



ソフトウェア工学実習 Software Engineering Practice (第07回)

SEP07-008 【実習編】NetBeans-SimpleCalc

拡張(4): 実数型を扱えるようにしよう

慶應義塾大学・理工学部・管理工学科
飯島 正

iiijima@ae.keio.ac.jp

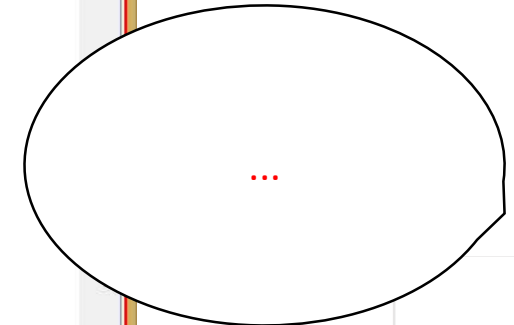
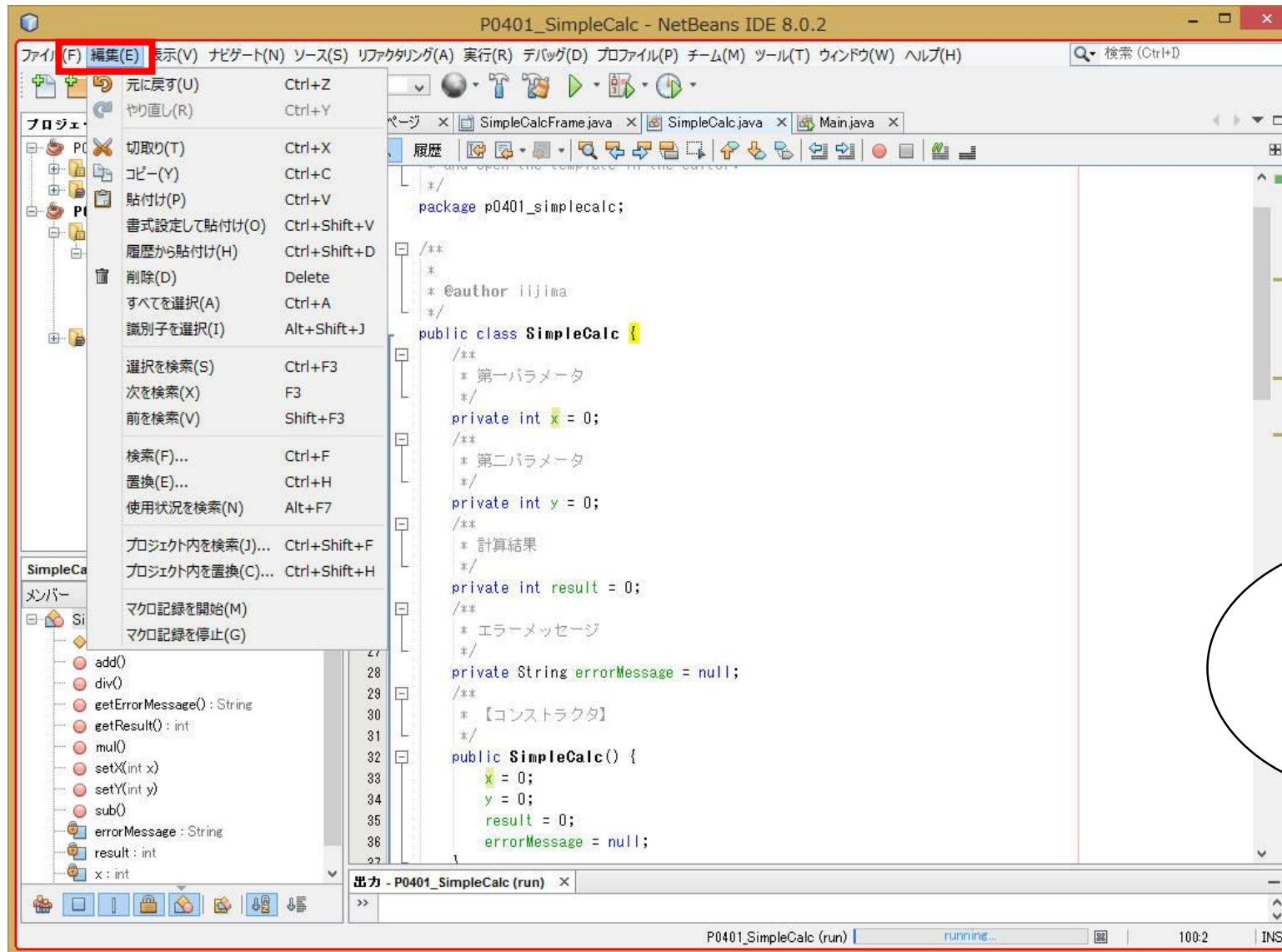
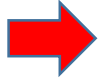
こんにちは。
この授業は、
ソフトウェア
工学実習
です



SimpleCalcクラス: intをfloatに一括変換します

SEP07

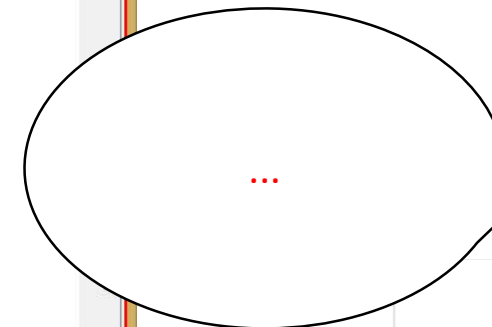
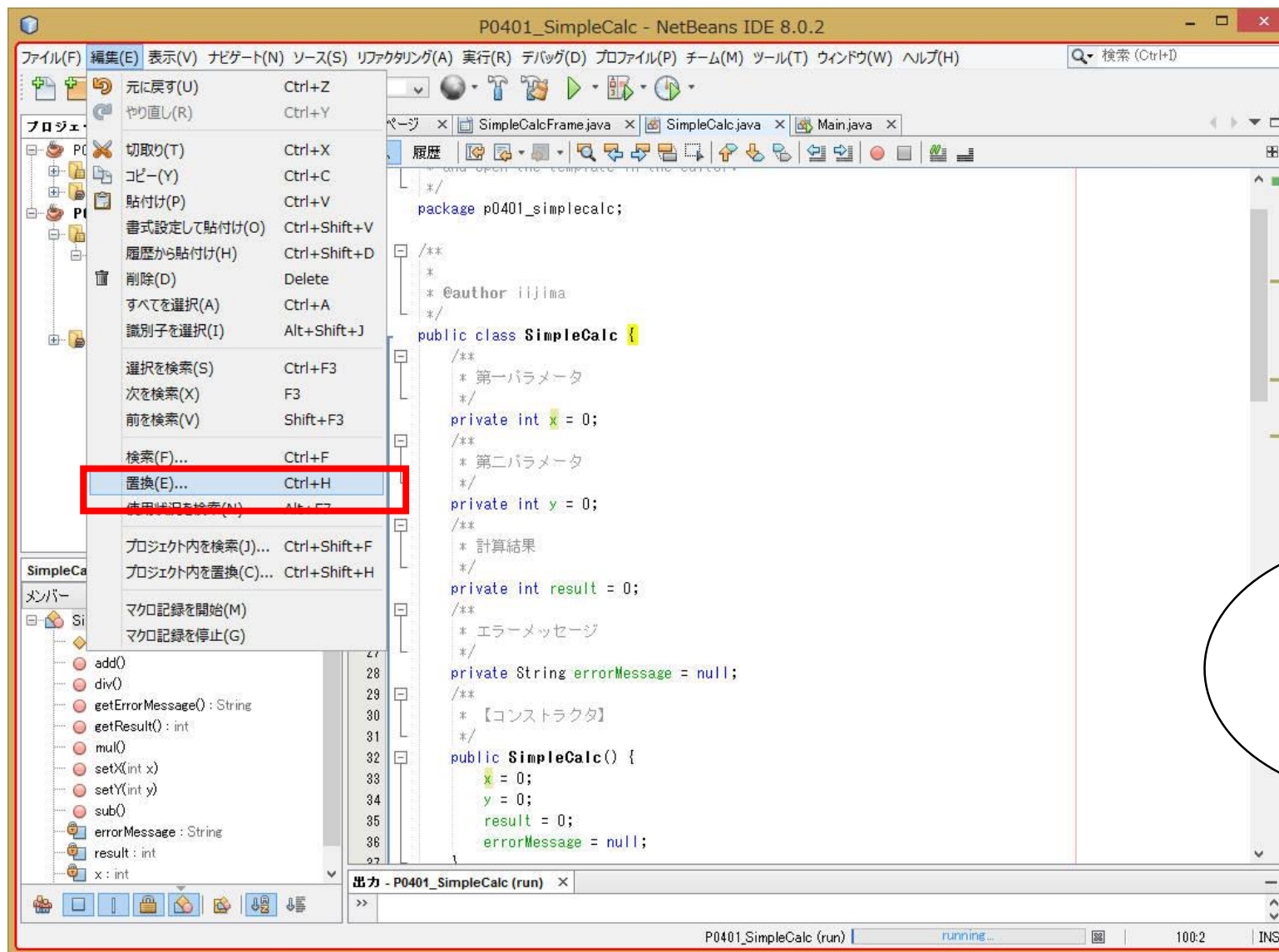
2



