



TECNICATURA UNIVERSITARIA EN DISEÑO
INTEGRAL DE VIDEOJUEGOS

FACULTAD DE INGENIERÍA
Universidad Nacional de Jujuy



FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS

Trabajo Practico N°1

Facundo Condori

TUV000667

Profesores:

Mg. Ing. Ariel Alejandro Vega

Ing. Carolina Cecilia Apaza

Año

Indice

Punto 1:	3
Punto 2:	4
Punto 3:	6
Punto 4:	6
Valores asignados a las variables:	6
Valores asignados a las variables:	6
Punto 5:	10
Punto 6:	14
Punto 7:	15
Punto 8:	16
Punto 9:	17
Punto 10:	17
Punto 11:	18
Punto 12:	19
Punto 13:	21
Punto 14:	23
Punto 15:	26
Punto 16:	28
Punto 17:	30
Punto 18:	33
Punto 19:	34
Punto 20:	36
Punto 21:	38
Punto 22:	41

Punto 1:

Ejercicio 1: Evaluar(obtener resultado) la siguiente expresión para

$A = 2$ y $B = 5$

Desarrollo del punto

$3 * A - 4 * B / A ^ 2$

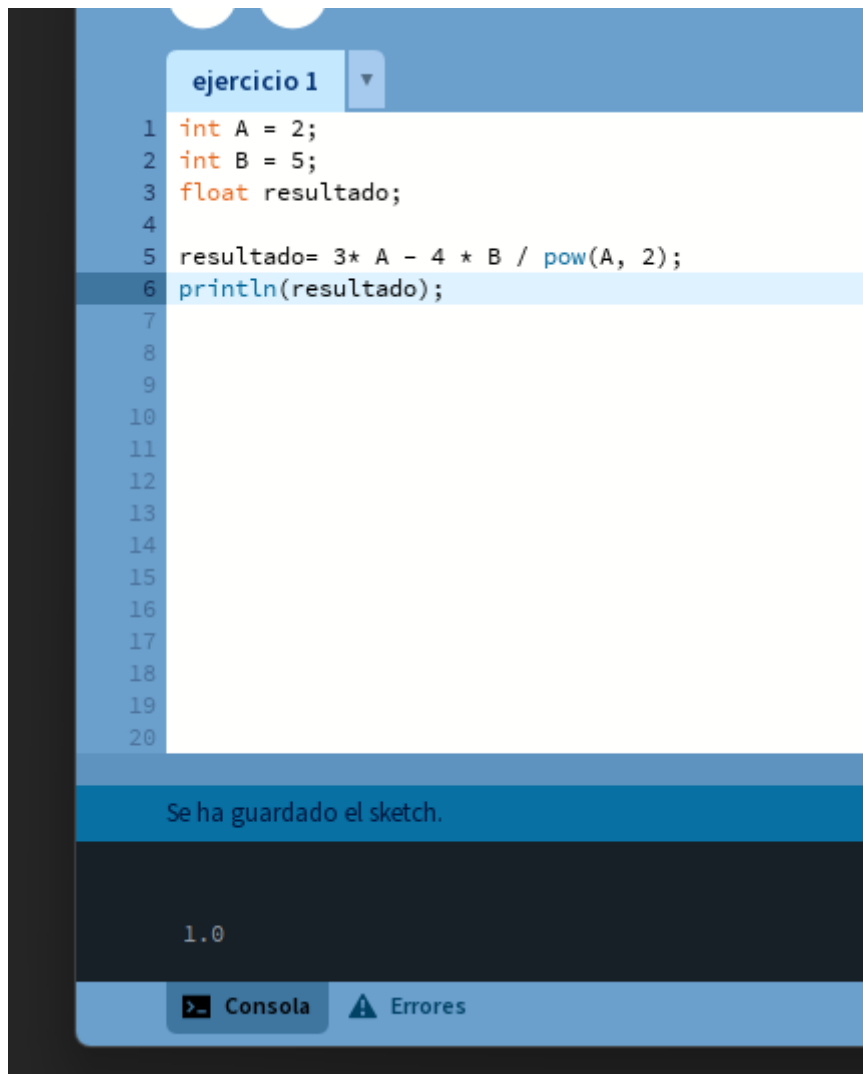
$(3*A)-(4*B/(A^2))$

$3 * A - 4 * B / 4$

$6 - 20 / 4$

$6 - 5$

1



```
ejercicio 1
1 int A = 2;
2 int B = 5;
3 float resultado;
4
5 resultado= 3* A - 4 * B / pow(A, 2);
6 println(resultado);
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

Se ha guardado el sketch.

1.0

Consola Errores



Punto 2:

Ejercicio 2: Evaluar la siguiente expresión

$$4 / 2 * 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 * 2$$

Desarrollo del punto

$$(4 / 2 * 3 / 6) + (6 / 2 / 1 / (5 ^ 2) / 4 * 2)$$

$$(4 / 2 * 3 / 6) + (6 / 2 / 1 / 25 / 4 * 2)$$

$$(2 * 3 / 6) + (6 / 2 / 1 / 25 / 4 * 2)$$

$$(6 / 6) + (6 / 2 / 1 / 25 / 4 * 2)$$

$$1 + (3 / 1 / 25 / 4 * 2)$$

$$1 + (3 / 25 / 4 * 2)$$

$$1 + (3 / 25 / 8)$$

$$1 + (3 / 200)$$

$$1 + 0,015$$

$$1,015$$

$$1,06$$



```
1 float resolver;  
2 resolver = (4 / 2 * 3 / 6) + (6 / 2 / 1 / pow(5, 2) / 4 * 2);  
3 println("resultado ejercicio 2)" + resolver);  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
--  
  
resultado ejercicio 2)1.06
```

Punto 3:

Ejercicio 3: Escribir las siguientes expresiones algebraicas como expresiones algorítmicas (en su forma aritmética dentro del algoritmo). En este caso no se pide evaluarlas ni programarlas.

Punto 4:

Ejercicio 4: Evaluar las siguientes expresiones aritméticas, para lo cual indicar en el caso de las variables, el valor indicado. Luego escribirlas como expresiones algebraicas.

Desarrollo del punto

a) $b^2 - 4 * a * c$

$b^2 - 4ac$

Valores asignados a las variables:

$b=2$

$a=2$

$c=2$

a) $(b^2) - (4 * a * c)$

$4 - (4 * a * c)$

$4 - (8 * c)$

$4 - 16$

-12

b) $3 * X^4 - 5 * X^3 + X^{12} - 17$

$3X^4 - 5X^3 + X^{12} - 17$

Valores asignados a las variables:

$X=2$

b) $(3 * (X^4)) - (5 * (X^3) + X^{12}) - 17$

$(3 * 16) - (5 * (X^3) + X^{12}) - 17$

$$(3 * 16) - (5 * 8 + X 12) - 17$$

$$48 - (5 * 8 + X 12) - 17$$

$$48 - (40 + X 12) - 17$$

$$48 - (40 + 24) - 17$$

$$48 - 64 - 17$$

$$- 16 - 17$$

$$-33$$

