1274 * Scanner
1275 *
1276 * Scanner
1277 *
1278 * Scanner
1279 *
1280 * Scanner
1281 *
1282 * Scanner
1283 *
1284 * Scanner
1285 *
1286 * Scanner
1287 *
1288 * Scanner
1289 *
1290 * Scanner
1291 *
1292 * Scanner
1293 *
1294 * Scanner
1295 *
1296 * Scanner
1297 *
1298 * Scanner
1299 *
1300 * Scanner
1301 *
1302 * Scanner
1303 *
1304 * Scanner
1305 *
1306 * Scanner
1307 *
1308 * Scanner
1309 *
1310 * Scanner
1311 *
1312 * Scanner
1313 *
1314 * Scanner
1315 *
1316 * Scanner
1317 *
1318 * Scanner
1319 *
1320 * Scanner
1321 *
1322 * Scanner
1323 *
1324 * Scanner
1325 *
1326 * Scanner
1327 *
1328 * Scanner
1329 *
1330 * Scanner
1331 *
1332 * Scanner
1333 *
1334 * Scanner
1335 *
1336 * Scanner
1337 *
1338 * Scanner
1339 *
1340 * Scanner
1341 *
1342 * Scanner
1343 *
1344 * Scanner
1345 *
1346 * Scanner
1347 *
1348 * Scanner
1349 *
1350 * Scanner
1351 *
1352 * Scanner
1353 *
1354 * Scanner
1355 *
1356 * Scanner
1357 *
1358 * Scanner
1359 *
1360 * Scanner
1361 *
1362 * Scanner
1363 *
1364 * Scanner
1365 *
1366 * Scanner
1367 *
1368 * Scanner
13
```

# Remember ADTs?



# Recall from L6: Wait, ADT? Interfaces?

- **Abstract Data Type (ADT):** A *description of the idea* of a data structure including what operations are available on it and how those operations should behave. For example, the English explanation of what a list should be.

- **Interface:** Java construct that lets programmers *specify what methods a class should have*. For example the `List` interface in java.

- **Implementation:** *Concrete code* that meets the specified interface. For example, the `ArrayList` and `LinkedList` classes that implement the `List` interface.

```
List<String> myList = new ArrayList<>();
```

# Interfaces

**Interfaces** serve as a sort of “contract” – in order for a class to implement an interface, it must fulfill the contract requirements.

The contract requirements are certain methods that the class must implement.

# building noun

build·ing ('bil-dɪŋ)

Synonyms of *building* >

1 : a usually roofed and walled structure **built** for permanent use (as for a dwelling)

Our "ADT"



Our "Interface"



Our "Implementation"

# Lists

One ADT we've talked a lot about in this course is a list.

Within Java, there exists a `List` interface – its contract includes methods like:

`add`, `clear`, `contains`, `get`, `isEmpty`, `size`

There's also an `ArrayList` class (implementation)

To get the certificate, it must include all these methods (and any others the `List` interface specifies)

# Interfaces vs. Implementation

Interfaces require certain methods, but they do not say anything about how those methods should be implemented – that's up to the class! 🏆

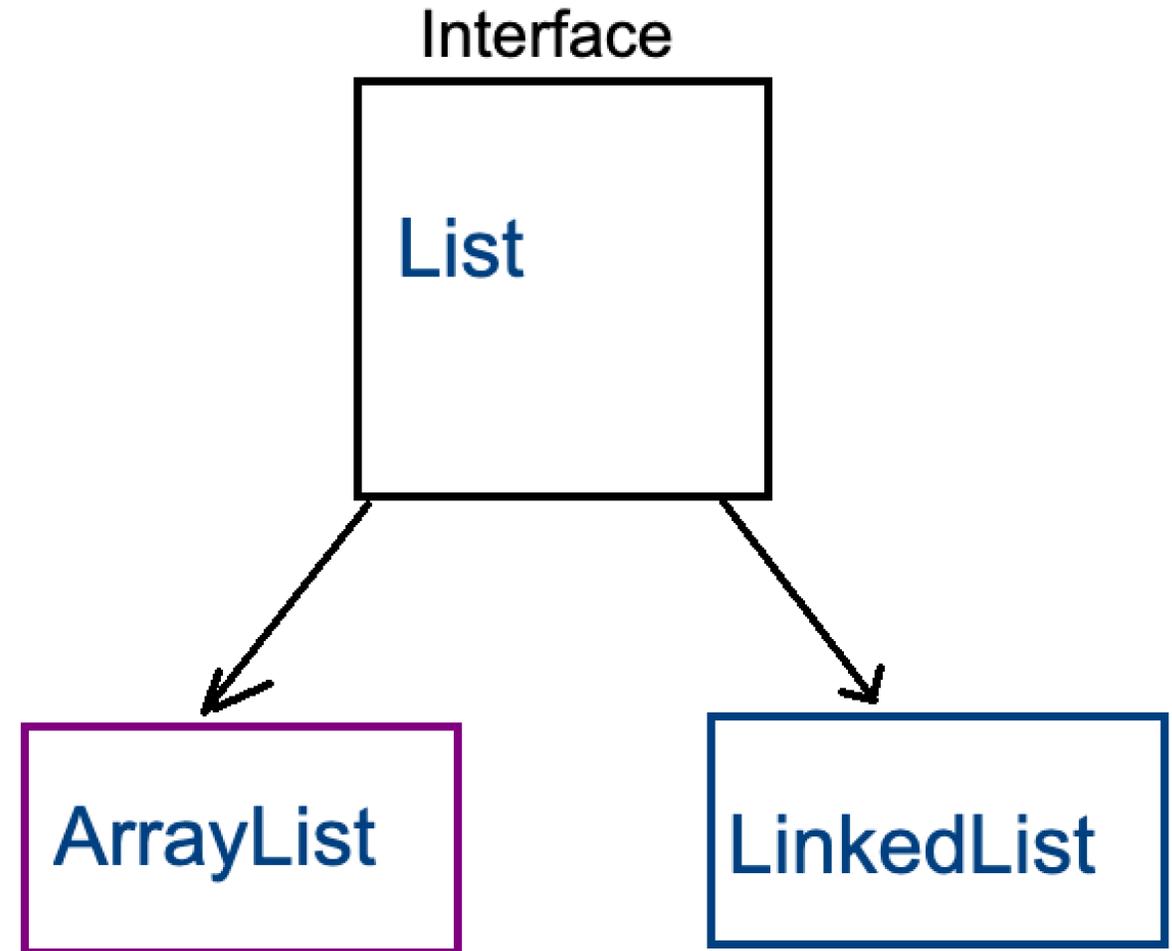
# Interfaces vs. Implementation

List is an interface

ArrayList is a class that implements the List interface

LinkedList is a class that implements the List interface

...



# Why interfaces?