SimpleCalcクラス: intをfloatに一括変換します

SEP07

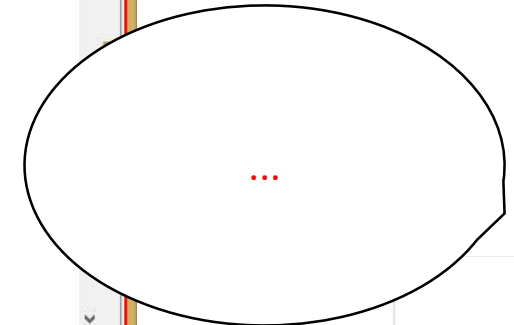
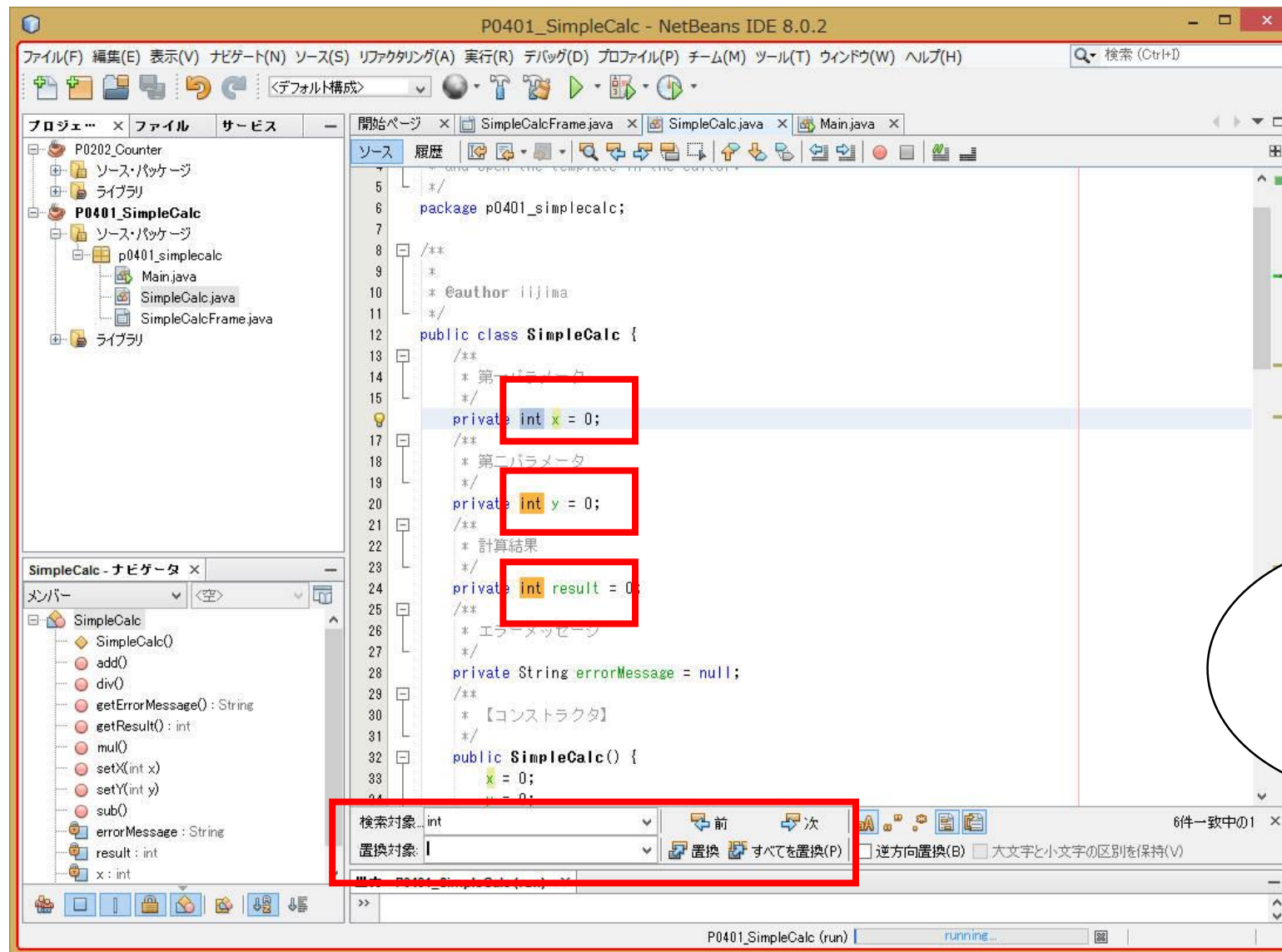
3



SimpleCalcクラス: intをfloatに一括変換します

SEP07

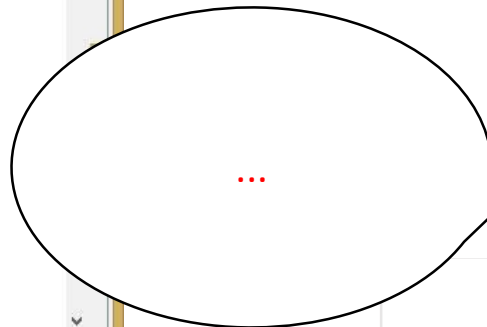
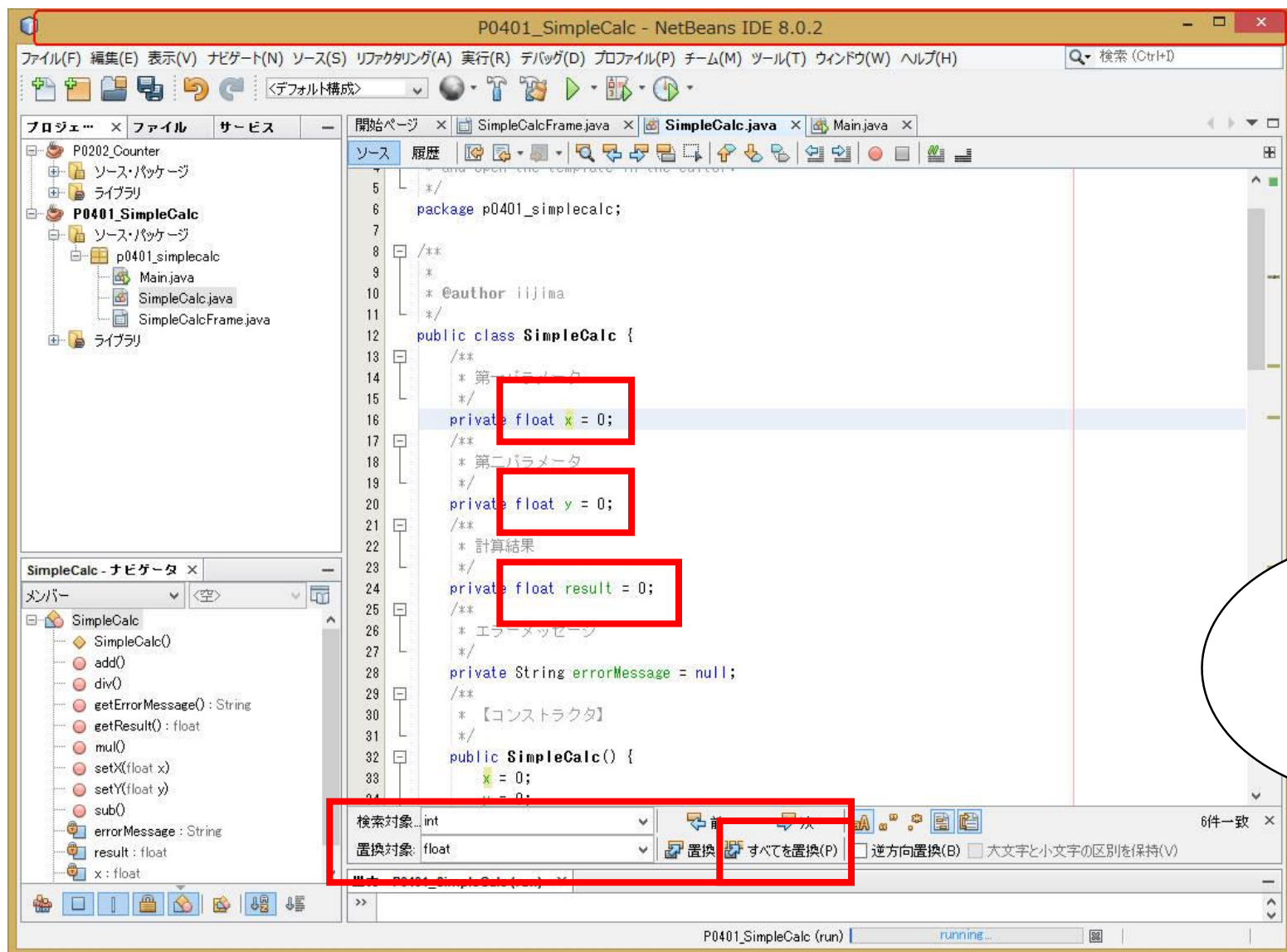
4