```
ejercicio 1  ejercicio 2  ejercicio 4  ▼
1  int b = 2;
2  int a = 2;
3  int c = 2;
4  float resuelve;
5
6  resultado = pow(b, 2) - 4 * a * c;
7  println(resultado);
8
9  int X = 2;
10 float reseuelve;
11 resultado = (3 * pow(X, 4)) - (5 * pow(X, 3) + X * 12) - 17;
12 println(resultado);
13
14
15
16
17
18
19
```

```
1.0
1.06
-12.0
-33.0
```

$$c) (b + d) / (c + 4)$$

$$Bd/c4$$

$$b = h$$

$$b = 2$$

$$d = 1$$

$c = 3$

$(b + d) / (c + 4)$

$3/(c + 4)$

$3/7$

0.5

ejercicio 1	ejercicio 2	ejercicio 4	ejercicio
<pre> 1 int h = 2; ///cambie la b por la h/// 2 int d = 1; 3 int c = 3; 4 resultado = (float) (b + d) / (c + 4); 5 println("resultado c): " + resultado); 6 7 /// me tira error, por eso lo hice aparte /// 8 9 10 11 12 13 14 15 16 17 18 19 </pre>			
<p>El valor de la variable local «h» no se utiliza</p> <pre> 1.06 resultado a): -12.0 resultado b): -33.0 resultado c): 0.5 resultado d): 0.0 </pre>			
<p>Consola Errores</p>			

$d) (x^2 + y^2)^{(1/2)}$

$X = 0$

$Y = 0$

$(x^2 + y^2)^{(1/2)}$

$(0 + y^2)^{(1/2)}$

$(0 + 0)^{(1/2)}$

$0^{1/2}$

0^2

0



```
ejercicio 1 | ejercicio 2 | ejercicio 4 | <
1 int x = 0;
2 int y = 0;
3 float resultado;
4 resultado = sqrt(pow(x, 2) + pow(y, 2));
5 println("resultado d): " + resultado);
6
7
8
9
0
1
2
3
4
5
6
7
8
9
0
1.06
resultado a): -12.0
resultado b): -33.0
resultado c): 0.5
resultado d): 0.0
Consola Errores
```



Para aclarar que indicamos con "Luego escribirlas como expresiones algebraicas" lo aplicamos con el punto

a) $b^2 - 4.a.c$

b) $3 \cdot X^4 - 5 \cdot X^3 + X^2 - 17$

c) $(b + d) / (c + 4)$

d) $(x^2 + y^2)^{1/2}$

 <p>TUDI Videojuegos Fundamentos de Programación Orientada a Objetos</p>	<p>FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS FACULTAD DE INGENIERÍA Universidad Nacional de Jujuy Trabajo Practico N° / Actividad</p>	
---	---	---

Punto 5:

Ejercicio 5: Si el valor de A es 4, el valor de B es 5 y el valor de C es 1, evaluar las siguientes expresiones:

Desarrollo del punto



a) $B * A - B^2 / 4 * C$

$(B * A) - (B^2 / 4 * C)$

$20 - (B^2 / 4 * C)$

$20 - (25 / 4 * C)$

$20 - 6,25$

$13,75$

```
sketch 240412a
1 println("resultado ejercicio 5");
2 float A = 4;
3 float B = 5;
4 float C = 1;
5 float resolver;
6
7 resolver = (B * A) - (float) Math.pow(B, 2) / (4 * C);
8 System.out.println(resolver);
9
10
11
12
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
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
```

b) $(A * B) / 3^2$

$20 / 6$

2.2223

```
sketch 240412a
1 println("resultado ejercicio 5");
2 float A = 4;
3 float B = 5;
4 float C = 1;
5 float resolver;
6
7 resolver = (B * A) - (float) Math.pow(B, 2) / (4 * C);
8 resolver = (A * B) / (float) Math.pow(3, 2);
9 System.out.println(resolver);
10
11
12
13
14
15
16
17
18
19
20
```

```
resultado ejercicio 5
2.2222223
```

Consola Errores

c) $((B + C) / 2 * A + 10) * 3 * B) - 6$

$((6 / 2 * A + 10) * 3 * B) - 6$

$((3 * A + 10) * 3 * B) - 6$

$((12 + 10) * 3 * B) - 6$

$(22 * 3 * B) - 6$

$(66 * B) - 6$

$330 - 6$

324



```
sketch 240412a
1 println("resultado ejercicio 5");
2 float A = 4;
3 float B = 5;
4 float C = 1;
5 float resolver;
6
7 resolver = (B * A) - (float) Math.pow(B, 2) / (4 * C);
8 resolver = (A * B) / (float) Math.pow(3, 2);
9 resolver = (((B + C) / 2 * A + 10) * 3 * B) - 6;
10 System.out.println(resolver);
```

```
resultado ejercicio 5
324.0
```



Punto 6:

Ejercicio 6: Para $x=3$, $y=4$; $z=1$, evaluar el resultado de

$$R1 = y+z$$

$$R2 = x \geq R1$$

Desarrollo del punto

$$R1 = y+z$$

5

$$R2 = x \geq R1$$

$$x \geq 5$$

Falso

```
1 println("Ejercicio 6:");
2 int x = 3;
3 int y = 4;
4 int z = 1;
5
6 int R1 = y + z;
7 boolean R2 = x >= R1;
8
9 println("R1 = " + R1);
10 println("R2 = " + R2);
11 exit();
```

Ejercicio 6):
R1 = 5
R2 = false

Punto 7:

Ejercicio 7: Para contador1=3, contador3=4, evaluar el resultado de

R1 = ++contador1

R2 = contador1 < contador2

Desarrollo del punto

```
ejercicio 7
1 println("resultado ejercicio 7:");
2 int contador1 = 3;
3 int contador2 = 4;
4 ///se me bugg el void setup()///
5     int R1 = ++contador1;
6     boolean R2 = contador1 < contador2;
7     println("R1: " + R1);
8     println("R2: " + R2);
9     noLoop();
10
11
12
13
14
15
16
17
18
19
20

resultado ejercicio 7):
R1: 4
R2: false
```

Punto 8:

Ejercicio 8: Para $a=31$, $b=-1$; $x=3$, $y=2$, evaluar el resultado de

$$a+b-1 < x*y$$

Desarrollo del punto

```
sketch 240413b
1 println("ejercicio 8:");
2   int a = 31;
3   int b = -1;
4   int x = 3;
5   int y = 2;
6   boolean resultado = (a + b - 1) < (x * y);
7   println(resultado);
8   noLoop();
9
10
11
12
13
14
15
16
17
18
19
20
ejercicio 8):
false
```


Punto 9:

Ejercicio 9: Para $x=6$, $y=8$, evaluar el resultado de

$!(x < 5) \text{ CC } !(y \geq 7)$

Desarrollo del punto

