

Accueil ► SI - Sciences Informatiques ► SI3 ► Intro POO ► Stuff to do - unevaluated ► More about inheritance

**Commencé le** jeudi 23 novembre 2017, 10:21

**État** Terminé

**Terminé le** jeudi 23 novembre 2017, 11:01

**Temps mis** 39 min 44 s

**En retard** 39 min 43 s

**Points** 11,50/12,00

**Note** 19,17 sur 20,00 (96%)

#### Description

The last exercise was a bit unsatisfying. All the code worked (hopefully), but the result displayed was not what we wanted.

We traced the problem to the fact that **Post#display** only had access to the attributes of **Post** and not to the attributes specific to the subclasses, like **message**, **fileName**, **caption**, etc. Well, we know that methods in a superclass don't have access to anything in their subclasses, whereas stuff in superclasses can be accessible from subclasses.

So, move **Post#display** back into the subclasses **MessagePost** and **PhotoPost**. You may have to change the method to get these classes to compile. Do not change **NewsFeed**, just (try to) recompile it.

#### Question 1

Correct

Note de 1,00 sur 1,00

Previously, you were asked to move the display method from the superclass **Post** back down to its subclasses **MessagePost** and **PhotoPost**.

How did recompiling **NewsFeed** work out for you?

Veillez choisir une réponse :

- ☐ a. Worked just fine - compiled and executed.
- ☒ b. Compiler error. That's normal: **NewsFeed** only knows about the class **Post** and **Post#display** no longer exists. You *should* get a compilation error. ✓

Vérifier

Your answer is correct.

**Correct**

Note pour cet envoi : 1,00/1,00.

#### Description

We seem to have a problem:

- if **display** is only in the superclass **Post**, the code compiles but doesn't do the right thing;
- if **display** is in the subclasses instead, **NewsFeed** doesn't even compile.

What to do, what to do?

#### Question 2

Correct

Note de 4,50 sur 5,00

Looks like the method **display** has to be in the superclass *and* in the subclasses. Try that.

Remember:

- **Post#display** has to display the attributes of **Post**;
- **MessagePost#display** has to display the attributes of **MessagePost**;
- **PhotoPost#display** has to display the attributes of **PhotoPost**.

There are two more issues. First, variables are declared as being of the superclass type, but are created as subclass objects, eg,

```
Post message = new MessagePost(...);
```

What happens here in the following expression?

```
message.display();
```

Does it compile? Yes, because **Post#display** is a declared method. But that's not the method that gets executed - through the miracle of polymorphism, it's actually **MessagePost#display** which is executed.

Second, how to get **Post#display** to execute when we've just said that it's **MessagePost#display** which is executed? *Both* of them have to execute to display the attributes of both the superclass and the subclasses. So you'll have to call the superclass method from the subclass method, something like, for **MessagePost**:

```
public void display() {
    // maybe some code
    super.display();
    // maybe some more code
}
```

The **super.display()** calls the method of the superclass, ie, **Post#display**. For testing, put **super.display()** as the *last* expression in your method definition.

Submit all four classes (**Post**, **MessagePost**, **PhotoPost**, **NewsFeed**) into the Answer box.

Note that all the superclass and subclass attributes are displayed, but not necessarily in the original order.

For example:

Test	Result
<pre>Post mp = new MessagePost("Leonardo da Vinci", "Code this...!"); mp.display();</pre>	<pre>Code this...! Leonardo da Vinci 0 seconds ago No comments.</pre>

