

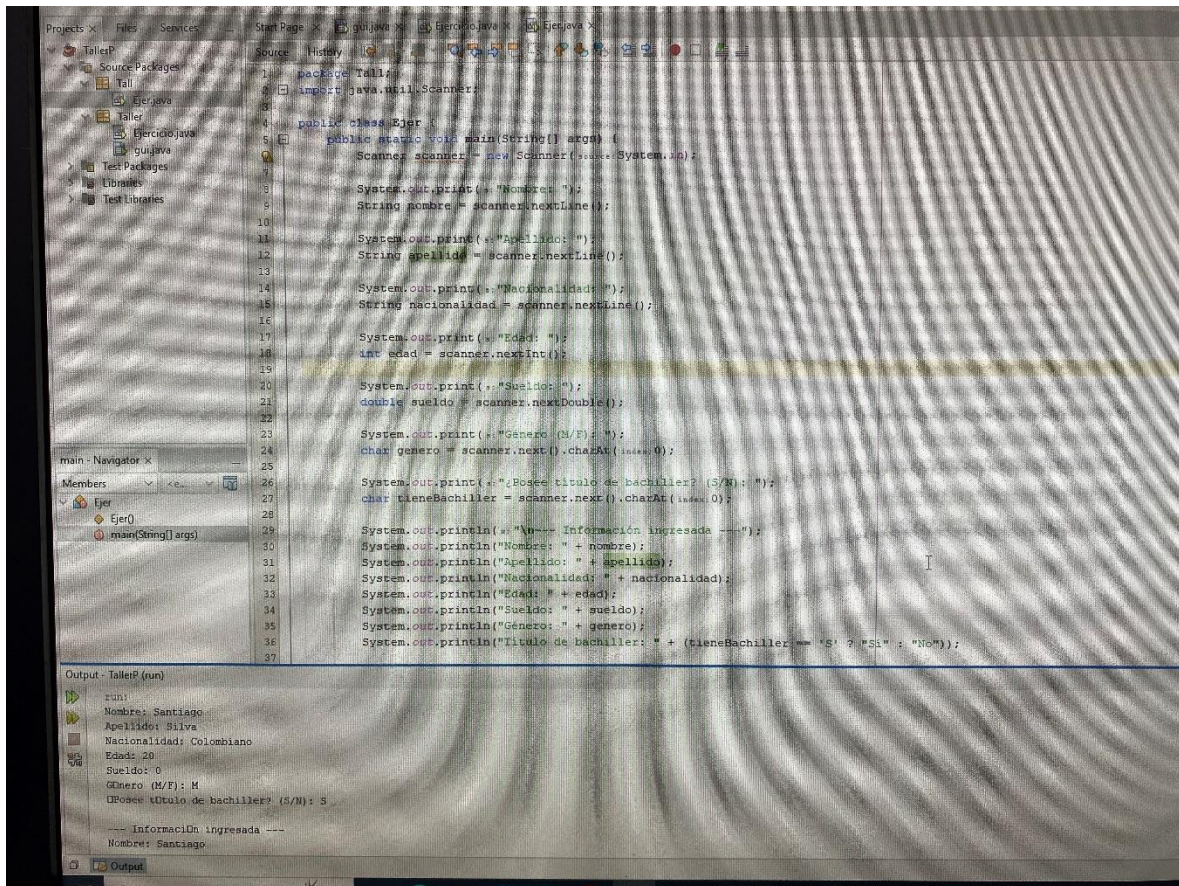
# **TALLER PROGRAMACION**

**SANTIAGO SILVA  
BRAYAN MANRIQUE  
JOSUE BERNAL**

**DOCENTE  
MIGUEL ANGEL MONROY DIAZ**

**UNIVERSIDAD MILITAR NUEVA GRANADA  
INGENIERÍA MECATRÓNICA  
BOGOTÁ D.C  
JUNIO 2023**

## 1. Código simple



The screenshot shows an IDE with a Java project named 'TallerP'. The source code is in a file named 'Ejer.java'. The code uses a Scanner to collect user input for various fields and then prints the collected information. The output window shows the results of the program execution.

```
1 package TallerP;
2 import java.util.Scanner;
3
4 public class Ejer {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7
8         System.out.print("Nombre: ");
9         String nombre = scanner.nextLine();
10
11         System.out.print("Apellido: ");
12         String apellido = scanner.nextLine();
13
14         System.out.print("Nacionalidad: ");
15         String nacionalidad = scanner.nextLine();
16
17         System.out.print("Edad: ");
18         int edad = scanner.nextInt();
19
20         System.out.print("Sueldo: ");