```
1 void setup() {  
2   println("resultado ejercicio 9");  
3   int x = 6;  
4   int y = 8;  
5   boolean resultado = !(x < 5) && !(y >= 7);  
6   println(resultado);  
7   noLoop();  
8 }  
9  
10  
11  
12  
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
```

resultado ejercicio 9
false

Punto 10:

Ejercicio 10: Para $i=22$, $j=3$, evaluar el resultado de

$!((i > 4) \parallel !(j \leq 6))$

Desarrollo del punto

```
sketch 240413d
1 void setup() {
2   println("ejercicio 10:");
3   int i = 22;
4   int j = 3;
5   boolean resultado = !((i > 4) || !(j <= 6));
6   println(resultado);
7   noLoop();
8 }
9
10
11
12
13
14
15
16
17
18
19
20
ejercicio 10):
false
```

Punto 11:

Ejercicio 11: Para $a=34$, $b=12$, $c=8$, evaluar el resultado de $!(a+b==c) \parallel (c!=0)CC(b-c \geq 19)$

Desarrollo del punto

```
sketch 240413e ▼
1 void setup() {
2   println("ejercicio 11:");
3   int a = 34;
4   int b = 12;
5   int c = 8;
6   boolean resultado = !(a + b == c) || (c != 0) && (b - c >= 19);
7   println(resultado);
8   noLoop();
9 }
10
11
12
13
14
15
16
17
18
19
ejercicio 11):
true
```

Punto 12:

Sección Análisis – Diseño y Codificación de algoritmos – Aplicación de estructuras de control Para cada ejercicio, en el archivo Word agregar las secciones de análisis y diseño, mientras que, para la codificación, crear el archivo de Processing.

Ejercicio 12: Un problema sencillo. Deberá pedir por teclado al usuario un nombre y posteriormente realizará la presentación en pantalla de un saludo con el nombre indicado.

Desarrollo del punto

Análisis

Datos de entrada:

Nombre del usuario: string

Datos de salida:

Saludo personalizado que incluya el nombre ingresado: string

Proceso:

Solicitar al usuario que ingrese su nombre.

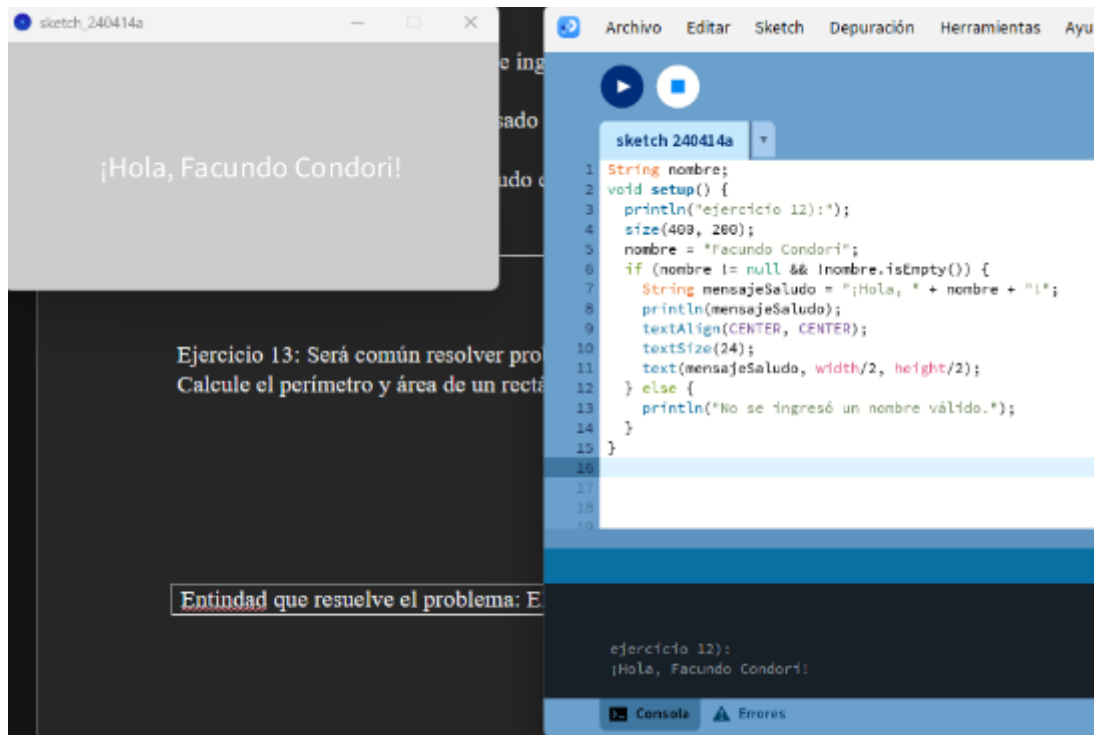
Leer y almacenar el nombre proporcionado por el usuario.

Construir un mensaje de saludo que incluya el nombre ingresado.

Mostrar el mensaje de saludo en pantalla.

Diseño:

Entidad que resuelve el problema: El programador
Variables iniciar el programa mostrarSaludo: String
Nombre del algoritmo nombre_saludo Proceso del algoritmo <ol style="list-style-type: none"> 1. Solicitar entrada 2. <i>Mostrar</i> un mensaje solicitando al usuario que ingrese su nombre 3. <i>Leer</i> Entrada 4. <i>Leer</i> y almacenar el nombre ingresado por el usuario 5. Construir Saludo 6. Combinar el nombre ingresado en un mensaje de saludo 7. <i>Mostrar</i> Saludo 8. Imprimir el mensaje de saludo en la pantalla 9. Fin



Punto 13:

Ejercicio 13: Será común resolver problemas utilizando variables.
Calcule el perímetro y área de un rectángulo dada su base y su altura.

Desarrollo del punto

Análisis:

Datos de entrada:

Base del rectángulo: real.

Altura del rectángulo: real.

Datos de salida:

Perímetro del rectángulo: real.

Área del rectángulo: real.

Fórmulas para el Cálculo:

El perímetro de un rectángulo se calcula como: $2 * (base + altura)$.

El área de un rectángulo se calcula como: $base * altura$.

Entidad que resuelve el problema: El programador
<p>Entidad que resuelve el problema: El programador</p> <p>Variables</p> <p>base: Real // almacena la base del rectángulo</p> <p>altura: Real // almacena la altura del rectángulo</p> <p>perimetro: Real // almacena el valor del perímetro del rectángulo</p> <p>area: Real // almacena el valor del área del rectángulo</p> <p>Variables</p> <p>base: Real // almacena la base del rectángulo</p> <p>altura: Real // almacena la altura del rectángulo</p> <p>perimetro: Real // almacena el valor del perímetro del rectángulo</p> <p>area: Real // almacena el valor del área del rectángulo</p>
<p>Nombre del algoritmo: calcular_perimetro_area_rectangulo</p> <p>Proceso del algoritmo:</p> <ol style="list-style-type: none"> 1. Leer base desde la entrada de usuario. 2. Leer altura desde la entrada de usuario. 3. Calcular el perímetro del rectángulo usando la fórmula: $perimetro = 2 * (base + altura)$. 4. Calcular el área del rectángulo usando la fórmula: $area = base * altura$. 5. Mostrar el valor del perímetro y el área del rectángulo en la pantalla.