Réponse:

```
1 import java.util.ArrayList;
2 import java.util.List;
3 import java.util.stream.*;
4
5 /**
6  * This class stores information about a post in a social network news feed. The
7  * main part of the post consists of a (possibly multi-line) text message. Other
8  * data, such as author and time, are also stored.
9  *
10 * @author Michael Kolling and David J. Barnes
11 * @version 0.3
12 */
13 public class MessagePost extends Post {
14     private String message; // an arbitrarily long, multi-line message
15
16     /**
17      * Constructor for objects of class MessagePost.
18      *
19      * @param author the author of the post
20      * @param message the message of the post
21      * @param time the time of the post
22      */
23     public MessagePost(String author, String message, long time) {
24         super(author, time);
25         this.message = message;
26     }
27
28     /**
29      * Returns the message of this post.
30      *
31      * @return the message of this post
32      */
33     public String getMessage() {
34         return message;
35     }
36
37     /**
38      * Returns the time of this post.
39      *
40      * @return the time of this post
41      */
42     public long getTime() {
43         return time;
44     }
45
46     /**
47      * Returns the author of this post.
48      *
49      * @return the author of this post
50      */
51     public String getAuthor() {
52         return author;
53     }
54
55     /**
56      * Returns the time of this post.
57      *
58      * @return the time of this post
59      */
60     public long getTime() {
61         return time;
62     }
63
64     /**
65      * Returns the author of this post.
66      *
67      * @return the author of this post
68      */
69     public String getAuthor() {
70         return author;
71     }
72
73     /**
74      * Returns the time of this post.
75      *
76      * @return the time of this post
77      */
78     public long getTime() {
79         return time;
80     }
81
82     /**
83      * Returns the author of this post.
84      *
85      * @return the author of this post
86      */
87     public String getAuthor() {
88         return author;
89     }
90
91     /**
92      * Returns the time of this post.
93      *
94      * @return the time of this post
95      */
96     public long getTime() {
97         return time;
98     }
99
100    /**
101     * Returns the author of this post.
102     *
103     * @return the author of this post
104     */
105    public String getAuthor() {
106        return author;
107    }
108
109    /**
110     * Returns the time of this post.
111     *
112     * @return the time of this post
113     */
114    public long getTime() {
115        return time;
116    }
117
118    /**
119     * Returns the author of this post.
120     *
121     * @return the author of this post
122     */
123    public String getAuthor() {
124        return author;
125    }
126
127    /**
128     * Returns the time of this post.
129     *
130     * @return the time of this post
131     */
132    public long getTime() {
133        return time;
134    }
135
136    /**
137     * Returns the author of this post.
138     *
139     * @return the author of this post
140     */
141    public String getAuthor() {
142        return author;
143    }
144
145    /**
146     * Returns the time of this post.
147     *
148     * @return the time of this post
149     */
150    public long getTime() {
151        return time;
152    }
153
154    /**
155     * Returns the author of this post.
156     *
157     * @return the author of this post
158     */
159    public String getAuthor() {
160        return author;
161    }
162
163    /**
164     * Returns the time of this post.
165     *
166     * @return the time of this post
167     */
168    public long getTime() {
169        return time;
170    }
171
172    /**
173     * Returns the author of this post.
174     *
175     * @return the author of this post
176     */
177    public String getAuthor() {
178        return author;
179    }
180
181    /**
182     * Returns the time of this post.
183     *
184     * @return the time of this post
185     */
186    public long getTime() {
187        return time;
188    }
189
190    /**
191     * Returns the author of this post.
192     *
193     * @return the author of this post
194     */
195    public String getAuthor() {
196        return author;
197    }
198
199    /**
200     * Returns the time of this post.
201     *
202     * @return the time of this post
203     */
204    public long getTime() {
205        return time;
206    }
207
208    /**
209     * Returns the author of this post.
210     *
211     * @return the author of this post
212     */
213    public String getAuthor() {
214        return author;
215    }
216
217    /**
218     * Returns the time of this post.
