

Informe de Laboratorio 03

Tema: Arreglos de Objetos

Nota

Estudiante	Escuela	Asignatura
Christian Mestas Zegarra cmestasz@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Fundamentos de la Programación 2 Semestre: II Código: 1701213

Laboratorio	Tema	Duración
03	Arreglos de Objetos	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 18 Setiembre 2023	Al 25 Setiembre 2023

1. Tarea

- **Actividad 1:** Analice, complete y pruebe el Código de la clase DemoBatalla.
- **Actividad 2:** Solucionar la Actividad 4 de la Práctica 1 pero usando arreglo de objetos.
- **Actividad 3:** Solucionar la Actividad 5 de la Práctica 1 pero usando arreglos de objetos.

2. Equipos, materiales y temas utilizados

- Sistema Operativo Microsoft Windows 10 Pro 64 bits
- Visual Studio Code 1.82.2
- Java Development Kit 17.0.1
- Git 2.41.0.windows.1
- Windows PowerShell 5.1.19041.3031
- Cuenta en GitHub con el correo institucional.
- Arreglos de objetos.

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- <https://github.com/cmestasz/fp2-23b.git>
- URL para el laboratorio 03 en el Repositorio GitHub.
- <https://github.com/cmestasz/fp2-23b/tree/main/fase01/lab03>

4. Actividades con el repositorio GitHub

Creando plantillas

```
$ mkdir lab03
$ cd lab03
$ code Nave.java
$ code DemoBatalla.java
$ code Soldado.java
$ code Actividad2.java
$ code Actividad3.java
```

Primer Commit / Plantillas

```
$ git add .
$ git commit -m "Plantillas de archivos necesarios para resolver las actividades del
    laboratorio"
$ git push
```

Commit

Plantillas de archivos necesarios para resolver las actividades del L...
laboratorio

17 min
cmestaz committed 1 hour ago

Showing 5 changed files with 133 additions and 0 deletions.

Filter changed files

- head/17a00
 - Actividad2.java
 - Actividad3.java
 - DemoBatalla.java
 - Nave.java
 - Soldado.java

Files changed:

- 0 Fases/Lab01/Actividad2.java


```
1 // Laboratorio Nro 1 - Actividad 2
2 // Autor: Christian Mestas
3
4 public class Actividad2 {
5     public static void main(String[] args) {
6
7     }
8 }
```
- 0 Fases/Lab01/Actividad3.java


```
1 // Laboratorio Nro 1 - Actividad 3
2 // Autor: Christian Mestas
3
4 public class Actividad3 {
5     public static void main(String[] args) {
6
7     }
8 }
```
- 0 Fases/Lab01/DemoBatalla.java


```
1 // Laboratorio Nro 1 - Actividad 1
2 // Autor: Christian Mestas
3 // Solución: Naveos, Naves, clases DemoBatalla y Nave
4
5 import java.util.*;
6
7 public class DemoBatalla {
8     public static void main(String[] args) {
9         Nave[] naves = new Nave[10];
10        Scanner sc = new Scanner(System.in);
11        String nombre;
12        int fil, punt;
13        boolean est;
14        for (int i = 0; i < naves.length; i++) {
15            System.out.println("Nave " + (i + 1));
16            System.out.print("Nombre: ");
```

Primer Commit.

Actualizando Nave.java y DemoBatalla.java

```
$ code Nave.java
$ code DemoBatalla.java
```

Segundo - Sexto Commit / Nave.java y DemoBatalla.java

```
$ git add .
$ git commit -m "Metodo mostrarNaves() y metodo toString() de la clase Nave"
$ code DemoBatalla.java
$ git add DemoBatalla.java
$ git commit -m "Metodo mostrarPorNombre()"
$ code DemoBatalla.java
$ git add DemoBatalla.java
$ git commit -m "Metodo mostrarPorPuntos()"
$ code DemoBatalla.java
$ git add DemoBatalla.java
$ git commit -m "Metodo mostrarMayorPuntos()"
$ code DemoBatalla.java
$ git add DemoBatalla.java
$ git commit -m "Metodo desordenar()"
$ git push
```

Commit

Metodo mostrarNaves() y metodo toString() de la clase Nave

main

cmestasz committed 45 minutes ago

1 parent 7c38db commit e5680e3

Showing 2 changed files with 10 additions and 1 deletion.