SimpleCalcクラス:「すべてを置換」で問題なさそうです

SEP07

5



SimpleCalcFrameクラスの方も，調整しましょう

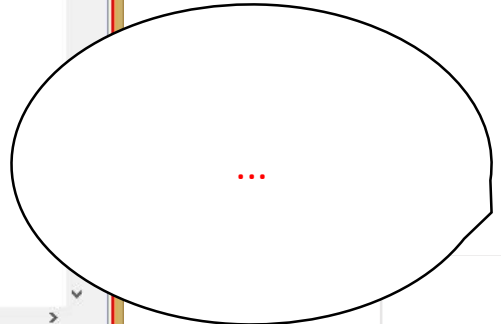
SEP07

6

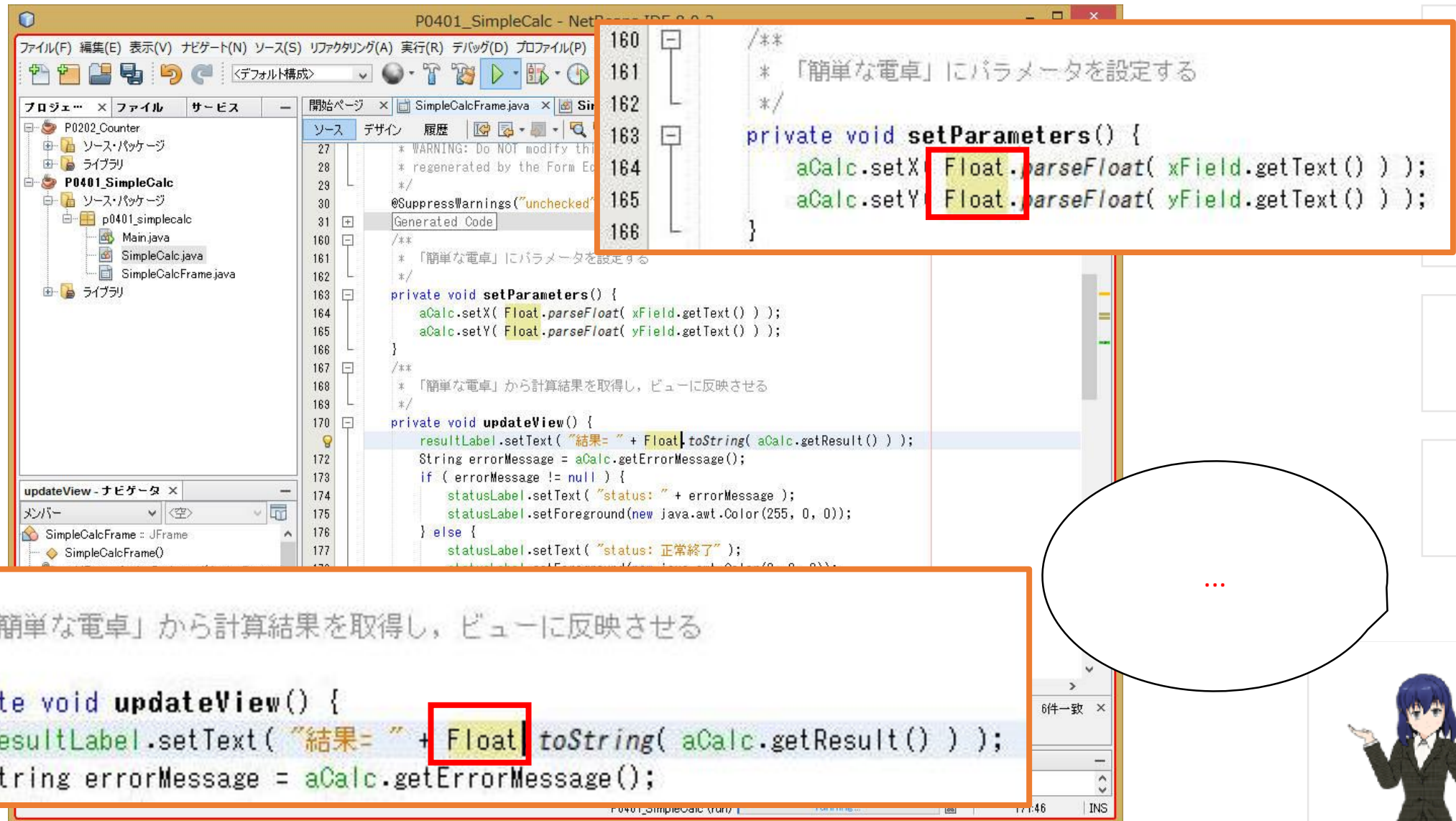
```
160 /**
161  * 「簡単な電卓」にパラメータを設定する
162  */
163 private void setParameters() {
164     aCalc.setX( Integer.parseInt( xField.getText() ) );
165     aCalc.setY( Integer.parseInt( yField.getText() ) );
166 }

167 /**
168  * 「簡単な電卓」から計算結果を取得し，ビューに反映させる
169  */
170 private void updateView() {
171     resultLabel.setText( "結果= " + Integer.toString( aCalc.getResult() ) );
172     String errorMessage = aCalc.getErrorMessage();
173     if ( errorMessage != null ) {
174         statusLabel.setText( "status: " + errorMessage );
175         statusLabel.setForeground( new java.awt.Color( 255, 0, 0 ) );
176     } else {
177         statusLabel.setText( "status: 正常終了" );
178         statusLabel.setForeground( new java.awt.Color( 0, 0, 0 ) );
179     }
180 }

181 /**
182  * 「簡単な電卓」から計算結果を取得し，ビューに反映させる
183  */
184 private void updateView() {
185     resultLabel.setText( "結果= " + Integer.toString( aCalc.getResult() ) );
186     String errorMessage = aCalc.getErrorMessage();
187     if ( errorMessage != null ) {
188         statusLabel.setText( "status: " + errorMessage );
189         statusLabel.setForeground( new java.awt.Color( 255, 0, 0 ) );
190     } else {
191         statusLabel.setText( "status: 正常終了" );
192         statusLabel.setForeground( new java.awt.Color( 0, 0, 0 ) );
193     }
194 }
```



SimpleCalcFrameクラス: getParameterやupdateViewに修正が必要です



```
160  /**
161   * 「簡単な電卓」にパラメータを設定する
162   */
163   private void setParameters() {
164       aCalc.setX( Float.parseFloat( xField.getText() ) );
165       aCalc.setY( Float.parseFloat( yField.getText() ) );
166   }

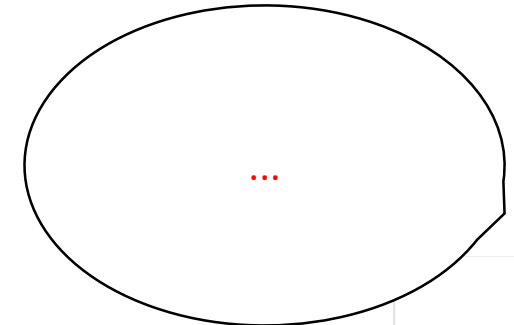
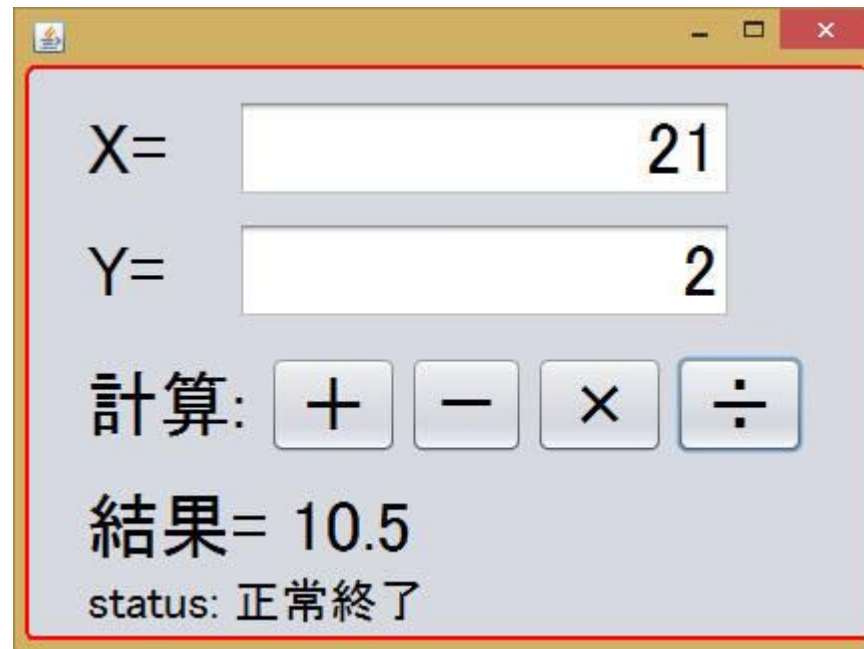
167  /**
168   * 「簡単な電卓」から計算結果を取得し、ビューに反映させる
169   */
170   private void updateView() {
171       resultLabel.setText( "結果=" + Float.toString( aCalc.getResult() ) );
172       String errorMessage = aCalc.getErrorMessage();
173       if ( errorMessage != null ) {
174           statusLabel.setText( "status: " + errorMessage );
175           statusLabel.setForeground( new java.awt.Color( 255, 0, 0 ) );
176       } else {
177           statusLabel.setText( "status: 正常終了" );
178       }
179   }
```