219     *
220     * @return the time of this post
221     */
222    public long getTime() {
223        return time;
224    }
225
226    /**
227     * Returns the author of this post.
228     *
229     * @return the author of this post
230     */
231    public String getAuthor() {
232        return author;
233    }
234
235    /**
236     * Returns the time of this post.
237     *
238     * @return the time of this post
239     */
240    public long getTime() {
241        return time;
242    }
243
244    /**
245     * Returns the author of this post.
246     *
247     * @return the author of this post
248     */
249    public String getAuthor() {
250        return author;
251    }
252
253    /**
254     * Returns the time of this post.
255     *
256     * @return the time of this post
257     */
258    public long getTime() {
259        return time;
260    }
261
262    /**
263     * Returns the author of this post.
264     *
265     * @return the author of this post
266     */
267    public String getAuthor() {
268        return author;
269    }
270
271    /**
272     * Returns the time of this post.
273     *
274     * @return the time of this post
275     */
276    public long getTime() {
277        return time;
278    }
279
280    /**
281     * Returns the author of this post.
282     *
283     * @return the author of this post
284     */
285    public String getAuthor() {
286        return author;
287    }
288
289    /**
290     * Returns the time of this post.
291     *
292     * @return the time of this post
293     */
294    public long getTime() {
295        return time;
296    }
297
298    /**
299     * Returns the author of this post.
300     *
301     * @return the author of this post
302     */
303    public String getAuthor() {
304        return author;
305    }
306
307    /**
308     * Returns the time of this post.
309     *
310     * @return the time of this post
311     */
312    public long getTime() {
313        return time;
314    }
315
316    /**
317     * Returns the author of this post.
318     *
319     * @return the author of this post
320     */
321    public String getAuthor() {
322        return author;
323    }
324
325    /**
326     * Returns the time of this post.
327     *
328     * @return the time of this post
329     */
330    public long getTime() {
331        return time;
332    }
333
334    /**
335     * Returns the author of this post.
336     *
337     * @return the author of this post
338     */
339    public String getAuthor() {
340        return author;
341    }
342
343    /**
344     * Returns the time of this post.
345     *
346     * @return the time of this post
347     */
348    public long getTime() {
349        return time;
350    }
351
352    /**
353     * Returns the author of this post.
354     *
355     * @return the author of this post
356     */
357    public String getAuthor() {
358        return author;
359    }
360
361    /**
362     * Returns the time of this post.
363     *
364     * @return the time of this post
365     */
366    public long getTime() {
367        return time;
368    }
369
370    /**
371     * Returns the author of this post.
372     *
373     * @return the author of this post
374     */
375    public String getAuthor() {
376        return author;
377    }
378
379    /**
380     * Returns the time of this post.
381     *
382     * @return the time of this post
383     */
384    public long getTime() {
385        return time;
386    }
387
388    /**
389     * Returns the author of this post.
390     *
391     * @return the author of this post
392     */
393    public String getAuthor() {
394        return author;
395    }
396
397    /**
398     * Returns the time of this post.
399     *
400     * @return the time of this post
401     */
402    public long getTime() {
403        return time;
404    }
405
406    /**
407     * Returns the author of this post.
408     *
409     * @return the author of this post
410     */
411    public String getAuthor() {
412        return author;
413    }
414
415    /**
416     * Returns the time of this post.
417     *
418     * @return the time of this post
419     */
420    public long getTime() {
421        return time;
422    }
423
424    /**
425     * Returns the author of this post.
426     *
427     * @return the author of this post
428     */
429    public String getAuthor() {
430        return author;
431    }
432
433    /**
434     * Returns the time of this post.
435     *
436     * @return the time of this post
437     */
438    public long getTime() {
439        return time;
440    }
441
442    /**
443     * Returns the author of this post.
444     *
445     * @return the author of this post
446     */
447    public String getAuthor() {
448        return author;
449    }
450
451    /**
452     * Returns the time of this post.