Filter changed files

ase07/ase03

Demotalla.java

Nave.java

```

1 @@ -5,7 +5,7 @@
2
3 public class Demotalla {
4     public static void main(String[] args) {
5         Nave[] naves = new Nave[10];
6         Scanner sc = new Scanner(System.in);
7         String nomb, col;
8         int f12, punt;
9
10        @@ -10,6 +10,8 @@ public static void main(String[] args) {
11
12        // Método para mostrar todas las naves
13        public static void mostrarNaves(Nave[] flota) {
14
15        }
16        // Método para mostrar todas las naves de un nombre que se pide por teclado
17        @@ -12,6 +14,8 @@ public static void mostrarNaves(Nave[] flota) {
18
19        // Método que devuelve la Nave con mayor número de Puntos
20        public static Nave mostrarMayorPuntos(Nave[] flota) {
21
22        }
23        // Crear un metodo que devuelva un nuevo arreglo de objetos con todos los
24        // objetos previamente ingresados
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

```

```

1 @@ -1,1 +1,5 @@
2
3 public class Nave {
4     private String nombre;
5     private int file;
6
7     @@ -11,1 +11,9 @@ public int getpuntos() {
8
9     return puntos;
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

```

Segundo Commit.

Commit

Metodo mostrarPorNombre()

main

cmestasz committed 42 minutes ago

1 parent e5680e3 commit 9242483

Showing 1 changed file with 8 additions and 0 deletions.

ase07/ase03

Demotalla.java

```

1 @@ -45,6 +45,14 @@ public static void mostrarNaves(Nave[] flota) {
2
3 // Método para mostrar todas las naves de un nombre que se pide por teclado
4 public static void mostrarPorNombre(Nave[] flota) {
5
6
7
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

```

```

1 @@ -45,6 +45,14 @@ public static void mostrarNaves(Nave[] flota) {
2
3 // Método para mostrar todas las naves de un nombre que se pide por teclado
4 public static void mostrarPorNombre(Nave[] flota) {
5
6     Scanner sc = new Scanner(System.in);
7     System.out.println("Ingresa el nombre a buscar:");
8     String nombre = sc.nextLine();
9     System.out.println("Naves con el nombre " + nombre + ":");
10     for (Nave nave : flota) {
11         if (nave.getNombre().equals(nombre)) {
12             System.out.println(nave.toString());
13         }
14     }
15
16 // Método para mostrar todas las naves con un número de puntos inferior o igual
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

```

Tercer Commit.

Commit

Metodo mostrarPorPuntos()
main
cmestaz committed 40 minutes ago
1 parent 8242483 commit 8668f45

Showing 1 changed file with 9 additions and 0 deletions.
Split Unified

```

1 @@ -58,6 +58,15 @@ public static void mostrarPorNombre(Nave[] flota) {
2 // Método para mostrar todas las naves con un número de puntos inferior o igual
3 // al número de puntos que se pide por teclado
4 public static void mostrarPorPuntos(Nave[] flota) {
5
6 // Método para mostrar todas las naves con un número de puntos inferior o igual
7 // al número de puntos que se pide por teclado
8 public static void mostrarPorPuntos(Nave[] flota) {
9
10 Scanner sc = new Scanner(System.in);
11 System.out.println("Ingrese la cantidad de puntos maximos:");
12 int puntos = sc.nextInt();
13 sc.nextLine();
14 System.out.println("Naves con " + puntos + " puntos como maximo:");
15 for (Nave nave : flota) {
16 if (nave.getPuntos() <= puntos)
17 System.out.println(nave.toString());
18 }
19 }
20 // Método que devuelve la nave con mayor número de Puntos
21
22 // Método que devuelve la nave con mayor número de Puntos

```

Cuarto Commit.

Commit

Metodo mostrarMayorPuntos()
main
cmestaz committed 35 minutes ago
1 parent 8668f45 commit 8242483

Showing 1 changed file with 12 additions and 4 deletions.
Split Unified

```

1 @@ -4,11 +4,11 @@
2
3 public class DemoBatalla {
4 public static void main(String[] args) {
5 Nave[] misNaves = new Nave[5];
6 Scanner sc = new Scanner(System.in);
7 String nombre;
8 int f1, punt;
9 boolean est;
10 for (int i = 0; i < misNaves.length; i++) {
11 System.out.println("Nave " + (i + 1));
12 System.out.print("Nombre: ");
13 nombre = sc.next();
14 System.out.println("Fila ");
15
16 @@ -34,11 +34,14 @@ public static void main(String[] args) {
17 mostrarNaves(misNaves);
18 mostrarPorNombre(misNaves);
19 mostrarPorPuntos(misNaves);
20 System.out.println("Nave con mayor número de puntos: " + mostrarMayorPuntos(misNaves));
21 }
22 // Método para mostrar todas las naves
23 public static void mostrarNaves(Nave[] flota) {
24 for (Nave nave : flota)
25 System.out.println(nave.toString());
26 }
27 // Método para mostrar todas las naves de un nombre que se pide por teclado
28
29 // Método para mostrar todas las naves de un nombre que se pide por teclado
30
31 @@ -53,6 +53,7 @@ public static void mostrarPorNombre(Nave[] flota) {
32 if (nave.getNombre().equals(nombre))
33 System.out.println(nave.toString());
34 }
35 }
36 // Método para mostrar todas las naves con un número de puntos inferior o igual
37
38 @@ -47,11 +47,17 @@ public static void mostrarPorPuntos(Nave[] flota) {
39 if (nave.getPuntos() <= puntos)
40 System.out.println(nave.toString());
41 }
42 }
43
44 if (nave.getPuntos() <= puntos)
45 System.out.println(nave.toString());
46 }
47 System.out.println();
48 }