動作確認:結果が小数になる計算はOKです

SEP07

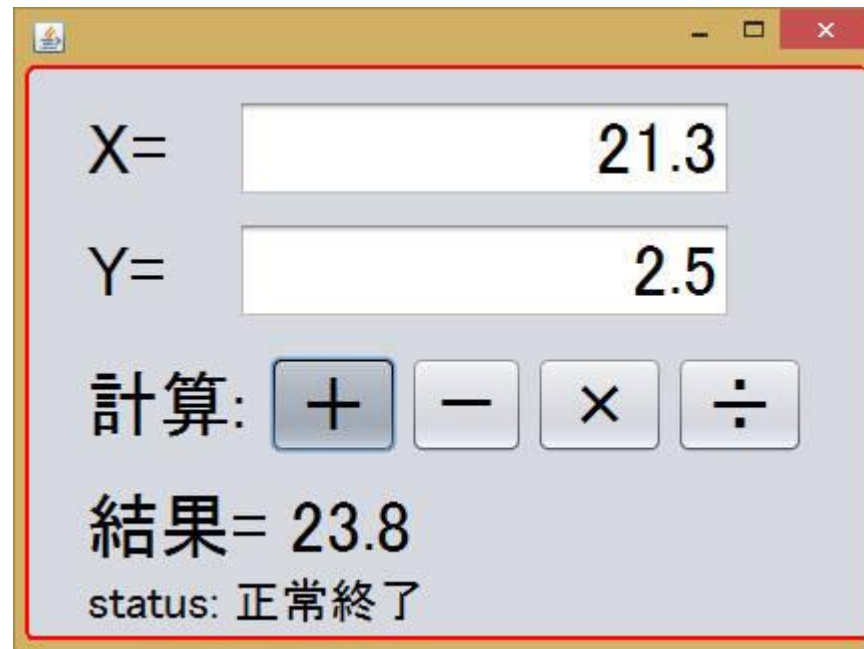
8



動作確認:パラメータに小数を与えてみても大丈夫です

SEP07

9



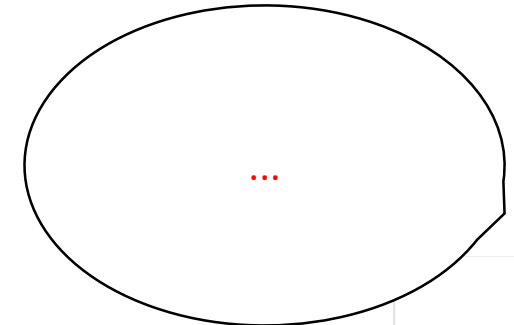
X= 21.3

Y= 2.5

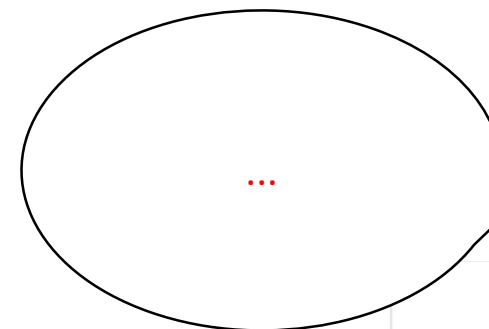
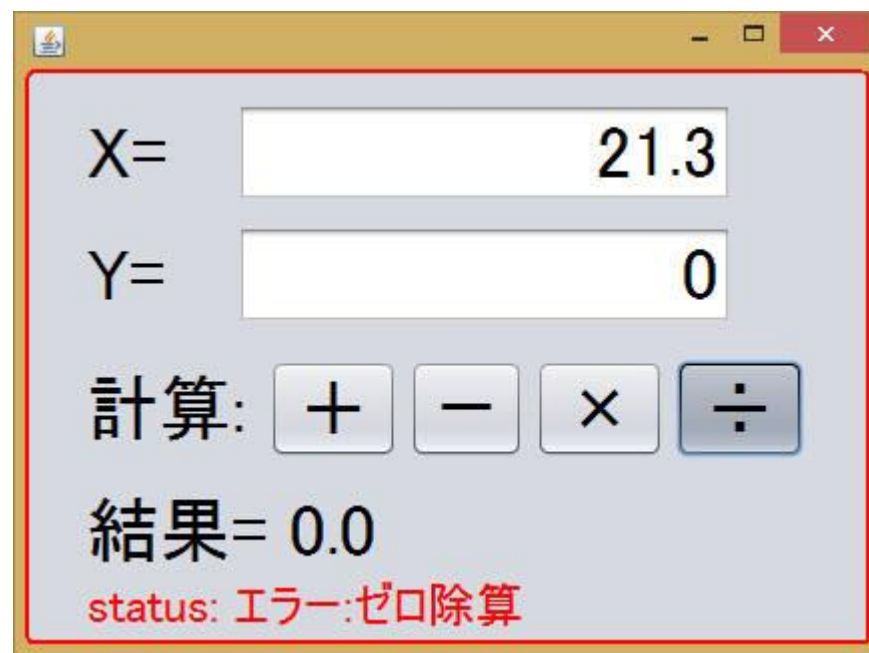
計算: + - × ÷

結果= 23.8

status: 正常終了



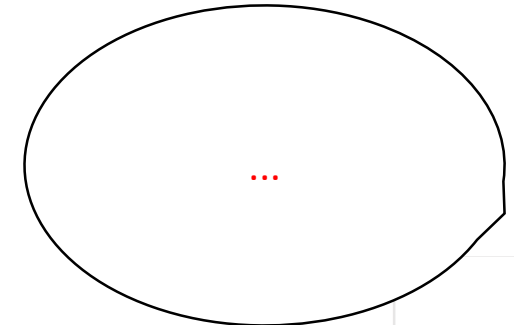
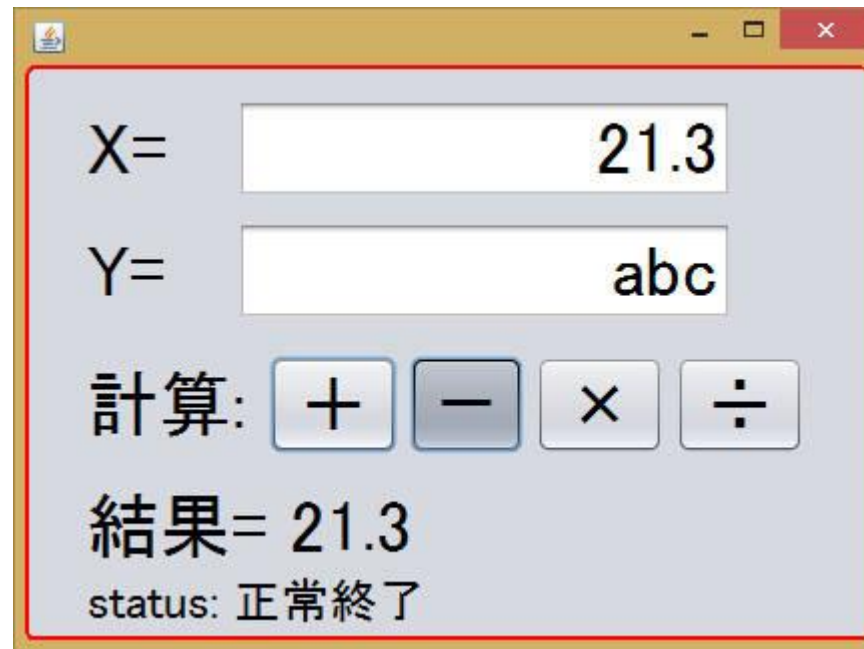
動作確認: ゼロ除算の検出は今まで通りです



次の拡張：数字の代わりに文字を入れてしまうと…

SEP07

11



例外NumberFormatExceptionを出して異常終了します

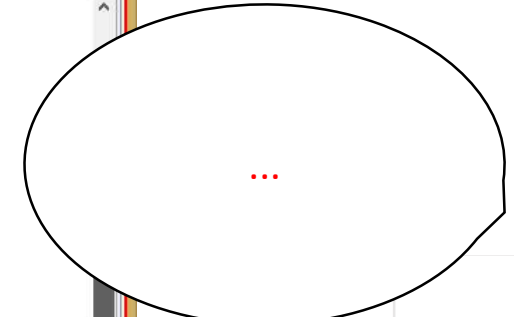
SEP07

12

The screenshot shows the NetBeans IDE interface. The main editor window displays the source code of SimpleCalc.java, which includes a warning from the Form Editor. The '出力' (Output) window shows the following exception stack trace:

```
Exception in thread "AWT-EventQueue-1" java.lang.NumberFormatException: For input string: "abc"
    at sun.misc.FloatingDecimal.readAndFormatDoubleStringWithoutRounding(FloatingDecimal.java:1842)
    at sun.misc.FloatingDecimal.parseFloat(FloatingDecimal.java:122)
    at java.lang.Float.parseFloat(Float.java:451)
    at p0401_simplecalc.SimpleCalcFrame.setParameters(SimpleCalcFrame.java:165)
    at p0401_simplecalc.SimpleCalcFrame.mulButtonActionPerformed(SimpleCalcFrame.java:204)
    at p0401_simplecalc.SimpleCalcFrame.access$200(SimpleCalcFrame.java:12)
    at p0401_simplecalc.SimpleCalcFrame$3.actionPerformed(SimpleCalcFrame.java:88)
    at javax.swing.AbstractButton.fireActionPerformed(AbstractButton.java:2022)
    at javax.swing.AbstractButton$Handler.actionPerformed(AbstractButton.java:2346)
    at javax.swing.DefaultButtonModel.fireActionPerformed(DefaultButtonModel.java:402)
    at javax.swing.DefaultButtonModel.setPressed(DefaultButtonModel.java:259)
    at javax.swing.plaf.basic.BasicButtonListener.mouseReleased(BasicButtonListener.java:252)
    at java.awt.Component.processMouseEvent(Component.java:6525)
    at javax.swing.JComponent.processMouseEvent(JComponent.java:3324)
    at java.awt.Component.processEvent(Component.java:6290)
    at java.awt.Container.processEvent(Container.java:2234)
    at java.awt.Component.dispatchEventImpl(Component.java:4881)
    at java.awt.Container.dispatchEventImpl(Container.java:2292)
    at java.awt.Component.dispatchEvent(Component.java:4703)
    at java.awt.LightweightDispatcher.retargetMouseEvent(Container.java:4898)
```