453     *
454     * @return the time of this post
455     */
456    public long getTime() {
457        return time;
458    }
459
460    /**
461     * Returns the author of this post.
462     *
463     * @return the author of this post
464     */
465    public String getAuthor() {
466        return author;
467    }
468
469    /**
470     * Returns the time of this post.
471     *
472     * @return the time of this post
473     */
474    public long getTime() {
475        return time;
476    }
477
478    /**
479     * Returns the author of this post.
480     *
481     * @return the author of this post
482     */
483    public String getAuthor() {
484        return author;
485    }
486
487    /**
488     * Returns the time of this post.
489     *
490     * @return the time of this post
491     */
492    public long getTime() {
493        return time;
494    }
495
496    /**
497     * Returns the author of this post.
498     *
499     * @return the author of this post
500     */
501    public String getAuthor() {
502        return author;
503    }
504
505    /**
506     * Returns the time of this post.
507     *
508     * @return the time of this post
509     */
510    public long getTime() {
511        return time;
512    }
513
514    /**
515     * Returns the author of this post.
516     *
517     * @return the author of this post
518     */
519    public String getAuthor() {
520        return author;
521    }
522
523    /**
524     * Returns the time of this post.
525     *
526     * @return the time of this post
527     */
528    public long getTime() {
529        return time;
530    }
531
532    /**
533     * Returns the author of this post.
534     *
535     * @return the author of this post
536     */
537    public String getAuthor() {
538        return author;
539    }
540
541    /**
542     * Returns the time of this post.
543     *
544     * @return the time of this post
545     */
546    public long getTime() {
547        return time;
548    }
549
550    /**
551     * Returns the author of this post.
552     *
553     * @return the author of this post
554     */
555    public String getAuthor() {
556        return author;
557    }
558
559    /**
560     * Returns the time of this post.
561     *
562     * @return the time of this post
563     */
564    public long getTime() {
565        return time;
566    }
567
568    /**
569     * Returns the author of this post.
570     *
571     * @return the author of this post
572     */
573    public String getAuthor() {
574        return author;
575    }
576
577    /**
578     * Returns the time of this post.
579     *
580     * @return the time of this post
581     */
582    public long getTime() {
583        return time;
584    }
585
586    /**
587     * Returns the author of this post.
588     *
589     * @return the author of this post
590     */
591    public String getAuthor() {
592        return author;
593    }
594
595    /**
596     * Returns the time of this post.
597     *
598     * @return the time of this post
599     */
600    public long getTime() {
601        return time;
602    }
603
604    /**
605     * Returns the author of this post.
606     *
607     * @return the author of this post
608     */
609    public String getAuthor() {
610        return author;
611    }
612
613    /**
614     * Returns the time of this post.
615     *
616     * @return the time of this post
617     */
618    public long getTime() {
619        return time;
620    }
621
622    /**
623     * Returns the author of this post.
624     *
625     * @return the author of this post
626     */
627    public String getAuthor() {
628        return author;
629    }
630
631    /**
632     * Returns the time of this post.
633     *
634     * @return the time of this post
635     */
636    public long getTime() {
637        return time;
638    }
639
640    /**
641     * Returns the author of this post.
642     *
643     * @return the author of this post
644     */
645    public String getAuthor() {
646        return author;
647    }
648
649    /**
650     * Returns the time of this post.
651     *
652     * @return the time of this post
653     */
654    public long getTime() {
655        return time;
656    }
657
658    /**
659     * Returns the author of this post.
660     *
661     * @return the author of this post
662     */
663    public String getAuthor() {
664        return author;
665    }
666
667    /**
668     * Returns the time of this post.
669     *
670     * @return the time of this post
671     */
672    public long getTime() {
673        return time;
674    }
675
676    /**
677     * Returns the author of this post.
678     *
679     * @return the author of this post
680     */
681    public String getAuthor() {
682        return author;
683    }
684
685    /**
686     * Returns the time of this post.
687     *