```
public static void fill(WaterBottle w) {...}
```

This method only accepts a WaterBottle!

# Why interfaces?

Flexibility



```
public static void fill(Container c) {...}
```

This method can accept either a:

- WaterBottle
- Brita
- Tub or
- Any other class that implements Container!

# Why interfaces?

## Flexibility



```
public static void method(Set<String> s) {...}
```

This method can accept either a:

- `HashSet<String>` or
- `TreeSet<String>` or
- Any other class that implements `Set` and whose element type is `String`!

# Why interfaces?

## Abstraction

Interfaces also support *abstraction*  
(the separation of ideas from details)



# Lecture Outline

- Announcements
- Interfaces Review
- **More Shapes!**  [Ed lesson link](#)
- Comparable

# Classes can Implement Multiple Interfaces

A class can implement multiple interfaces – it's like one person signing multiple contracts!

If a class implements an interface A and an interface B, it'll just have to include all of A's required methods along with all of B's required methods

# Classes can Implement Multiple Interfaces

```
public interface UWMerch {  
    public String getMaterial();  
}
```

```
public class WaterBottle implements Container, UWMerch {  
    ...  
    public String getMaterial() {  
        return this.material;  
    }  
}
```

But WaterBottle would have to implement:

- fill from Container AND getMaterial from UWMerch

# An interface can extend another

You can have one interface extend another

So if **public interface A extends B**, then any class that implements A must include all the methods in A's interface and all the methods in B's interface

# An interface can extend another

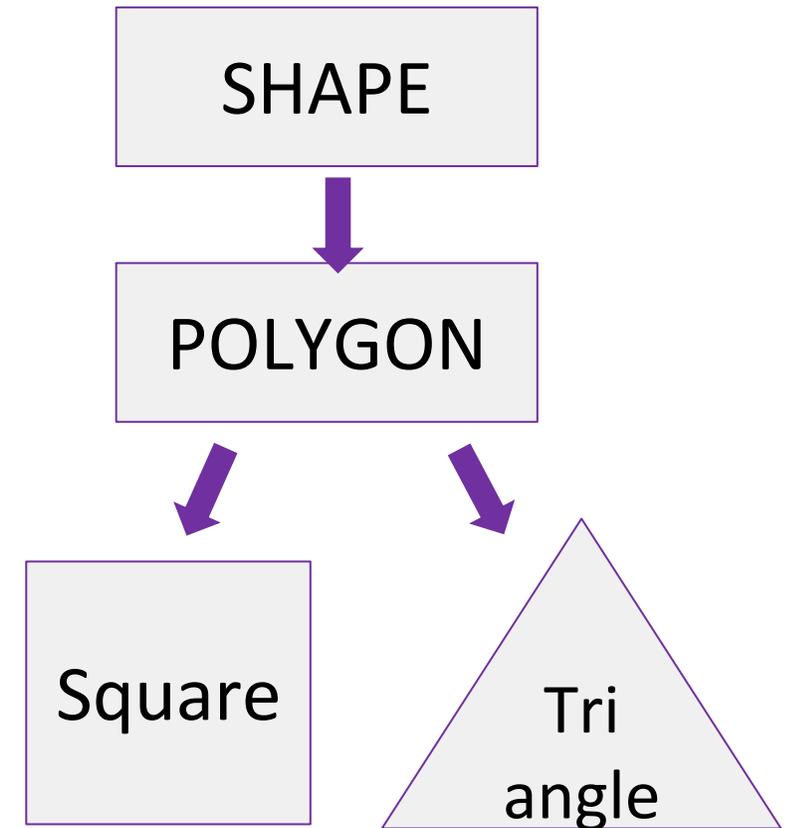
We can write another interface:

**Polygon** that extends **Shape**

- “**Polygon** is a **Shape**”

Make modifications such that:

- Square is a **Polygon** (and **Shape**)
- Triangle is a **Polygon** (and **Shape**)
- Circle is a **Shape** (but not a **Polygon**)



# Lecture Outline

- Announcements
- Interfaces Review
- More Shapes!
- **Comparable** ◀

# Recall the Student / Course Example last lecture

Course stored a field

```
private List<Student> roster;
```

Why not use a Set to store the students?...

Seems like a great idea (no duplicates, not worried about keeping a specific order or indexing into it) but ... Java reasons:

- HashSet won't work because of lack of hashCode() implementation
- TreeSet won't work because... what does it mean to "sort" Students?

# Comparable

TreeSet uses an **interface** called `Comparable<E>` to know how to sort its elements!

Only has one required method:

```
public int compareTo(E other)
```

Its return value is:

- < 0 if this is “less than” other
- 0 if this is equal to other
- > 0 if this is “greater than” other