```

Quinto Commit.

Metodo desordenar()

17 main

cmestaz committed 20 minutes ago

Showing 1 changed file with 20 additions and 3 deletions.

1

2

3

4

5

6

7

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

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

Sexto Commit.

Actualizando Soldado.java

```
$ code Soldado.java
```

Septimo Commit / Soldado.java

```
$ git add Soldado.java
$ git commit -m "Clase Soldado"
$ git push
```

Commit

Clase Soldado

main

cmestasz committed 17 minutes ago

1 parent 1b47e8a commit e406b59

Browse files

Showing 1 changed file with 27 additions and 1 deletion.

Split Unified

28

file01/1a003/Soldado.java

...

1

public class Soldado {

2

-

3

}

1

public class Soldado {

2

+ private String nombre;

3

+ private int vida;

4

+

5

+ public Soldado(String nombre, int vida) {

6

+ setNombre(nombre);

7

+ setVida(vida);

8

+ }

9

+

10

+ public void setNombre(String nombre) {

11

+ this.nombre = nombre;

12

+ }

13

+

14

+ public void setVida(int vida) {

15

+ this.vida = vida;

16

+ }

17

+

18

+ public String getNombre() {

19

+ return nombre;

20

+ }

21

+

22

+ public int getVida() {

23

+ return vida;

24

+ }

25

+

26

+ public String toString() {

27

+ return "Nombre: " + nombre + ", Vida: " + vida;

28

+ }

29

}

Septimo Commit.

Actualizando Actividad2.java

```
$ code Actividad2.java
```

Octavo Commit / Soldado.java

```
$ git add Actividad2.java
$ git commit -m "Actividad 2"
$ git push
```

Commit

Actividad 2

main

cmestasz committed 12 minutes ago

1 parent e406b59 commit 2b43345

Browse files

Showing 1 changed file with 14 additions and 1 deletion.

Split Unified

15

file01/1a003/Actividad2.java

...

1

// Laboratorio Nro 3 - Actividad 2

2

// Autor: Christian Mestas

3

-

4

public class Actividad2 {

5

public static void main(String[] args) {

6

-

7

}

8

}

1

// Laboratorio Nro 3 - Actividad 2

2

// Autor: Christian Mestas

3

-

4

+ import java.util.Scanner;

5

+

6

+ public class Actividad2 {

7

+ public static void main(String[] args) {

8

+ Scanner sc = new Scanner(System.in);

9

+ Soldado[] soldados = new Soldado[5];

10

+ for (int i = 0; i < soldados.length; i++) {

11

+ String nombre = sc.nextLine();

12

+ int vida = sc.nextInt();

13

+ sc.nextLine();

14

+ soldados[i] = new Soldado(nombre, vida);

15

+ }

16

+ for (int i = 0; i < soldados.length; i++) {

17

+ System.out.println("Soldado " + (i + 1) + ":");

18

+ System.out.println(soldados[i]);

19

+ }

20

}

21

}

Octavo Commit.

Actualizando Actividad3.java

```
$ code Actividad3.java
```

Noveno Commit / Actividad3.java

```
$ git add Actividad3.java
$ git commit -m "Actividad 3"
$ git push
```

Commit

Actividad 3

main

cmestasz committed 5 minutes ago

1 parent 2b43345 commit 7883586

Showing 1 changed file with 30 additions and 1 deletion.

Split Unified

```

1 // Laboratorio Nro 3 - Actividad 3
2 // Autor: Christian Mestas
3
4
5
6
7
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
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

```

Noveno Commit.