688     * @return the time of this post
689     */
690    public long getTime() {
691        return time;
692    }
693
694    /**
695     * Returns the author of this post.
696     *
697     * @return the author of this post
698     */
699    public String getAuthor() {
700        return author;
701    }
702
703    /**
704     * Returns the time of this post.
705     *
706     * @return the time of this post
707     */
708    public long getTime() {
709        return time;
710    }
711
712    /**
713     * Returns the author of this post.
714     *
715     * @return the author of this post
716     */
717    public String getAuthor() {
718        return author;
719    }
720
721    /**
722     * Returns the time of this post.
723     *
724     * @return the time of this post
725     */
726    public long getTime() {
727        return time;
728    }
729
730    /**
731     * Returns the author of this post.
732     *
733     * @return the author of this post
734     */
735    public String getAuthor() {
736        return author;
737    }
738
739    /**
740     * Returns the time of this post.
741     *
742     * @return the time of this post
743     */
744    public long getTime() {
745        return time;
746    }
747
748    /**
749     * Returns the author of this post.
750     *
751     * @return the author of this post
752     */
753    public String getAuthor() {
754        return author;
755    }
756
757    /**
758     * Returns the time of this post.
759     *
760     * @return the time of this post
761     */
762    public long getTime() {
763        return time;
764    }
765
766    /**
767     * Returns the author of this post.
768     *
769     * @return the author of this post
770     */
771    public String getAuthor() {
772        return author;
773    }
774
775    /**
776     * Returns the time of this post.
777     *
778     * @return the time of this post
779     */
780    public long getTime() {
781        return time;
782    }
783
784    /**
785     * Returns the author of this post.
786     *
787     * @return the author of this post
788     */
789    public String getAuthor() {
790        return author;
791    }
792
793    /**
794     * Returns the time of this post.
795     *
796     * @return the time of this post
797     */
798    public long getTime() {
799        return time;
800    }
801
802    /**
803     * Returns the author of this post.
804     *
805     * @return the author of this post
806     */
807    public String getAuthor() {
808        return author;
809    }
810
811    /**
812     * Returns the time of this post.
813     *
814     * @return the time of this post
815     */
816    public long getTime() {
817        return time;
818    }
819
820    /**
821     * Returns the author of this post.
822     *
823     * @return the author of this post
824     */
825    public String getAuthor() {
826        return author;
827    }
828
829    /**
830     * Returns the time of this post.
831     *
832     * @return the time of this post
833     */
834    public long getTime() {
835        return time;
836    }
837
838    /**
839     * Returns the author of this post.
840     *
841     * @return the author of this post
842     */
843    public String getAuthor() {
844        return author;
845    }
846
847    /**
848     * Returns the time of this post.
849     *
850     * @return the time of this post
851     */
852    public long getTime() {
853        return time;
854    }
855
856    /**
857     * Returns the author of this post.
858     *
859     * @return the author of this post
860     */
861    public String getAuthor() {
862        return author;
863    }
864
865    /**
866     * Returns the time of this post.
867     *
868     * @return the time of this post
869     */
870    public long getTime() {
871        return time;
872    }
873
874    /**
875     * Returns the author of this post.
876     *
877     * @return the author of this post
878     */
879    public String getAuthor() {
880        return author;
881    }
882
883    /**
884     * Returns the time of this post.
885     *
886     * @return the time of this post
887     */
888    public long getTime() {
889        return time;
890    }
891
892    /**
893     * Returns the author of this post.
894     *
895     * @return the author of this post
896     */
897    public String getAuthor() {
898        return author;
899    }
900
901    /**
902     * Returns the time of this post.
903     *
904     * @return the time of this post
905     */
906    public long getTime() {
907        return time;
908    }
909
910    /**
911     * Returns the author of this post.
912     *
913     * @return the author of this post
914     */
915    public String getAuthor() {
916        return author;
917    }
918
919    /**
920     * Returns the time of this post.
921     *