21         double sueldo = scanner.nextDouble();
22
23         System.out.print("Género (M/F): ");
24         char genero = scanner.next().charAt(0);
25
26         System.out.print("¿Posee título de bachiller? (S/N): ");
27         char tieneBachiller = scanner.next().charAt(0);
28
29         System.out.println("\n--- Información ingresada ---");
30         System.out.println("Nombre: " + nombre);
31         System.out.println("Apellido: " + apellido);
32         System.out.println("Nacionalidad: " + nacionalidad);
33         System.out.println("Edad: " + edad);
34         System.out.println("Sueldo: " + sueldo);
35         System.out.println("Género: " + genero);
36         System.out.println("Título de bachiller: " + (tieneBachiller == 'S' ? "SI" : "No"));
37     }
38 }
```

main - Navigator x

Members

- Ejer()
- main(String[] args)

Output - TallerP (run)

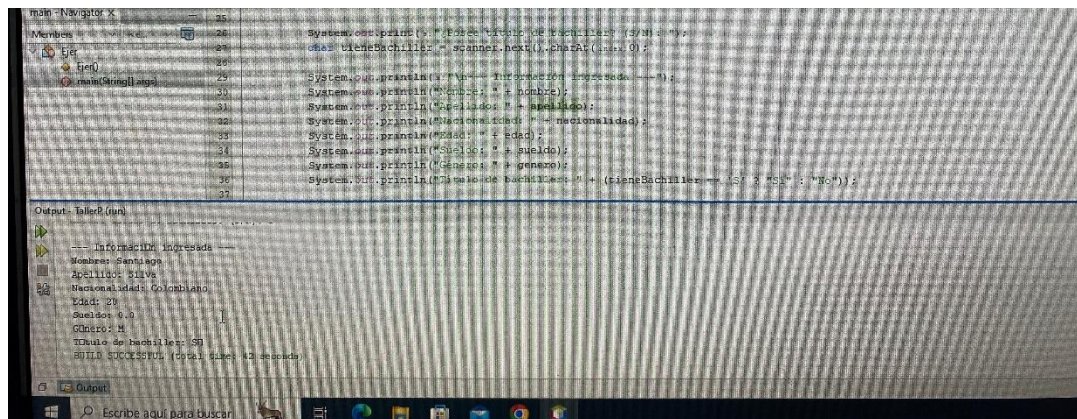
run:

Nombre: Santiago  
Apellido: Silva  
Nacionalidad: Colombiano  
Edad: 20  
Sueldo: 0  
Género (M/F): M  
¿Posee título de bachiller? (S/N): S