```
sketch 240414b
1 void setup() {
2   println("ejercicio 13:");
3   float base = 20.0;
4   float altura = 10.0;
5   float perimetro = 2 * (base + altura);
6   float area = base * altura;
7   println("Perímetro del rectángulo: " + perimetro);
8   println("Área del rectángulo: " + area);
9   noLoop();
10 }
11
12
13
14
15
16
17
18
19
```

```
ejercicio 13):
Perímetro del rectángulo: 60.0
Área del rectángulo: 200.0
```

Console Errores

Punto 14:

Ejercicio 14: Una ayuda importante al momento de resolver problemas con algoritmos es asumir que su gran amigo son las matemáticas. Obtenga la hipotenusa de un triángulo rectángulo conociendo sus catetos.

Desarrollo del punto

Datos de Entrada:

catetoA: real

catetoB: real

Datos de Salida:

hipotenusa: real

Fórmula Utilizada:

El cálculo de la hipotenusa (c) de un triángulo rectángulo se basa en el teorema de Pitágoras, que establece que en un triángulo rectángulo con catetos a y b, la hipotenusa c se calcula como:

$$C = (\text{raiz})a^2 + b^2(\text{fin de raiz})$$

a y b son las longitudes de los catetos del triángulo rectángulo.

Entidad que Resuelve el Problema: Calculadora de Hipotenusa
Variables: cateto1: Real cateto2: Real hipotenusa: Real
Nombre del Algoritmo: calcular_hipotenusa Proceso del Algoritmo: <ol style="list-style-type: none"> 1. <i>Leer</i> cateto1 2. <i>Leer</i> cateto2 3. Calcular la hipotenusa del triángulo rectángulo utilizando el teorema de Pitágoras: $\text{hipotenusa} = (\text{raiz})\text{cateto1}^2 + \text{cateto2}^2(\text{fin de raiz})$ 4. <i>Mostrar</i> el valor de hipotenusa en la consola.



sketch 240414c

```
1 void setup() {  
2   println("ejercicio 14:");  
3   float catetoA = 10;  
4   float catetoB = 5;  
5   float hipotenusa = sqrt(catetoA * catetoA + catetoB * catetoB);  
6   println("La longitud de la hipotenusa es: " + hipotenusa);  
7   noLoop();  
8 }
```

```
ejercicio 14):  
La longitud de la hipotenusa es: 11.18034
```

Punto 15:

Ejercicio 15: Si viste algo de los apuntes y vídeos, esto debería ser muy fácil de resolver. Dados dos números permita calcular la suma, resta, multiplicación y división de estos. Considere que cada una de estas operaciones es un algoritmo cuando realice el diseño. Obviamente muestre los resultados.

Análisis

Datos de Entrada:

numeroA: Real

numeroB: Real

Datos de Salida:

suma: Real

resta: Real

multiplicacion: Real

division: Real

Quién realiza el proceso: Una calculadora

Descripción del Proceso:

La calculadora realiza las siguientes operaciones:

Suma: $\text{suma} = \text{numeroA} + \text{numeroB}$

Resta: $\text{resta} = \text{numeroA} - \text{numeroB}$

Multiplicación: $\text{multiplicacion} = \text{numeroA} * \text{numeroB}$

División:

Si numeroB es distinto de cero: $\text{division} = \text{numeroA} / \text{numeroB}$

Si numeroB es igual a cero: No se puede realizar la división (mensaje de error)

ENTIDAD QUE RESUELVE EL PROBLEMA: Calculadora
Variables: numeroA: Real numeroB: Real resultadoSuma: Real resultadoResta: Real resultadoMultiplicacion: Real resultadoDivision: Real
Nombre del Algoritmo: calcular_operaciones Proceso del Algoritmo: Leer numeroA // Solicitar al usuario que ingrese el primer número (numeroA).



Leer numeroB //

Solicitar al usuario que ingrese el segundo número (numeroB).

Calcular y Mostrar la Suma //

Calcular la suma de numeroA y numeroB y almacenar el resultado en resultadoSuma.

Mostrar el resultado de la suma (resultadoSuma).

Calcular y Mostrar la Resta //

Calcular la resta de numeroA menos numeroB y almacenar el resultado en resultadoResta.

Mostrar el resultado de la resta (resultadoResta).

Calcular y Mostrar la Multiplicación //

Calcular el producto de numeroA por numeroB y almacenar el resultado en resultadoMultiplicacion.

Mostrar el resultado de la multiplicación (resultadoMultiplicacion).

Calcular y Mostrar la División //

Verificar si numeroB es distinto de cero:

Si es así, calcular la división de numeroA entre numeroB y almacenar el resultado en resultadoDivision.

Mostrar el resultado de la división (resultadoDivision).

Si numeroB es cero, mostrar un mensaje indicando que la división por cero no es posible.

```
ejercicio 15
1 void setup() {
2   println("Ejercicio 15:");
3   float numero1 = 10;
4   float numero2 = 20;
5   float suma = sumarNumeros(numero1, numero2);
6   println("Suma: " + suma);
7   float resta = restarNumeros(numero1, numero2);
8   println("Resta: " + resta);
9   float multiplicacion = multiplicarNumeros(numero1, numero2);
10  println("Multiplicación: " + multiplicacion);
11  if (numero2 != 0) {
12    float division = dividirNumeros(numero1, numero2);
13    println("División: " + division);
14  } else {
15    println("No se puede dividir entre cero.");
16  }
17  noLoop();
18 }
19 float sumarNumeros(float a, float b) {
20   return a + b;
21 }
22 float restarNumeros(float a, float b) {
23   return a - b;
24 }
25 float multiplicarNumeros(float a, float b) {
26   return a * b;
27 }
28 float dividirNumeros(float a, float b) {
29   return a / b;
30 }
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
```