922     * @return the time of this post
923     */
924    public long getTime() {
925        return time;
926    }
927
928    /**
929     * Returns the author of this post.
930     *
931     * @return the author of this post
932     */
933    public String getAuthor() {
934        return author;
935    }
936
937    /**
938     * Returns the time of this post.
939     *
940     * @return the time of this post
941     */
942    public long getTime() {
943        return time;
944    }
945
946    /**
947     * Returns the author of this post.
948     *
949     * @return the author of this post
950     */
951    public String getAuthor() {
952        return author;
953    }
954
955    /**
956     * Returns the time of this post.
957     *
958     * @return the time of this post
959     */
960    public long getTime() {
961        return time;
962    }
963
964    /**
965     * Returns the author of this post.
966     *
967     * @return the author of this post
968     */
969    public String getAuthor() {
970        return author;
971    }
972
973    /**
974     * Returns the time of this post.
975     *
976     * @return the time of this post
977     */
978    public long getTime() {
979        return time;
980    }
981
982    /**
983     * Returns the author of this post.
984     *
985     * @return the author of this post
986     */
987    public String getAuthor() {
988        return author;
989    }
990
991    /**
992     * Returns the time of this post.
993     *
994     * @return the time of this post
995     */
996    public long getTime() {
997        return time;
1000    }
1001
1002    /**
1003     * Returns the author of this post.
1004     *
1005     * @return the author of this post
1006     */
1007    public String getAuthor() {
1008        return author;
1009    }
1010
1011    /**
1012     * Returns the time of this post.
1013     *
1014     * @return the time of this post
1015     */
1016    public long getTime() {
1017        return time;
1018    }
1019
1020    /**
1021     * Returns the author of this post.
1022     *
1023     * @return the author of this post
1024     */
1025    public String getAuthor() {
1026        return author;
1027    }
1028
1029    /**
1030     * Returns the time of this post.
1031     *
1032     * @return the time of this post
1033     */
1034    public long getTime() {
1035        return time;
1036    }
1037
1038    /**
1039     * Returns the author of this post.
1040     *
1041     * @return the author of this post
1042     */
1043    public String getAuthor() {
1044        return author;
1045    }
1046
1047    /**
1048     * Returns the time of this post.
1049     *
1050     * @return the time of this post
1051     */
1052    public long getTime() {
1053        return time;
1054    }
1055
1056    /**
1057     * Returns the author of this post.
1058     *
1059     * @return the author of this post
1060     */
1061    public String getAuthor() {
1062        return author;
1063    }
1064
1065    /**
1066     * Returns the time of this post.
1067     *
1068     * @return the time of this post
1069     */
1070    public long getTime() {
1071        return time;
1072    }
1073
1074    /**
1075     * Returns the author of this post.
1076     *
1077     * @return the author of this post
1078     */
1079    public String getAuthor() {
1080        return author;
1081    }
1082
1083    /**
1084     * Returns the time of this post.
1085     *
1086     * @return the time of this post
1087     */
1088    public long getTime() {
1089        return time;
1090    }
1091
1092    /**
1093     * Returns the author of this post.
1094     *
1095     * @return the author of this post
1096     */
1097    public String getAuthor() {
1098        return author;
1099    }
1100
1101    /**
1102     * Returns the time of this post.
1103     *
1104     * @return the time of this post
1105     */
1106    public long getTime() {
1107        return time;
1108    }
1109
1110    /**
1111     * Returns the author of this post.
1112     *
1113     * @return the author of this post
1114     */
1115    public String getAuthor() {
1116        return author;
1117    }
1118
1119    /**
1120     * Returns the time of this post.
1121     *
1122     * @return the time of this post
1123     */
1124    public long getTime() {
1125        return time;
1126    }
1127
1128    /**
1129     * Returns the author of this post.
1130     *
1131     * @return the author of this post
1132     */
1133    public String getAuthor() {
1134        return author;
1135    }
1136
1137    /**
1138     * Returns the time of this post.
1139     *
1140     * @return the time of this post
1141     */
1142    public long getTime() {
1143        return time;
1144    }
1145
1146    /**
1147     * Returns the author of this post.
1148     *
1149     * @return the author of this post
1150     */