--- Información ingresada ---  
Nombre: Santiago



## 2. Código simple

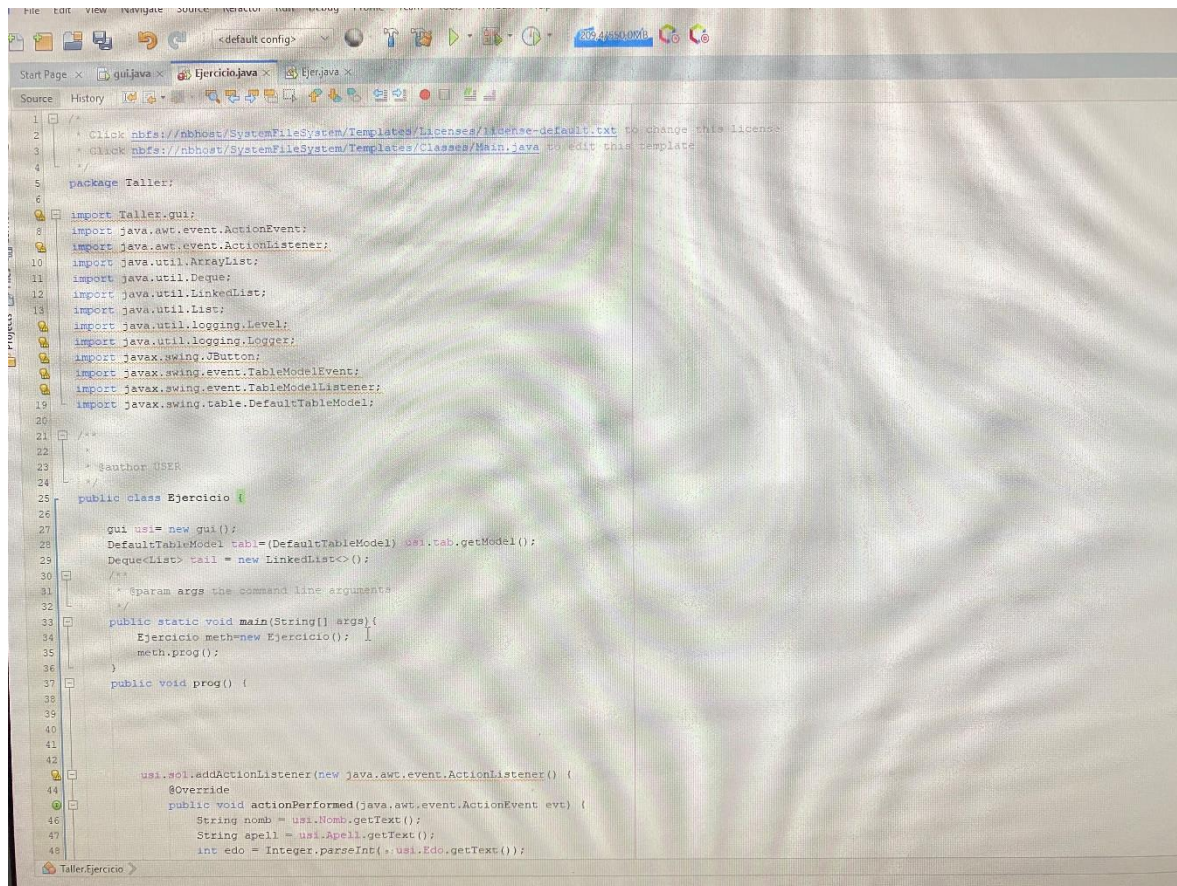


```
1 package Taller;
2
3 import java.util.Scanner;
4
5 public class Taller {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         System.out.println("Ingrese el nombre del trabajador:");
9         String nombre = sc.nextLine();
10        System.out.println("Ingrese el apellido del trabajador:");
11        String apellido = sc.nextLine();
12        System.out.println("Ingrese la nacionalidad del trabajador:");
13        String nacionalidad = sc.nextLine();
14        System.out.println("Ingrese la edad del trabajador:");
15        int edad = sc.nextInt();
16        System.out.println("Ingrese el sueldo del trabajador:");
17        double sueldo = sc.nextDouble();
18        System.out.println("Ingrese el genero del trabajador:");
19        String genero = sc.next();
20        System.out.println("Ingrese el titulo de bachiller del trabajador:");
21        boolean tieneBachiller = sc.next().equalsIgnoreCase("si");
22        System.out.println("El trabajador es: " + nombre + " " + apellido + " " + nacionalidad + " " + edad + " años " + sueldo + " de sueldo " + genero + " " + (tieneBachiller ? "Si" : "No"));
23    }
24 }
```

Output: Taller (Run)

```
--- Información ingresada ---
Nombre: Santiago
Apellido: Silva
Nacionalidad: Colombiano
Edad: 21
Sueldo: 0.0
Genero: M
Titulo de bachiller: Si
BACHILLER (Total: Si) 42 años
```