```
Ejercicio 15:
Suma: 30.0
Resta: -10.0
Multiplicación: 200.0
División: 0.5
```

Punto 16:

Ejercicio 16: Necesitamos convertir una temperatura Fahrenheit en grados Celsius. Si no conoce la forma en la que se realiza esta conversión, debería investigarlo; para eso sirve la etapa de análisis. Pero como somos buenos, daremos una ayuda.

Desarrollo del punto

Convertir una temperatura dada en grados Fahrenheit a grados Celsius.

Análisis

Datos de Entrada:

temperaturaFahrenheit: Real

Datos de Salida:

temperaturaCelsius: Real

Proceso:

Quién realiza el proceso: Algoritmo de conversión

Descripción del Proceso:

La conversión de temperatura de Fahrenheit a Celsius se realiza mediante la siguiente fórmula:

$$\text{temperaturaCelsius} = (\text{temperaturaFahrenheit} - 32) \times \frac{5}{9}$$

Se solicita al usuario ingresar la temperatura en grados Fahrenheit (temperaturaFahrenheit).

Se aplica la fórmula de conversión para calcular la temperatura equivalente en grados Celsius (temperaturaCelsius).

Se muestra el resultado de la conversión (temperaturaCelsius).



ENTIDAD QUE RESUELVE EL PROBLEMA: Conversor de temperatura

VARIABLES

tempFahrenheit, tempCelsius: Real // almacenan la temperatura en Fahrenheit y Celsius respectivamente

NOMBRE ALGORITMO: convertir_temperatura

PROCESO DEL ALGORITMO

1. Leer tempFahrenheit
2. $\text{tempCelsius} \leftarrow (\text{tempFahrenheit} - 32) \cdot (5/9)$ // fórmula de conversión
3. Mostrar tempCelsius

```
1 void setup() {  
2   println("ejercicio 16):");  
3   float temperaturaFahrenheit = 30.0;  
4   float temperaturaCelsius = convertirFahrenheitACelsius(temperaturaFahrenheit);  
5   println("La temperatura de " + temperaturaFahrenheit + " °F equivale a " + temperaturaCelsius);  
6   noLoop();  
7 }  
8 float convertirFahrenheitACelsius(float temperaturaF) {  
9   return (temperaturaF - 32) * (5.0 / 9.0);  
10 }  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20
```

ejercicio 16):
La temperatura de 30.0 °F equivale a -1.1111112 °C

Punto 17:

Ejercicio 17: Si queremos representar personajes o power ups (premios) en la pantalla debemos primero ubicarlos en alguna posición dentro de la pantalla. Imagine que está en un juego donde un power up desaparece porque el personaje se acerca a una distancia de x unidades, sin importar por donde se acerque. Por tanto, para que desaparezca, en primer lugar, hay que determinar esa distancia. La forma de representar la posición de un objeto en la pantalla es a través de las coordenadas de un punto. Suponga que la posición de Link está representada por la coordenada (x_1, y_1) , mientras que las de la caja de tesoro se halla en la posición (x_2, y_2) . Si observa con detenimiento se observa la conformación de un triángulo rectángulo, por lo que es posible aplicar Pitágoras para obtener la distancia. Para esto debe calcular el tamaño de los catetos y luego aplicar el teorema. Halle la distancia entre ambos objetos. Cuando programe, represente a Link con un Circulo, y al tesoro con un cuadrado. Además, mueva a Link mediante el mouse.

Desarrollo del punto

Datos de Entrada:

Coordenadas de Link:

x_1 : Real

y_1 : Real

Coordenadas del Tesoro:

x_2 : Real

y_2 : Real

Datos de Salida:

distancia: Real

Proceso:

Quién realiza el proceso: Algoritmo de juego

Descripción del Proceso:

Solicitar al usuario que mueva el mouse para actualizar la posición de Link (x_1, y_1) .

Calcular la distancia entre Link y el tesoro utilizando el teorema de Pitágoras: $\text{distancia} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ (fin de la raíz)



Entidad que resuelve el problema: Calculadora

Variables:

x1, y1: Real

x2, y2: Real

distancia: Real

Nombre del algoritmo: calcular_distancia

Proceso del algoritmo:

1. Leer x1, y1
2. Leer x2, y2
3. Calcular el tamaño del cateto horizontal: $\Delta x = |x2 - x1|$.
4. Calcular el tamaño del cateto vertical: $\Delta y = |y2 - y1|$.
5. Calcular la distancia usando el teorema de Pitágoras: $distancia = \sqrt{(\Delta x^2 + \Delta y^2)}$.
6. Mostrar distancia.

sketch 240415c

```
1 float x1, y1;
2 float x2, y2;
3
4 void setup() {
5   println("ejercicio 17:");
6   size(400, 400);
7   x1 = width / 2;
8   y1 = height / 2;
9   x2 = random(width);
10  y2 = random(height);
11 }
12
13 void draw() {
14   background(255);
15
16   fill(0, 0, 255);
17   ellipse(x1, y1, 30, 30);
18   fill(255, 0, 0);
19   rect(x2, y2, 30, 30);
20 }
```

```
ejercicio 17):
Distancia entre Link y el tesoro: 157.34404
Distancia entre Link y el tesoro: 157.34404
Distancia entre Link y el tesoro: 157.34404
```

Consola Errores



FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS
TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS
FACULTAD DE INGENIERÍA
Universidad Nacional de Jujuy
Trabajo Practico N° / Actividad



```
1 float x1, y1;
2 float x2, y2;
3
4 void setup() {
5   println("ejercicio 17:");
6   size(400, 400);
7   x1 = width / 2;
8   y1 = height / 2;
9   x2 = random(width);
10  y2 = random(height);
11 }
12
13 void draw() {
14   background(255);
15
16   fill(0, 0, 255);
17   ellipse(x1, y1, 30, 30);
18   fill(255, 0, 0);
19   rect(x2, y2, 30, 30);
20 }
```

Distancia entre Link y el tesoro: 159.21936
Distancia entre Link y el tesoro: 159.21936
Distancia entre Link y el tesoro: 159.21936

Punto 18:

Ejercicio 18: Desarrolle el análisis y diseño de un algoritmo que permita obtener las raíces de una ecuación de segundo grado. Además, utilice la estructura según para el análisis de la discriminante de la ecuación cuadrática. Obviamente codifique en Processing.

Desarrollo del punto

Descripción del Problema:

Se desea desarrollar un algoritmo que permita calcular las raíces de una ecuación de segundo grado de la forma $ax^2+bx+c=0$, donde a, b y c son coeficientes reales y a (no igual) 0.

Datos de Entrada:

a: Real

b: Real

c: Real

Datos de Salida:

raiz1: Real

raiz2: Real

Proceso:

Calcular el discriminante de la ecuación cuadrática y luego determinar el tipo de raíces (reales o complejas) según el valor de Δ . Posteriormente, se utilizan fórmulas específicas para calcular las raíces en cada caso.

Este proceso te permitirá resolver cualquier ecuación cuadrática y determinar las raíces correspondientes dependiendo de las características del discriminante.

Entidad que resuelve el problema: Calculadora
Variables: a, b, c: Real discriminante, raiz1, raiz2: Real
Nombre del algoritmo: calcular_raices Proceso del algoritmo: 1. Leer a, b, c // Coeficientes de la ecuación cuadrática. 2. Calcular el discriminante: $\text{discriminante} = b^2 - 4ac$. 3. Según el valor del discriminante: - Si $\text{discriminante} > 0$: - Calcular las raíces reales: $\text{raiz1} = (-b + \sqrt{\text{discriminante}}) / (2a)$ y $\text{raiz2} = (-b - \sqrt{\text{discriminante}}) / (2a)$. - Mostrar raiz1 y raiz2 como las raíces reales de la ecuación. - Si $\text{discriminante} = 0$: - Calcular la raíz real doble: $\text{raiz1} = \text{raiz2} = -b / (2a)$. - Mostrar raiz1 y raiz2 como la raíz real doble de la ecuación.

- Si discriminante < 0 :
- Mostrar un mensaje indicando que las raíces son complejas y no se pueden calcular en este algoritmo.