1151    public String getAuthor() {
1152        return author;
1153    }
1154
1155    /**
1156     * Returns the time of this post.
1157     *
1158     * @return the time of this post
1159     */
1160    public long getTime() {
1161        return time;
1162    }
1163
1164    /**
1165     * Returns the author of this post.
1166     *
1167     * @return the author of this post
1168     */
1169    public String getAuthor() {
1170        return author;
1171    }
1172
1173    /**
1174     * Returns the time of this post.
1175     *
1176     * @return the time of this post
1177     */
1178    public long getTime() {
1179        return time;
1180    }
1181
1182    /**
1183     * Returns the author of this post.
1184     *
1185     * @return the author of this post
1186     */
1187    public String getAuthor() {
1188        return author;
1189    }
1190
1191    /**
1192     * Returns the time of this post.
1193     *
1194     * @return the time of this post
1195     */
1196    public long getTime() {
1197        return time;
1198    }
1199
1200    /**
1201     * Returns the author of this post.
1202     *
1203     * @return the author of this post
1204     */
1205    public String getAuthor() {
1206        return author;
1207    }
1208
1209    /**
1210     * Returns the time of this post.
1211     *
1212     * @return the time of this post
1213     */
1214    public long getTime() {
1215        return time;
1216    }
1217
1218    /**
1219     * Returns the author of this post.
1220     *
1221     * @return the author of this post
1222     */
1223    public String getAuthor() {
1224        return author;
1225    }
1226
1227    /**
1228     * Returns the time of this post.
1229     *
1230     * @return the time of this post
1231     */
1232    public long getTime() {
1233        return time;
1234    }
1235
1236    /**
1237     * Returns the author of this post.
1238     *
1239     * @return the author of this post
1240     */
1241    public String getAuthor() {
1242        return author;
1243    }
1244
1245    /**
1246     * Returns the time of this post.
1247     *
1248     * @return the time of this post
1249     */
1250    public long getTime() {
1251        return time;
1252    }
1253
1254    /**
1255     * Returns the author of this post.
1256     *
1257     * @return the author of this post
1258     */
1259    public String getAuthor() {
1260        return author;
1261    }
1262
1263    /**
1264     * Returns the time of this post.
1265     *
1266     * @return the time of this post
1267     */
1268    public long getTime() {
1269        return time;
1270    }
1271
1272    /**
1273     * Returns the author of this post.
1274     *
1275     * @return the author of this post
1276     */
1277    public String getAuthor() {
1278        return author;
1279    }
1280
1281    /**
1282     * Returns the time of this post.
1283     *
1284     * @return the time of this post
1285     */
1286    public long getTime() {
1287        return time;
1288    }
1289
1290    /**
1291     * Returns the author of this post.
1292     *
1293     * @return the author of this post
1294     */
1295    public String getAuthor() {
1296        return author;
1297    }
1298
1299    /**
1300     * Returns the time of this post.
1301     *
1302     * @return the time of this post
1303     */
1304    public long getTime() {
1305        return time;
1306    }
1307
1308    /**
1309     * Returns the author of this post.
1310     *
1311     * @return the author of this post
1312     */
1313    public String getAuthor() {
1314        return author;
1315    }
1316
1317    /**
1318     * Returns the time of this post.
1319     *
1320     * @return the time of this post
1321     */
1322    public long getTime() {
1323        return time;
1324    }
1325
1326    /**
1327     * Returns the author of this post.
1328     *
1329     * @return the author of this post
1330     */
1331    public String getAuthor() {
1332        return author;
1333    }
1334
1335    /**
1336     * Returns the time of this post.
1337     *
1338     * @return the time of this post
1339     */
1340    public long getTime() {
1341        return time;
1342    }
1343
1344    /**
1345     * Returns the author of this post.
1346     *
1347     * @return the author of this post
1348     */
1349    public String getAuthor() {
1350        return author;
1351    }
1352
1353    /**
1354     * Returns the time of this post.
1355     *
1356     * @return the time of this post
1357     */
1358    public long getTime() {
1359        return time;
1360    }
1361
1362    /**
1363     * Returns the author of this post.
1364     *
1365     * @return the author of this post
1366     */
1367    public String getAuthor() {
1368        return author;
1369    }
1370
1371    /**
1372     * Returns the time of this post.
1373     *
1374     * @return the
```