## 3. Código con interfaz



```
1 package Taller;
2
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import java.util.ArrayList;
6 import java.util.Deque;
7 import java.util.LinkedList;
8 import java.util.List;
9 import java.util.logging.Level;
10 import java.util.logging.Logger;
11 import javax.swing.JButton;
12 import javax.swing.event.TableModelEvent;
13 import javax.swing.event.TableModelListener;
14 import javax.swing.table.DefaultTableModel;
15
16 /**
17  * @author USER
18  */
19 public class Ejercicio {
20     private static final Logger logger = Logger.getLogger(Ejercicio.class);
21     private static final DefaultTableModel tab = new DefaultTableModel();
22     private static final Deque<List> tail = new LinkedList<>();
23
24     /**
25      * Param args the command line arguments
26      */
27     public static void main(String[] args) {
28         Ejercicio meth = new Ejercicio();
29         meth.prog();
30     }
31
32     public void prog() {
33         // Create a new GUI
34         GUI gui = new GUI();
35         DefaultTableModel tab = (DefaultTableModel) gui.tab.getModel();
36         Deque<List> tail = new LinkedList<>();
37
38         // Add an ActionListener to the GUI
39         gui.wol.addActionListener(new java.awt.event.ActionListener() {
40             @Override
41             public void actionPerformed(java.awt.event.ActionEvent evt) {
42                 String nomb = gui.Nomb.getText();
43                 String apell = gui.Apell.getText();
44                 int edo = Integer.parseInt(gui.Edo.getText());
45             }
46         });
47     }
48 }
```

## 4. Código con interfaz



```
75     usi.comd.setSelectedIndex(selectedIndex);
76 }
77
78 }
79
80 // TODO code application logic here
81
82
83
84
85
86
87
88 public void updat() {
89     tabl.setRowCount(0);
90     tabl.setRowCount(tabl.getRowCount());
91     int i=0;
92     for (List ob: tabl) {
93         tabl.setValueAt(ob.get(index1), row1, column0);
94         tabl.setValueAt(ob.get(index0), row1, column1);
95         i++;
96     }
97 }
98
99
100
101 public List persona(String nomb, String apell, String nac, String titu, String gen, int edad) {
102     List<Object> person = new ArrayList<>();
103     person.add(nomb);
104     person.add(apell);
105     person.add(edad);
106     person.add(nac);
107     person.add(titu);
108     person.add(gen);
109     return person;
110 }
111
112 }
113
114
```