```
void setup() {  
    println("ejercicio 18:");  
    float a = 1.0;  
    float b = -3.0;  
    float c = 2.0;  
    float discriminante = b*b - 4*a*c;  
  
    if (discriminante > 0) {  
        float raiz1 = (-b + sqrt(discriminante)) / (2*a);  
        float raiz2 = (-b - sqrt(discriminante)) / (2*a);  
        println("Raíces reales distintas:");  
        println("Raíz 1: " + raiz1);  
        println("Raíz 2: " + raiz2);  
    } else if (discriminante == 0) {  
        float raiz = -b / (2*a);  
        println("Raíz real doble:");  
        println("Raíz: " + raiz);  
    } else {  
        float parteReal = -b / (2*a);  
        float parteImag = sqrt(-discriminante) / (2*a);  
        println("Raíces complejas:");  
        println("Raíz 1: " + parteReal + " + " + parteImag + "i");  
        println("Raíz 2: " + parteReal + " - " + parteImag + "i");  
    }  
}
```

```
ejercicio 18):  
Raíces reales distintas:  
Raíz 1: 2.0  
Raíz 2: 1.0
```

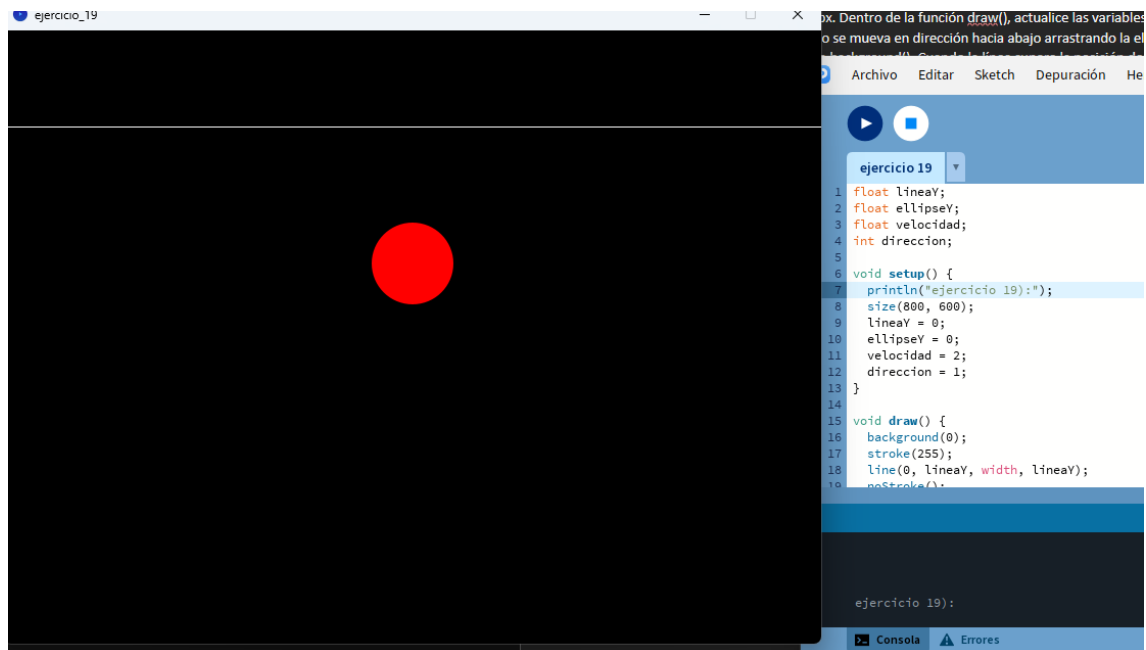
Consola Errores

Punto 19:

Ejercicio 19: Declare las variables necesarias para dibujar una línea que se dibuja desde las coordenadas iniciales del lienzo y se extiende por todo el ancho. Sobre el punto medio de la línea y a una distancia de 40px (en sentido vertical desde la línea) dibuje una elipse que tenga como ancho 80px y de alto 80px. Dentro de la función draw(), actualice las variables necesarias para que la línea desde su inicio se mueva en dirección hacia abajo arrastrando la elipse.

Mantenga en cero el valor para background(). Cuando la línea supere la posición de la altura del lienzo, debe invertir su sentido, es decir dirigirse hacia arriba arrastrando la elipse. Cuando la línea alcance nuevamente el valor 0 para su posición en y, el desplazamiento debe ser hacia abajo y así sucesivamente. El lienzo debería verse como en las siguientes figuras.

Desarrollo del punto



Entidad que resuelve el problema: Dibujador de línea con elipse

Variables:

xLinea: Real

yLinea: Real

yElipse: Real

direccion: Booleano



Nombre del algoritmo: dibujarLineaConElipse

Proceso del algoritmo:

1. Inicializar xLinea
2. Inicializar yLinea
3. Inicializar yElipse
4. Inicializar direccion

Dentro de la función draw():

1. Limpiar el lienzo con background(0).
2. Dibujar una línea horizontal que se extiende por todo el ancho del lienzo desde (xLinea, yLinea) hasta (width, yLinea).
3. Dibujar una elipse en el punto medio de la línea, en las coordenadas (xLinea + (width - xLinea) / 2, yElipse), con ancho 80 y alto 80.
4. Actualizar la posición de yLinea y yElipse para mover la línea y la elipse hacia abajo o hacia arriba, dependiendo de la dirección actual.
5. Verificar si la línea ha superado los límites del lienzo (height). Si es así, invertir la dirección de movimiento.
6. Repetir este proceso en cada iteración de draw() para crear la animación.

	<p>FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS</p> <p>FACULTAD DE INGENIERÍA Universidad Nacional de Jujuy Trabajo Practico N° / Actividad</p>	
---	--	---

Punto 20:

Ejercicio 20: Dibuje en toda la extensión del lienzo de (440, 420) rectángulos de idénticas medidas (40 ancho y 20 de alto) y que mantengan una distancia de 20 pixeles entre ellos tanto horizontal como verticalmente. Utilice la estructura de control repetitiva for. El lienzo debería verse así:

Desarrollo del punto

Análisis

Datos de Entrada:

Dimensiones del lienzo: Ancho de 440 pixeles y alto de 420 pixeles.

Medidas del rectángulo: Ancho de 40 pixeles y alto de 20 pixeles.

Espacio entre rectángulos: 20 pixeles tanto horizontal como verticalmente.

Datos de Salida:

Un lienzo que muestra rectángulos de tamaño 40x20 pixeles, distribuidos uniformemente con un espacio de 20 pixeles entre ellos tanto horizontal como verticalmente.

Proceso:

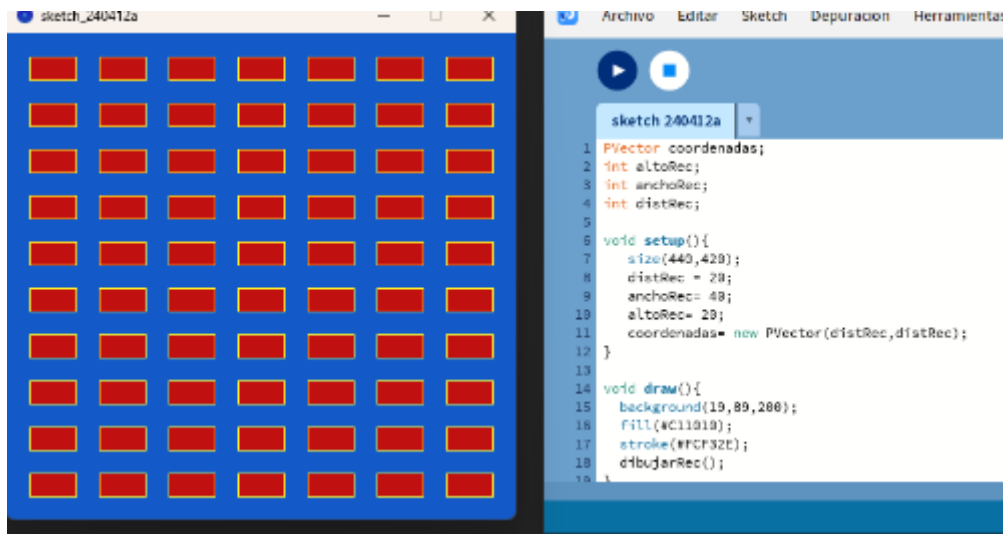
Utilizaremos una estructura de control repetitiva for para iterar sobre las filas y columnas del lienzo y calcular las posiciones de cada rectángulo.

En cada iteración:

Calcular la posición x y y del rectángulo utilizando las dimensiones del rectángulo y el espacio entre ellos.