✓	<pre> MessagePost mp = new MessagePost("Homer Simpson", "Facts are meaningless. You can use facts mp.like(); mp.like(); mp.unlike(); mp.display(); </pre>
✓	<pre> final MessagePost tsmp = new MessagePost("Taylor Twit", "Party Like It's 1989\\n(c)1989 Tayl IntStream.range(0, 12345).forEach(i -&gt; tsmp.addComment("That sucks.")); tsmp.display();  // note that \\n is an artefact of the CodeRunner test // you will use the usual single backslash in your testing code instead </pre>
✓	<pre> MessagePost mp = new MessagePost("Homer Simpson", "Facts are meaningless. You can use facts mp.like(); mp.like(); mp.unlike(); mp.addComment("Doh - go Donald!"); mp.display(); </pre>

Passed all tests! ✓

**Correct**

Points pour cet envoi : 5,00/5,00. En tenant compte des tentatives précédentes, cela donne **4,50/5,00**.

### Question 3

Correct

Note de 4,00 sur  
4,00

The **PhotoPost** class follows the same schema as the **MessagePost**.

For testing, put **super.display()** as the *last* expression in your **PhotoPost#display** method definition.

Submit all four classes into Answer.

Note that all the superclass and subclass attributes are displayed, but not necessarily in the original order.

Réponse:

```

1 package network.v3;
2
3 import java.util.ArrayList;
4
5 /**
6  * This class stores information about a post in a social network news feed. The
7  * main part of the post consists of a (possibly multi-line) text message. Other
8  * data, such as author and time, are also stored.
9  *
10 * @author Michael Kolling and David J. Barnes
11 * @version 0.3
12 */
13 public class MessagePost extends Post {
14     private String message; // an arbitrarily long, multi-line message
15
16     /**
17      * Constructor for objects of class MessagePost.
18      *
19      * @param author the author of the post
20      * @param message the message of the post
21      * @param time the time of the post
22      */
23     public MessagePost(String author, String message, long time) {
24         super(author, time);
25         this.message = message;
26     }
27
28     /**
29      * Returns the message of this post.
30      *
31      * @return the message of this post
32      */
33     public String getMessage() {
34         return message;
35     }
36
37     /**
38      * Sets the message of this post.
39      *
40      * @param message the message of this post
41      */
42     public void setMessage(String message) {
43         this.message = message;
44     }
45
46     /**
47      * Returns a string representation of this post.
48      *
49      * @return a string representation of this post
50      */
51     public String toString() {
52         return "MessagePost: " + author + " " + time + " " + message;
53     }
54
55     /**
56      * Displays this post.
57      *
58      * @return true if the post was displayed
59      */
60     public boolean display() {
61         super.display();
62         System.out.println(message);
63         return true;
64     }
65 }

```

Vérifier

Passed all tests! ✓

**Correct**

Note pour cet envoi : 4,00/4,00.

**Question 4**

Correct

Note de 2,00 sur  
2,00

There should be nothing to change in **NewsFeed**. Just checking that everything works together.

Paste all your classes **MessagePost**, **PhotoPost**, **Post**, **NewsFeed** into the Answer.

Réponse:

```
1 package network.v3;
2
3 import java.util.ArrayList;
4
5 /**
6  * This class stores information about a post in a social network news feed. The
7  * main part of the post consists of a (possibly multi-line) text message. Other
8  * data, such as author and time, are also stored.
9  *
10 * @author Michael Kolling and David J. Barnes
11 * @version 0.3
12 */
13 public class MessagePost extends Post {
14     private String message; // an arbitrarily long, multi-line message
15
16     /**
17      * Constructor for objects of class MessagePost.
18      *
19      * @param author the author of the post
20      * @param time the time of the post
21      * @param message the message of the post
22      */
23     public MessagePost(String author, long time, String message) {
24         super(author, time);
25         this.message = message;
26     }
27
28     /**
29      * Returns the message of this post.
30      *
31      * @return the message of this post
32      */
33     public String getMessage() {
34         return message;
35     }
36
37     /**
38      * Returns the message of this post as a list of lines.
39      *
40      * @return the message of this post as a list of lines
41      */
42     public ArrayList<String> getLines() {
43         ArrayList<String> lines = new ArrayList<String>();
44         String[] linesArray = message.split("\n");
45         for (String line : linesArray) {
46             lines.add(line);
47         }
48         return lines;
49     }
50
51     /**
52      * Returns the message of this post as a list of lines.
53      *
54      * @return the message of this post as a list of lines
55      */
56     public String toString() {
57         return "MessagePost: " + author + " " + time + " " + message;
58     }
59 }
```

Vérifier

Passed all tests! ✓

**Correct**

Note pour cet envoi : 2,00/2,00.