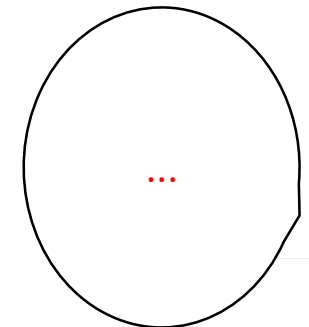
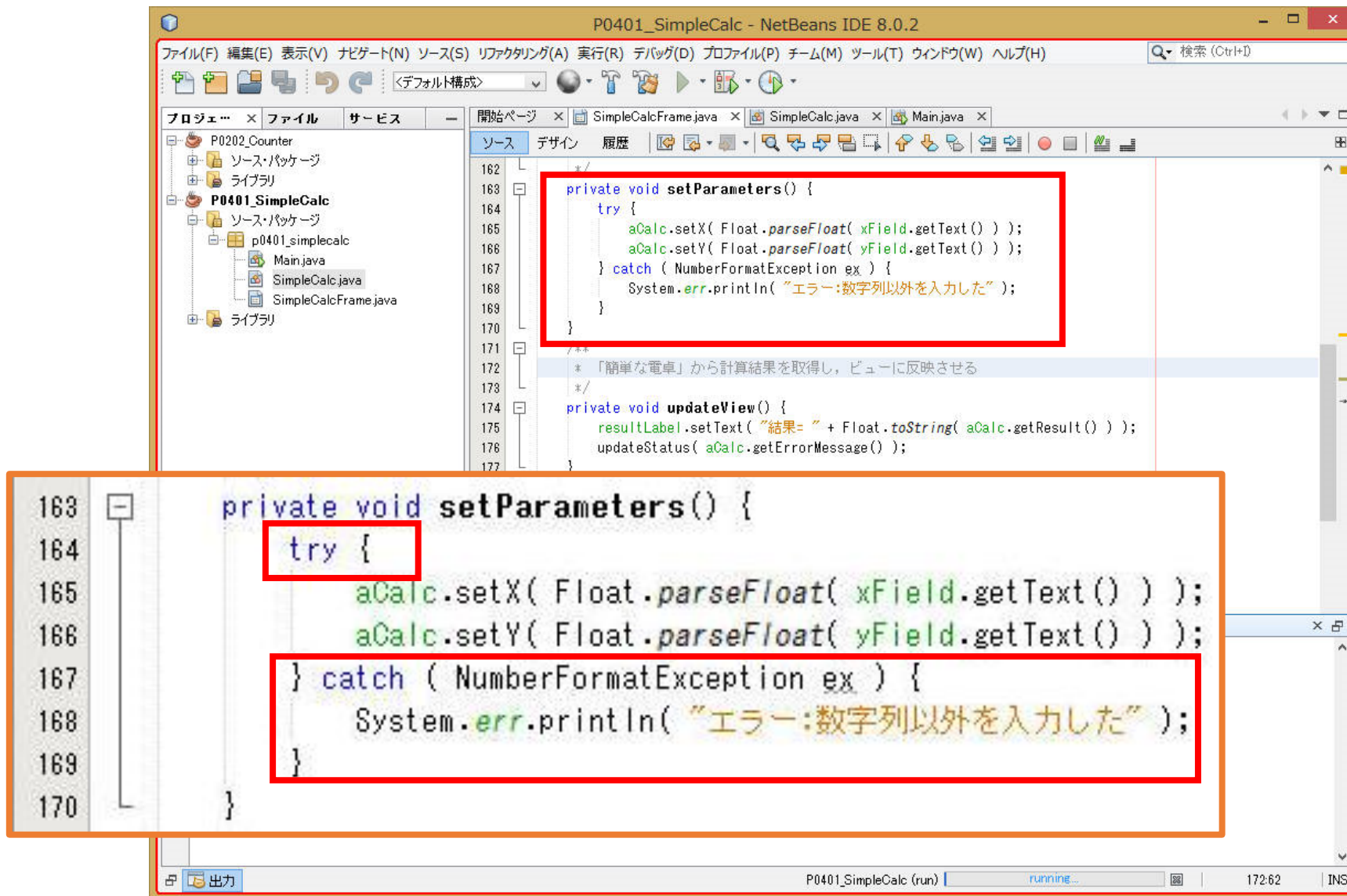
The 'ナビゲータ' (Navigator) window shows the project structure, including the SimpleCalcFrame class and its methods.



Try-catch文で、例外を捕捉 (catch) します

SEP07

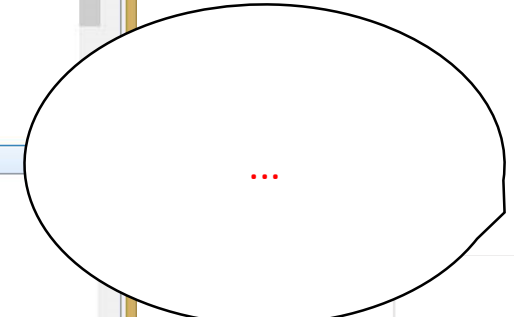
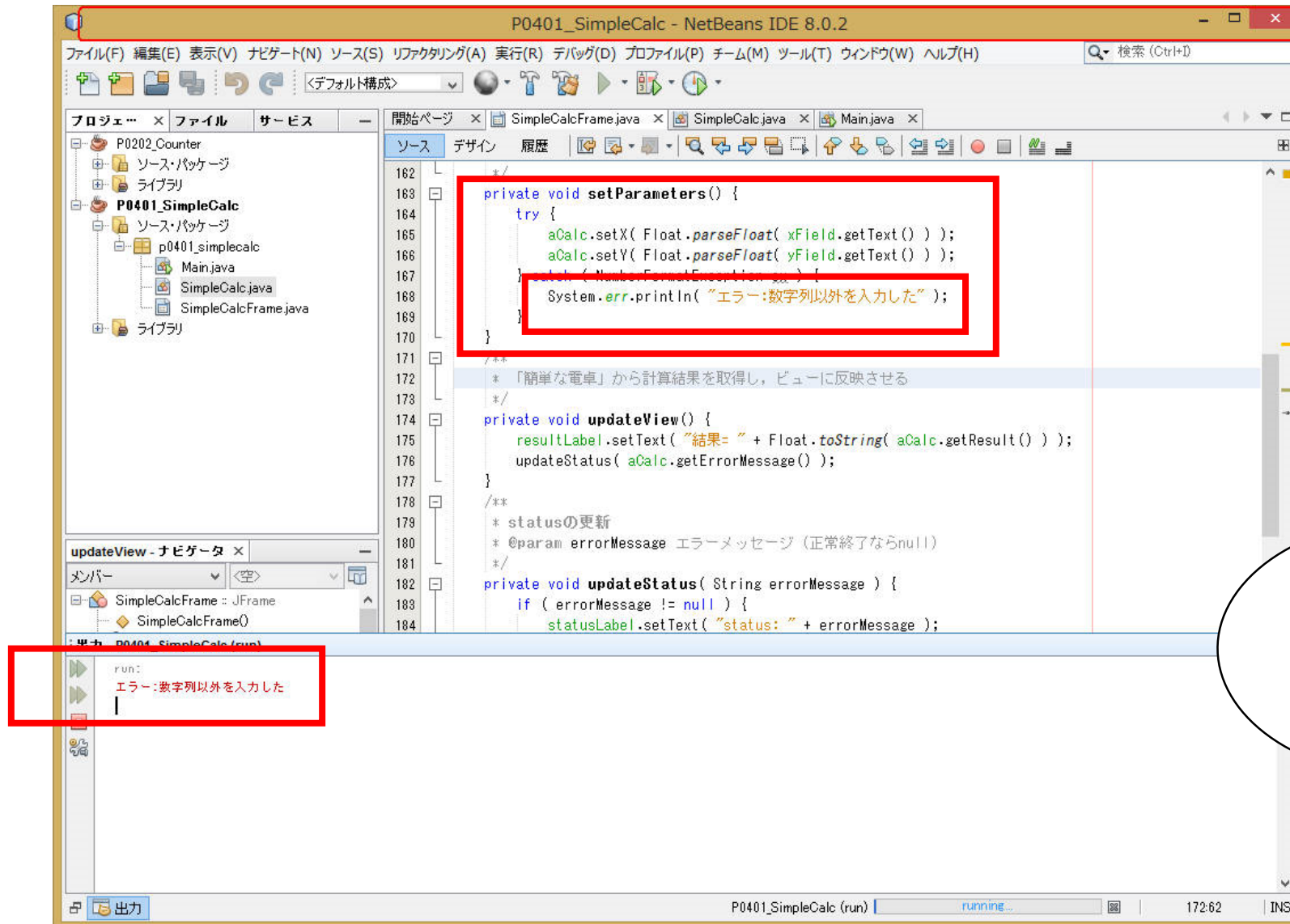
13