Dibujar un rectángulo en la posición calculada.

ENTIDAD QUE RESUELVE EL PROBLEMA: Dibujador de Rectángulos
<p>VARIABLES</p> <p>anchoLienzo: Real // Ancho del lienzo en pixeles</p> <p>altoLienzo: Real // Alto del lienzo en pixeles</p> <p>anchoRect: Real // Ancho de los rectángulos en pixeles</p> <p>altoRect: Real // Alto de los rectángulos en pixeles</p> <p>espacioEntreRect: Real // Espacio entre los rectángulos en pixeles</p>
<p>NOMBRE ALGORITMO: Dibujar_Rectangulos</p> <p>PROCESO DEL ALGORITMO</p> <ol style="list-style-type: none"> 1. Leer anchoLienzo 2. Leer altoLienzo 3. Leer anchoRect 4. Leer altoRect 5. Leer espacioEntreRect 6. Configurar el lienzo con dimensiones (anchoLienzo, altoLienzo) 7. Establecer el color de fondo del lienzo en blanco 8. Iterar sobre las filas del lienzo utilizando la estructura de control repetitiva for: <p>Iniciar en la posición fila = 0</p> <p>Incrementar fila en altoRect + espacioEntreRect en cada iteración hasta que fila sea menor que altoLienzo</p> 9. Dentro del bucle de filas, iterar sobre las columnas utilizando otra estructura for: <p>Iniciar en la posición col = 0</p> <p>Incrementar col en anchoRect + espacioEntreRect en cada iteración hasta que col sea menor que anchoLienzo</p> 10. Dentro de cada iteración de filas y columnas: <p>Calcular la posición del rectángulo en (col, fila)</p> <p>Dibujar un rectángulo en la posición calculada utilizando las dimensiones anchoRect y altoRect</p> <p>Establecer el color del rectángulo en negro (fill(0)) y sin contorno (noStroke())</p>



Punto 21:

Ejercicio 21: Utilizando la estructura de control repetitiva while() dibuje la siguiente imagen utilizando líneas que forman escalones y sobre cada borde de escalón se dibuje un punto de color rojo. El tamaño del lienzo es size(500,500). La estructura while() se ejecuta dentro de la función setup(). La condición es que solo se dibuje dentro del lienzo. Utilice variables que puedan ayudar a la construcción del dibujo, por ej: x, y, anchoEscalon, altoEscalon, etc.

Desarrollo del punto

Análisis

Datos de Entrada:

Tamaño del lienzo: Ancho y alto de 500x500 pixeles.

Dimensiones de los escalones:

anchoEscalon: 60 pixeles

altoEscalon: 30 pixeles

Datos de Salida:

Una imagen en forma de escalones dibujados dentro del lienzo, donde cada borde de escalón tiene un punto rojo.

Proceso:

Utilizaremos la estructura de control repetitiva `while()` para dibujar líneas que formen escalones y puntos rojos en cada borde de escalón.

El bucle `while()` se ejecutará dentro de la función `setup()` para asegurar que el dibujo se realice al iniciar el programa y solo dentro del área del lienzo.

Utilizaremos variables como `x` y `y` para rastrear la posición actual mientras se dibujan los escalones y los puntos.

Dibujaremos líneas horizontales para formar escalones de forma ascendente en el eje Y.

En cada borde de escalón, dibujaremos un punto rojo utilizando la función `point()`.

Entidad que Resuelve el Problema: Dibujador de Escalones
Variables <code>x</code> : Real <code>y</code> : Real <code>anchoEscalon</code> : Real <code>altoEscalon</code> : Real <code>dibujarPunto</code> : Booleano
NOMBRE ALGORITMO: Dibujar_Escalones Proceso del algoritmo: <ol style="list-style-type: none"> Configuración del Lienzo: Se utiliza la función <code>size()</code> para crear un lienzo de 500x500 píxeles. Se inicializan las variables <code>x</code>, <code>y</code>, <code>anchoEscalon</code> y <code>altoEscalon</code> con valores adecuados para la construcción del dibujo. Dibujo de Escalones con Puntos Rojos: Utilizando una estructura <code>while()</code>, se dibujan escalones de líneas horizontales en el lienzo. En cada iteración, se verifica si se debe dibujar un punto rojo en el borde del escalón (<code>dibujarPunto</code> es verdadero).

Se controla la posición y el tamaño de cada escalón con las variables x, y, anchoEscalon y altoEscalon.

Se actualizan las variables x e y para pasar al siguiente escalón.

3. Condición de Finalización:

La estructura while() se ejecuta mientras las coordenadas y no superen la altura del lienzo (y <= height).

ejercicio 21