## 5. Código con interfaz (GUI)

```
Source Design History 
13 
14 
15 
16 
17 
18 
19 
20 
21 
22 
23 
24 
25 
26 
27 
28 
29 
30 
31 
32 
33 
34 
35 
36 
37 
38 
39 
40 
41 
42 
43 
44 
45 
46 
47 
48 
49 
50 
51 
52 
53 
54 
55 
56 
57 
58 
59 
60 
61 
62 
63 
64 
65 
66 
67 
68 
69 
70 
71 
72 
73 
74 
75 
76 
77 
78 
79 
80 
81 
82 
83 
84 
85 
86 
87 
88 
89 
90 
91 
92 
93 
94 
95 
96 
97 
98 
99 
100 
101 
102 
103 
104 
105 
106 
107 
108 
109 
110 
111 
112 
113 
114 
115 
116 
117 
118 
119 
120 
121 
122 
123 
124 
125 
126 
127 
128 
129 
130 
131 
132 
133 
134 
135 
136 
137 
138 
139 
140 
141 
142 
143 
144 
145 
146 
147 
148 
149 
150 
151 
152 
153 
154 
155 
156 
157 
158 
159 
160 
161 
162 
163 
164 
165 
166 
167 
168 
169 
170 
171 
172 
173 
174 
175 
176 
177 
178 
179 
180 
181 
182 
183 
184 
185 
186 
187 
188 
189 
190 
191 
192 
193 
194 
195 
196 
197 
198 
199 
200 
201 
202 
203 
204 
205 
206 
207 
208 
209 
210 
211 
212 
213 
214 
215 
216 
217 
218 
219 
220 
221 
222 
223 
224 
225 
226 
227 
228 
229 
230 
231 
232 
233 
234 
235 
236 
237 
238 
239 
240 
241 
242 
243 
244 
245 
246 
247 
248 
249 
250 
251 
252 
253 
254 
255 
256 
257 
258 
259 
260 
261 
262 
263 
264 
265 
266 
267 
268 
269 
270 
271 
272 
273 
274 
275 
276 
277 
278 
279 
280 
281 
282 
283 
284 
285 
286 
287 
288 
289 
290 
291 
292 
293 
294 
295 
296 
297 
298 
299 
300 
301 
302 
303 
304 
305 
306 
307 
308 
309 
310 
311 
312 
313 
314 
315 
316 
317 
318 
319 
320 
321 
322 
323 
324 
325 
326 
327 
328 
329 
330 
331 
332 
333 
334 
335 
336 
337 
338 
339 
340 
341 
342 
343 
344 
345 
346 
347 
348 
349 
350 
351 
352 
353 
354 
355 
356 
357 
358 
359 
360 
361 
362 
363 
364 
365 
366 
367 
368 
369 
370 
371 
372 
373 
374 
375 
376 
377 
378 
379 
380 
381 
382 
383 
384 
385 
386 
387 
388 
389 
390 
391 
392 
393 
394 
395 
396 
397 
398 
399 
400 
401 
402 
403 
404 
405 
406 
407 
408 
409 
410 
411 
412 
413 
414 
415 
416 
417 
418 
419 
420 
421 
422 
423 
424 
425 
426 
427 
428 
429 
430 
431 
432 
433 
434 
435 
436 
437 
438 
439 
440 
441 
442 
443 
444 
445 
446 
447 
448 
449 
450 
451 
452 
453 
454 
455 
456 
457 
458 
459 
460 
461 
462 
463 
464 
465 
466 
467 
468 
469 
470 
471 
472 
473 
474 
475 
476 
477 
478 
479 
480 
481 
482 
483 
484 
485 
486 
487 
488 
489 
490 
491 
492 
493 
494 
495 
496 
497 
498 
499 
500 
501 
502 
503 
504 
505 
506 
507 
508 
509 
510 
511 
512 
513 
514 
515 
516 
517 
518 
519 
520 
521 
522 
523 
524 
525 
526 
527 
528 
529 
530 
531 
532 
533 
534 
535 
536 
537 
538 
539 
540 
541 
542 
543 
544 
545 
546 
547 
548 
549 
550 
551 
552 
553 
554 
555 
556 
557 
558 
559 
560 
561 
562 
563 
564 
565 
566 
567 
568 
569 
570 
571 
572 
573 
574 
575 
576 
577 
578 
579 
580 
581 
582 
583 
584 
585 
586 
587 
588 
589 
590 
591 
592 
593 
594 
595 
596 
597 
598 
599 
600 
601 
602 
603 
604 
605 
606 
607 
608 
609 
610 
611 
612 
613 
614 
615 
616 
617 
618 
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
```



```

1  }
2
3  // Parse args the command line arguments
4
5  public static void main(String args[]) {
6      // Set the Nimbus look and feel.
7      //
8      // For more information on this, see
9      // "What's New in Java 9"
10     //
11     // Create and display the form
12     java.awt.EventQueue.invokeLater(new Runnable() {
13         public void run() {
14             new gui().setVisible(true);
15         }
16     });
17 }
18
19 // Variables declaration here:
20
21 public javax.swing.JTextField pNama;
22 public javax.swing.JTextField pJalan;
23 public javax.swing.JTextField pKantor;
24 public javax.swing.JTextField pKode;
25 public javax.swing.JTextField pKelas;
26 public javax.swing.JTextField pNomb;
27 public javax.swing.JButton pSim;
28 public javax.swing.JComboBox<String> pJenis;
29 private javax.swing.JLabel pLabel1;
30 private javax.swing.JLabel pLabel2;
31 private javax.swing.JLabel pLabel3;
32 private javax.swing.JLabel pLabel4;
33 private javax.swing.JLabel pLabel5;
34 private javax.swing.JLabel pLabel6;
35 private javax.swing.JLabel pLabel7;
36 private javax.swing.JLabel pLabel8;
37 private javax.swing.JLabel pLabel9;
38 private javax.swing.JPanel pPanel1;
39 private javax.swing.JPanel pPanel2;
40 private javax.swing.JScrollPane pScrollPane;
41 public javax.swing.JButton pSim;
42 public javax.swing.JTable pTab;
43 public javax.swing.JLabel pSim;
44 // End of variables declaration
45
46 }

```

## 7. Interfaz

TallerP

Silva Santiago, Manrique Brayan, Bernal Josue

## Formulario

Apellido	Nombre	Edad	Nacionalidad	Titulo academ...	Genero	Sueldo

Ingrese sus datos

Nombre:

Apellidos:

Edad:  años

Nacionalidad:

Titulo academico:

Genero:

Sueldo:

```
private void ApellActionPerformed(java.awt.event.ActionEvent evt) {  
    // GUARDAR VARIABLE DEL APELLIDO  
}
```