コンソール画面には、エラー表示が出せました

SEP07

14

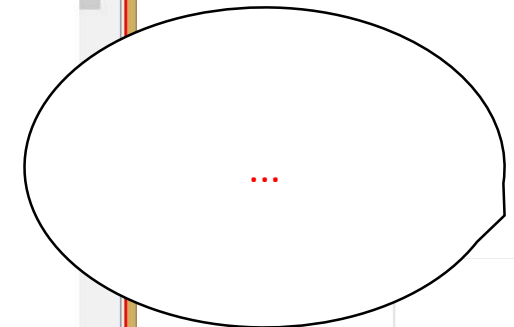


では、フレームにエラーメッセージを出させるには？

SEP07

15

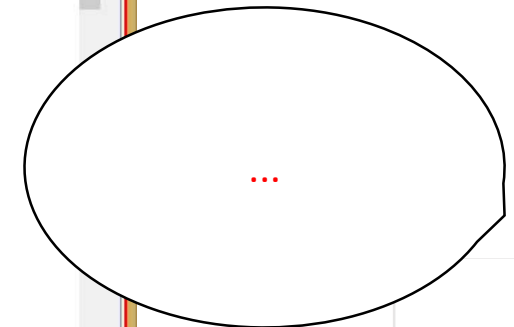
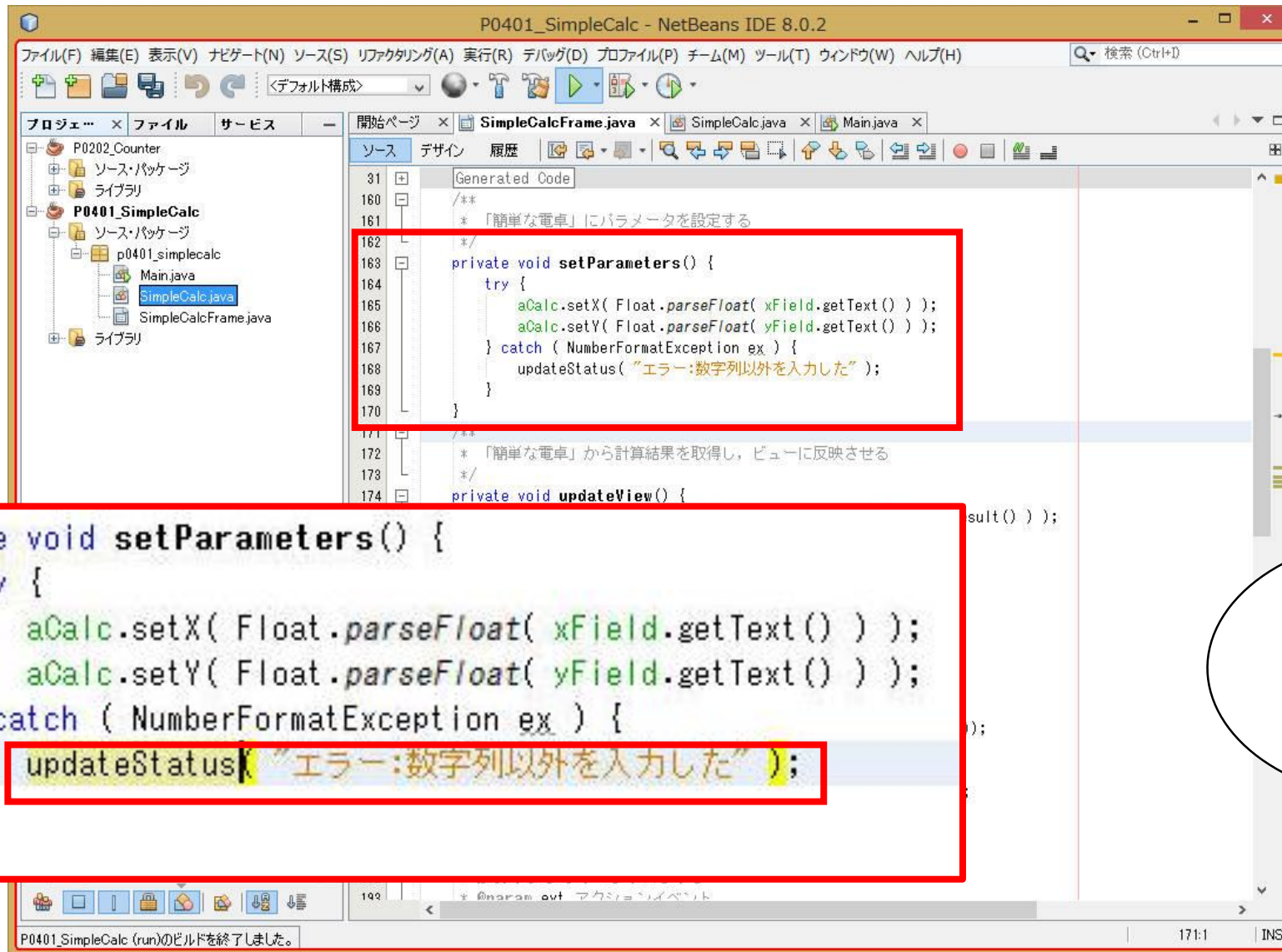
```
160 /**  
161  * 「簡単な電卓」にパラメータを設定する  
162  */  
163 private void setParameters() {  
164     try {  
165         aCalc.setX( Float.parseFloat( xField.getText() ) );  
166         aCalc.setY( Float.parseFloat( yField.getText() ) );  
167     } catch ( NumberFormatException ex ) {  
168         updateStatus( "エラー:数字列以外を入力した" );  
169     }  
170 }  
  
171 /**  
172  * 「簡単な電卓」から計算結果を取得し、ビューに反映させる  
173  */  
174 private void updateView() {  
175     result.setText( aCalc.getResult() );  
176 }  
  
177 /**  
178  * 計算結果を計算機に反映させる  
179  */  
180 private void updateResult() {  
181     aCalc.setResult( result.getText() );  
182 }  
  
183 /**  
184  * 計算機の状態を計算機に反映させる  
185  */  
186 private void updateStatus() {  
187     status.setText( aCalc.getStatus() );  
188 }  
  
189 /**  
190  * 計算機の状態を計算機に反映させる  
191  */  
192 private void updateStatus() {  
193     status.setText( aCalc.getStatus() );  
194 }  
  
195 /**  
196  * 計算機の状態を計算機に反映させる  
197  */  
198 private void updateStatus() {  
199     status.setText( aCalc.getStatus() );  
200 }  
  
201 /**  
202  * 計算機の状態を計算機に反映させる  
203  */  
204 private void updateStatus() {  
205     status.setText( aCalc.getStatus() );  
206 }  
  
207 /**  
208  * 計算機の状態を計算機に反映させる  
209  */  
210 private void updateStatus() {  
211     status.setText( aCalc.getStatus() );  
212 }  
  
213 /**  
214  * 計算機の状態を計算機に反映させる  
215  */  
216 private void updateStatus() {  
217     status.setText( aCalc.getStatus() );  
218 }  
  
219 /**  
220  * 計算機の状態を計算機に反映させる  
221  */  
222 private void updateStatus() {  
223     status.setText( aCalc.getStatus() );  
224 }  
  
225 /**  
226  * 計算機の状態を計算機に反映させる  
227  */  
228 private void updateStatus() {  
229     status.setText( aCalc.getStatus() );  
230 }  
  
231 /**  
232  * 計算機の状態を計算機に反映させる  
233  */  
234 private void updateStatus() {  
235     status.setText( aCalc.getStatus() );  
236 }  
  
237 /**  
238  * 計算機の状態を計算機に反映させる  
239  */  
240 private void updateStatus() {  
241     status.setText( aCalc.getStatus() );  
242 }  
  
243 /**  
244  * 計算機の状態を計算機に反映させる  
245  */  
246 private void updateStatus() {  
247     status.setText( aCalc.getStatus() );  
248 }  
  
249 /**  
250  * 計算機の状態を計算機に反映させる  
251  */  
252 private void updateStatus() {  
253     status.setText( aCalc.getStatus() );  
254 }  
  
255 /**  
256  * 計算機の状態を計算機に反映させる  
257  */  
258 private void updateStatus() {  
259     status.setText( aCalc.getStatus() );  
260 }  
  
261 /**  
262  * 計算機の状態を計算機に反映させる  
263  */  
264 private void updateStatus() {  
265     status.setText( aCalc.getStatus() );  
266 }  
  
267 /**  
268  * 計算機の状態を計算機に反映させる  
269  */  
270 private void updateStatus() {  
271     status.setText( aCalc.getStatus() );  
272 }  
  
273 /**  
274  * 計算機の状態を計算機に反映させる  
275  */  
276 private void updateStatus() {  
277     status.setText( aCalc.getStatus() );  
278 }  
  
279 /**  
280  * 計算機の状態を計算機に反映させる  
281  */  
282 private void updateStatus() {  
283     status.setText( aCalc.getStatus() );  
284 }  
  
285 /**  
286  * 計算機の状態を計算機に反映させる  
287  */  
288 private void updateStatus() {  
289     status.setText( aCalc.getStatus() );  
290 }  
  
291 /**  
292  * 計算機の状態を計算機に反映させる  
293  */  
294 private void updateStatus() {  
295     status.setText( aCalc.getStatus() );  
296 }  
  
297 /**  
298  * 計算機の状態を計算機に反映させる  
299  */  
300 private void updateStatus() {  
301     status.setText( aCalc.getStatus() );  
302 }  
  
303 /**  
304  * 計算機の状態を計算機に反映させる  
305  */  
306 private void updateStatus() {  
307     status.setText( aCalc.getStatus() );  
308 }  
  
309 /**  
310  * 計算機の状態を計算機に反映させる  
311  */  
312 private void updateStatus() {  
313     status.setText( aCalc.getStatus() );  
314 }  
  
315 /**  
316  * 計算機の状態を計算機に反映させる  