```
1 PVector puntoA, puntoB, puntoC, puntoD;  
2 int distLinea;  
3  
4 void setup() {  
5   println("Ejercicio 21:");  
6   size(500, 500);  
7   distLinea = 60;  
8   puntoA = new PVector(0, distLinea);  
9  
10  while (puntoA.y < height) {  
11    actualizarCoordenadasA();  
12    dibujarEscalon();  
13  }  
14 }  
15  
16 void dibujarEscalon() {  
17   stroke(#11F0DF);  
18   puntoB = new PVector(puntoA.x + distLinea, puntoA.y);  
19   line(puntoA.x, puntoA.y, puntoB.x, puntoB.y);
```

Se ha guardado el sketch.

Ejercicio 21:
NullPointerException

Consola Errores

Punto 22:

Ejercicio 22: Utilizando la estructura de control repetitiva do-while. Replique la siguiente imagen La imagen debe ser construida desde la función setup(). Defina el tamaño del lienzo en size(600,600), verticalmente se divide el lienzo en franjas de igual medida, se deben dibujar los círculos sobre cada línea de por medio es decir en la línea 1 se dibujan círculos con distanciamiento, en la línea 2 no se dibuja y así sucesivamente. Las líneas tienen un color fijo, los círculos asumen colores aleatorios.

Desarrollo del punto

Análisis

Datos de Entrada:

anchoLienzo: Real

altoLienzo: Real

Datos de Salida:

Visualización de círculos dibujados en el lienzo según el patrón especificado.

Proceso del Algoritmo:

Establecer el tamaño del lienzo utilizando size(anchoLienzo, altoLienzo).

Dividir verticalmente el lienzo en franjas de igual medida.

Utilizar la estructura do-while para dibujar los círculos siguiendo un patrón de franjas alternas.

En cada iteración, dibujar círculos sobre las franjas impares y omitir las franjas pares.

Asignar colores aleatorios a los círculos.

ENTIDAD QUE RESUELVE EL PROBLEMA: Dibujador de Círculos en Franjas alternas
VARIABLES: anchoLienzo: Real. altoLienzo: Real. numFranjas: Entero. altoFanja: Real. y: Real. dibujarCirculos: Booleano
NOMBRE ALGORITMO: Dibujar_Circulos_Franjas_Alternas PROCESO DEL ALGORITMO: <ol style="list-style-type: none"> Definir las dimensiones del lienzo utilizando size(anchoLienzo, altoLienzo). Inicializar el fondo del lienzo con background(255) para un fondo blanco.

3. Calcular el altoFranja dividiendo altoLienzo por numFranjas.
4. Inicializar la posición vertical y en 0.
5. Utilizar la estructura do-while para iterar sobre las franjas del lienzo.
6. En cada iteración:

Verificar si se debe dibujar círculos en la franja actual (dibujarCirculos es verdadero).

Si se deben dibujar círculos:

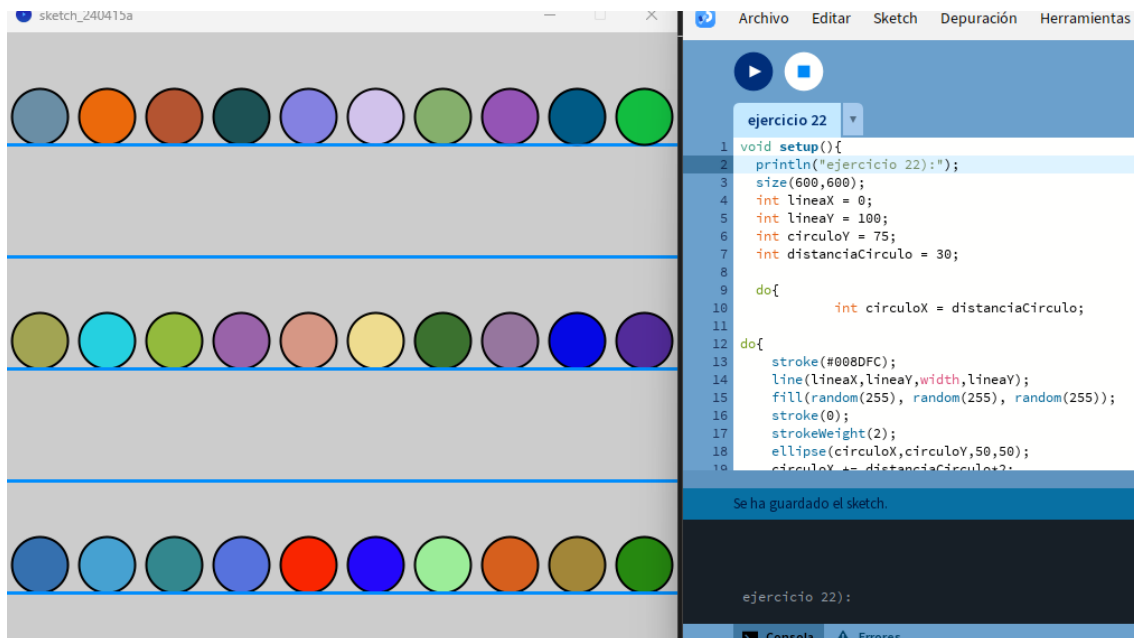
Iterar horizontalmente sobre la franja y dibujar círculos en posiciones específicas.

Asignar un color aleatorio a cada círculo utilizando fill(random(255), random(255), random(255)).

Actualizar la posición vertical y para la siguiente franja.

Alternar la variable dibujarCirculos para cambiar entre franjas impares y pares.

7. Continuar el bucle do-while hasta que se haya completado el dibujo de todas las franjas verticales del lienzo.



Conclusión



Párrafos de las conclusiones

Fuentes bibliográficas

Se deben enunciar las fuentes (apuntes de la materia, páginas web, videos de youtube, libro (nombre, autores, año), etc)

El unico que me lei fue de Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction" (Segunda Edición 2015) por Daniel Shiffman en español.



 <p>TUDI Videojuegos Fundamentos de Programación Orientada a Objetos</p>	<p>FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS FACULTAD DE INGENIERÍA Universidad Nacional de Jujuy Trabajo Practico N° / Actividad</p>	
---	---	--