317  */  
318 private void updateStatus() {  
319     status.setText( aCalc.getStatus() );  
320 }  
  
321 /**  
322  * 計算機の状態を計算機に反映させる  
323  */  
324 private void updateStatus() {  
325     status.setText( aCalc.getStatus() );  
326 }  
  
327 /**  
328  * 計算機の状態を計算機に反映させる  
329  */  
330 private void updateStatus() {  
331     status.setText( aCalc.getStatus() );  
332 }  
  
333 /**  
334  * 計算機の状態を計算機に反映させる  
335  */  
336 private void updateStatus() {  
337     status.setText( aCalc.getStatus() );  
338 }  
  
339 /**  
340  * 計算機の状態を計算機に反映させる  
341  */  
342 private void updateStatus() {  
343     status.setText( aCalc.getStatus() );  
344 }  
  
345 /**  
346  * 計算機の状態を計算機に反映させる  
347  */  
348 private void updateStatus() {  
349     status.setText( aCalc.getStatus() );  
350 }  
  
351 /**  
352  * 計算機の状態を計算機に反映させる  
353  */  
354 private void updateStatus() {  
355     status.setText( aCalc.getStatus() );  
356 }  
  
357 /**  
358  * 計算機の状態を計算機に反映させる  
359  */  
360 private void updateStatus() {  
361     status.setText( aCalc.getStatus() );  
362 }  
  
363 /**  
364  * 計算機の状態を計算機に反映させる  
365  */  
366 private void updateStatus() {  
367     status.setText( aCalc.getStatus() );  
368 }  
  
369 /**  
370  * 計算機の状態を計算機に反映させる  
371  */  
372 private void updateStatus() {  
373     status.setText( aCalc.getStatus() );  
374 }  
  
375 /**  
376  * 計算機の状態を計算機に反映させる  
377  */  
378 private void updateStatus() {  
379     status.setText( aCalc.getStatus() );  
380 }  
  
381 /**  
382  * 計算機の状態を計算機に反映させる  
383  */  
384 private void updateStatus() {  
385     status.setText( aCalc.getStatus() );  
386 }  
  
387 /**  
388  * 計算機の状態を計算機に反映させる  
389  */  
390 private void updateStatus() {  
391     status.setText( aCalc.getStatus() );  
392 }  
  
393 /**  
394  * 計算機の状態を計算機に反映させる  
395  */  
396 private void updateStatus() {  
397     status.setText( aCalc.getStatus() );  
398 }  
  
399 /**  
400  * 計算機の状態を計算機に反映させる  
401  */  
402 private void updateStatus() {  
403     status.setText( aCalc.getStatus() );  
404 }  
  
405 /**  
406  * 計算機の状態を計算機に反映させる  
407  */  
408 private void updateStatus() {  
409     status.setText( aCalc.getStatus() );  
410 }  
  
411 /**  
412  * 計算機の状態を計算機に反映させる  
413  */  
414 private void updateStatus() {  
415     status.setText( aCalc.getStatus() );  
416 }  
  
417 /**  
418  * 計算機の状態を計算機に反映させる  
419  */  
420 private void updateStatus() {  
421     status.setText( aCalc.getStatus() );  
422 }  
  
423 /**  
424  * 計算機の状態を計算機に反映させる  
425  */  
426 private void updateStatus() {  
427     status.setText( aCalc.getStatus() );  
428 }  
  
429 /**  
430  * 計算機の状態を計算機に反映させる  
431  */  
432 private void updateStatus() {  
433     status.setText( aCalc.getStatus() );  
434 }  
  
435 /**  
436  * 計算機の状態を計算機に反映させる  
437  */  
438 private void updateStatus() {  
439     status.setText( aCalc.getStatus() );  
440 }  
  
441 /**  
442  * 計算機の状態を計算機に反映させる  
443  */  
444 private void updateStatus() {  
445     status.setText( aCalc.getStatus() );  
446 }  
  
447 /**  
448  * 計算機の状態を計算機に反映させる  
449  */  
450 private void updateStatus() {  
451     status.setText( aCalc.getStatus() );  
452 }  
  
453 /**  
454  * 計算機の状態を計算機に反映させる  
455  */  
456 private void updateStatus() {  
457     status.setText( aCalc.getStatus() );  
458 }  
  
459 /**  
460  * 計算機の状態を計算機に反映させる  
461  */  
462 private void updateStatus() {  
463     status.setText( aCalc.getStatus() );  
464 }  
  
465 /**  
466  * 計算機の状態を計算機に反映させる  
467  */  
468 private void updateStatus() {  
469     status.setText( aCalc.getStatus() );  
470 }  
  
471 /**  
472  * 計算機の状態を計算機に反映させる  
473  */  
474 private void updateStatus() {  
475     status.setText( aCalc.getStatus() );  
476 }  
  
477 /**  
478  * 計算機の状態を計算機に反映させる  
479  */  
480 private void updateStatus() {  
481     status.setText( aCalc.getStatus() );  
482 }  
  
483 /**  
484  * 計算機の状態を計算機に反映させる  
485  */  
486 private void updateStatus() {  
487     status.setText( aCalc.getStatus() );  
488 }  
  
489 /**  
490  * 計算機の状態を計算機に反映させる  
491  */  
492 private void updateStatus() {  
493     status.setText( aCalc.getStatus() );  
494 }  
  
495 /**  
496  * 計算機の状態を計算機に反映させる  
497  */  
498 private void updateStatus() {  
499     status.setText( aCalc.getStatus() );  
500 }  
  
501 /**  
502  * 計算機の状態を計算機に反映させる  
503  */  
504 private void updateStatus() {  
505     status.setText( aCalc.getStatus() );  
506 }  
  
507 /**  
508  * 計算機の状態を計算機に反映させる  
509  */  
510 private void updateStatus() {  
511     status.setText( aCalc.getStatus() );  
512 }  
  
513 /**  
514  * 計算機の状態を計算機に反映させる  
515  */  
516 private void updateStatus() {  
517     status.setText( aCalc.getStatus() );  
518 }  
  
519 /**  
520  * 計算機の状態を計算機に反映させる  
521  */  
522 private void updateStatus() {  
523     status.setText( aCalc.getStatus() );  
524 }  
  
525 /**  
526  * 計算機の状態を計算機に反映させる  
527  */  
528 private void updateStatus() {  
529     status.setText( aCalc.getStatus() );  
530 }  
  
531 /**  
532  * 計算機の状態を計算機に反映させる  
533  */  
534 private void updateStatus() {  
535     status.setText( aCalc.getStatus() );  
536 }  
  
537 /**  
538  * 計算機の状態を計算機に反映させる  
539  */  
540 private void updateStatus() {  
541     status.setText( aCalc.getStatus() );  
542 }  
  
543 /**  
544  * 計算機の状態を計算機に反映させる  
545  */  
546 private void updateStatus() {  
547     status.setText( aCalc.getStatus() );  
548 }  
  
549 /**  
550  * 計算機の状態を計算機に反映させる  
551  */  
552 private void updateStatus() {  
553     status.setText( aCalc.getStatus() );  
554 }  
  
555 /**  
556  * 計算機の状態を計算機に反映させる  
557  */  
558 private void updateStatus() {  
559     status.setText( aCalc.getStatus() );  
560 }  
  
561 /**  
562  * 計算機の状態を計算機に反映させる  
563  */  
564 private void updateStatus() {  
565     status.setText( aCalc.getStatus() );  
566 }  
  
567 /**  
568  * 計算機の状態を計算機に反映させる  
569  */  
570 private void updateStatus() {  
571     status.setText( aCalc.getStatus() );  
572 }  
  
573 /**  
574  * 計算機の状態を計算機に反映させる  
575  */  
576 private void updateStatus() {  
577     status.setText( aCalc.getStatus() );  
578 }  
  
579 /**  
580  * 計算機の状態を計算機に反映させる  
581  */  
582 private void updateStatus() {  
583     status.setText( aCalc.getStatus() );  
584 }  
  
585 /**  
586  * 計算機の状態を計算機に反映させる  
587  */  
588 private void updateStatus() {  
589     status.setText( aCalc.getStatus() );  
590 }  
  
591 /**  
592  * 計算機の状態を計算機に反映させる  
593  */  
594 private void updateStatus() {  
595     status.setText( aCalc.getStatus() );  
596 }  
  
597 /**  
598  * 計算機の状態を計算機に反映させる  
599  */  
600 private void updateStatus() {  
601     status.setText( aCalc.getStatus() );  
602 }  
  
603 /**  
604  * 計算機の状態を計算機に反映させる  
605  */  
606 private void updateStatus() {  
607     status.setText( aCalc.getStatus() );  
608 }  
  
609 /**  
610  * 計算機の状態を計算機に反映させる  
611  */  
612 private void updateStatus() {  
613     status.setText( aCalc.getStatus() );  
614 }  
  
615 /**  
616  * 計算機の状態を計算機に反映させる  
617  */  
618 private void updateStatus() {  
619     status.setText( aCalc.getStatus() );  
620 }  
  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```



実は, try-catch文の位置に問題があります

SEP07

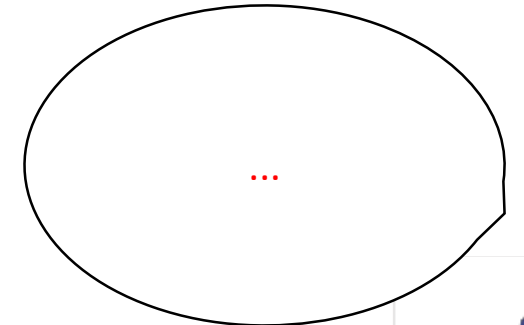
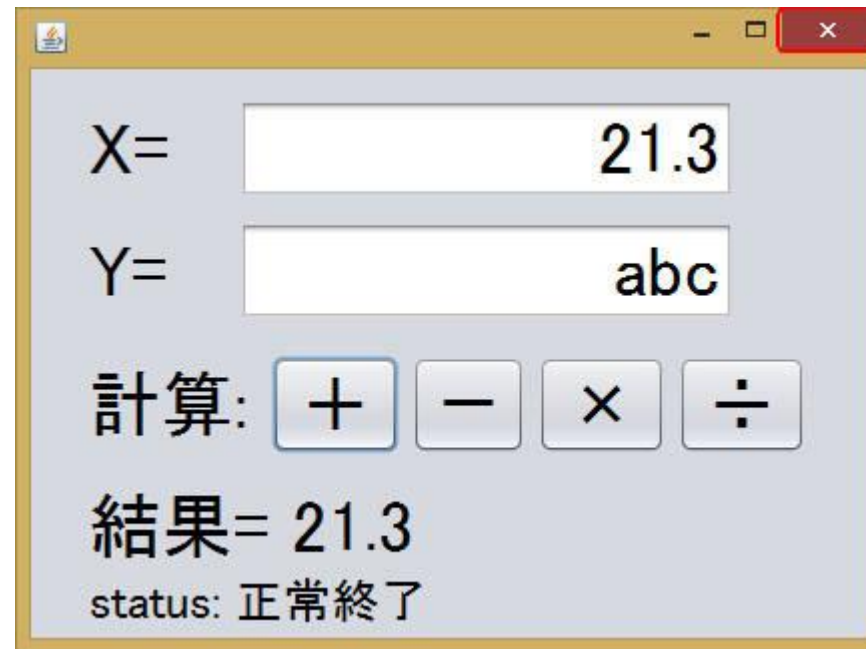
16



うまくエラーメッセージが出てくれません

SEP07

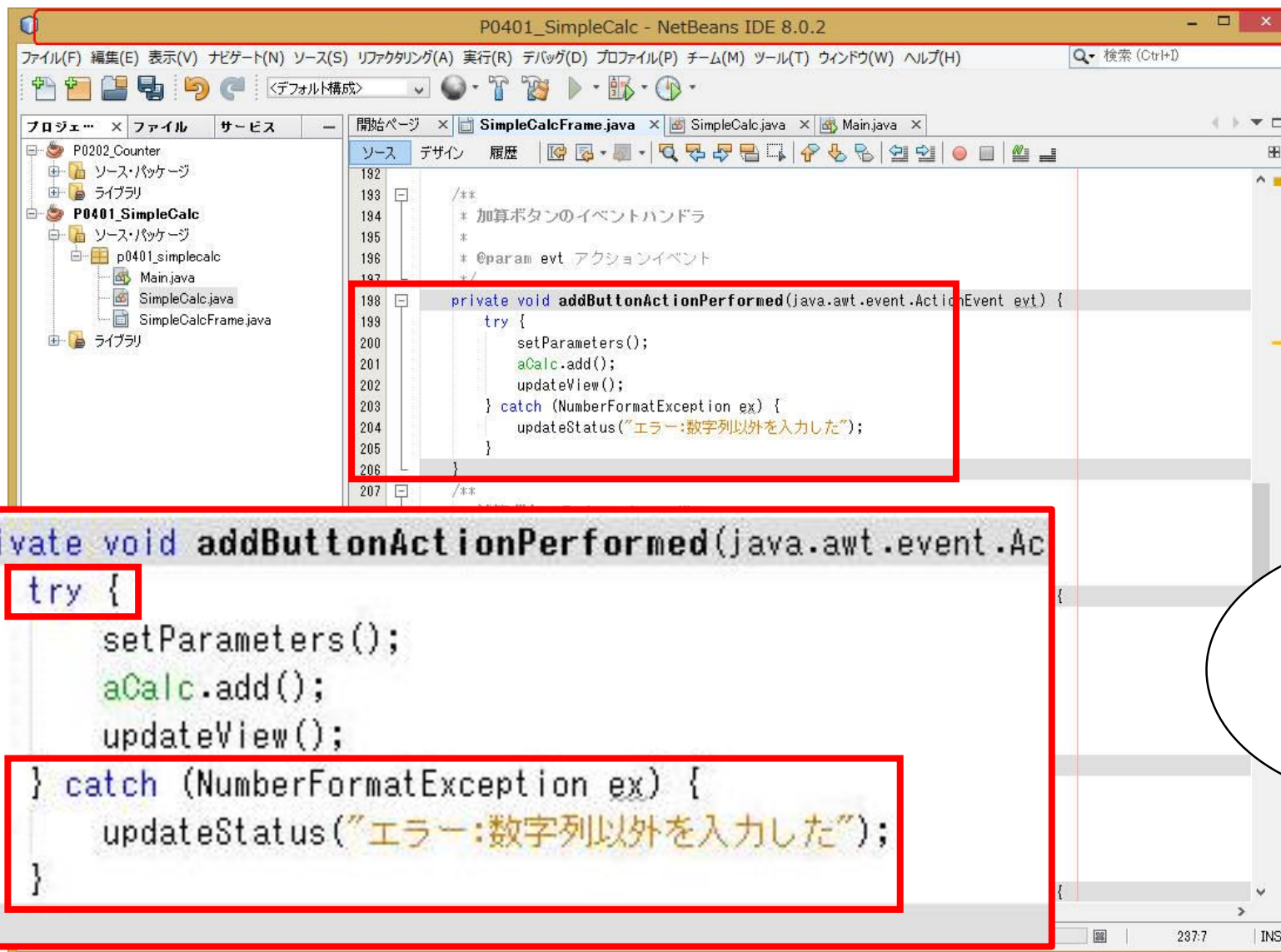
17



Tryブロックには、個々の計算も含まれるように、イベントハンドラの中へ移動させます

SEP07

18



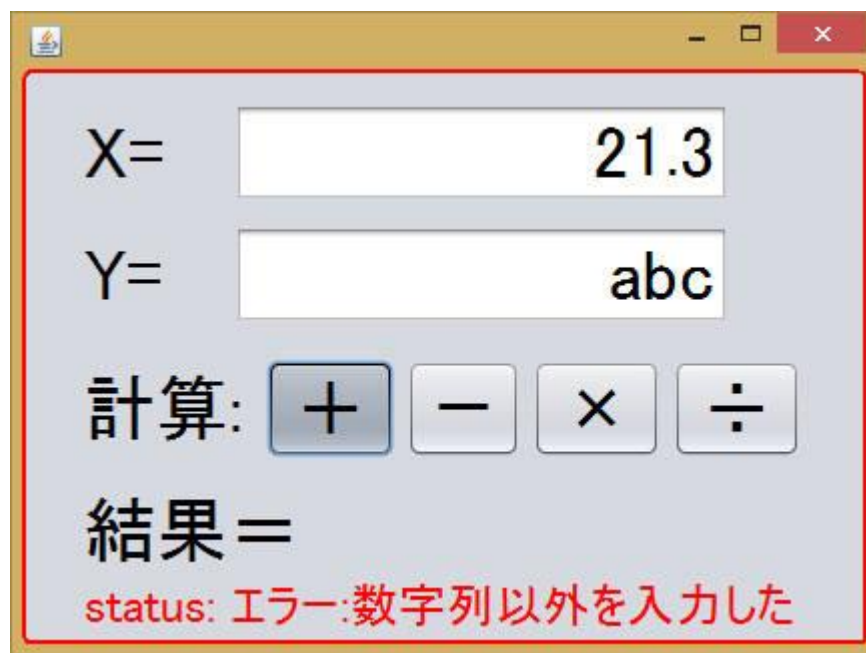
```
192  /**
193  * 加算ボタンのイベントハンドラ
194  *
195  * @param evt アクションイベント
196  */
197
198  private void addButtonActionPerformed(java.awt.event.ActionEvent evt) {
199      try {
200          setParameters();
201          aCalc.add();
202          updateView();
203      } catch (NumberFormatException ex) {
204          updateStatus("エラー:数字列以外を入力した");
205      }
206  }
```



うまくいきました

SEP07

19



X= 21.3

Y= abc

計算: + - × ÷

結果=

status: エラー: 数字列以外を入力した