5. Código desarrollado

Nave.java

```
1 public class Nave {
2     private String nombre;
3     private int fila;
4     private String columna;
5     private boolean estado;
6     private int puntos;
7
8     // Metodos mutadores
9     public void setNombre(String n) {
10         nombre = n;
11     }
12
13     public void setFila(int f) {
14         fila = f;
15     }
16
17     public void setColumna(String c) {
18         columna = c;
19     }
20
21     public void setEstado(boolean e) {
22         estado = e;
23     }
24
25     public void setPuntos(int p) {
26         puntos = p;
27     }
28
29     // Metodos accesoires
30     public String getNombre() {
31         return nombre;
32     }
33
34
35     public int getFila() {
36         return fila;
37     }
38
39
40     public String getColumna() {
41         return columna;
42     }
43
44
45     public boolean getEstado() {
46         return estado;
47     }
48
49
50     public int getPuntos() {
51         return puntos;
52 }
```

```
53     }  
54  
55     public String toString() {  
56         return (nombre + ": (" + fila + ", " + columna + "). Estado: " + estado + ". Puntos:  
57             " + puntos);  
58     }  
59 }
```

- Clase que guarda nombre, fila, columna, estado y puntos de la nave.
- Posee tanto setters como getters para todos los atributos.
- Posee el metodo toString() para poder imprimir el objeto.

DemoBatalla.java

```
1 // Laboratorio Nro 3 - Actividad 1
2 // Autor: Christian Mestas
3 // Colaboro: Marco Aedo, clases DemoBatalla y Nave
4
5 import java.util.*;
6
7 public class DemoBatalla {
8     public static void main(String[] args) {
9         Nave[] misNaves = new Nave[10];
10        Scanner sc = new Scanner(System.in);
11        String nomb, col;
12        int fil, punt;
13        boolean est;
14        for (int i = 0; i < misNaves.length; i++) {
15            System.out.println("Nave " + (i + 1));
16            System.out.print("Nombre: ");
17            nomb = sc.nextLine();
18            System.out.print("Fila: ");
19            fil = sc.nextInt();
20            sc.nextLine();
21            System.out.print("Columna: ");
22            col = sc.nextLine();
23            System.out.print("Estado: ");
24            est = sc.nextBoolean();
25            sc.nextLine();
26            System.out.print("Puntos: ");
27            punt = sc.nextInt();
28            sc.nextLine();
29            misNaves[i] = new Nave(); // Se crea un objeto Nave y se asigna su referencia a
30                                   // misNaves
31            misNaves[i].setNombre(nomb);
32            misNaves[i].setFila(fil);
33            misNaves[i].setColumna(col);
34            misNaves[i].setEstado(est);
35            misNaves[i].setPuntos(punt);
36            System.out.println();
37        }
38        System.out.println("Naves creadas:");
39        mostrarNaves(misNaves);
40        mostrarPorNombre(misNaves);
41        mostrarPorPuntos(misNaves);
42        System.out.println("Nave con mayor numero de puntos: " +
43            mostrarMayorPuntos(misNaves));
44        System.out.println();
45        System.out.println("Naves desordenadas: ");
46        mostrarNaves(desordenar(misNaves));
47    }
48
49    // Metodo para mostrar todas las naves
50    public static void mostrarNaves(Nave[] flota) {
51        for (Nave nave : flota)
52            System.out.println(nave.toString());
53    }
```

```
54 // Metodo para mostrar todas las naves de un nombre que se pide por teclado
55 public static void mostrarPorNombre(Nave[] flota) {
56     Scanner sc = new Scanner(System.in);
57     System.out.println("Ingrese el nombre a buscar:");
58     String nombre = sc.nextLine();
59     System.out.println("Naves con el nombre " + nombre + ":");
60     for (Nave nave : flota) {
61         if (nave.getNombre().equals(nombre))
62             System.out.println(nave.toString());
63     }
64     System.out.println();
65 }
66
67 // Metodo para mostrar todas las naves con un numero de puntos inferior o igual
68 // al numero de puntos que se pide por teclado
69 public static void mostrarPorPuntos(Nave[] flota) {
70     Scanner sc = new Scanner(System.in);
71     System.out.println("Ingrese la cantidad de puntos maximos:");
72     int puntos = sc.nextInt();
73     sc.nextLine();
74     System.out.println("Naves con " + puntos + " puntos como maximo:");
75     for (Nave nave : flota) {
76         if (nave.getPuntos() <= puntos)
77             System.out.println(nave.toString());
78     }
79     System.out.println();
80 }
81
82 // Metodo que devuelve la Nave con mayor numero de Puntos
83 public static Nave mostrarMayorPuntos(Nave[] flota) {
84     int maxIdx = 0;
85     for (int i = 0; i < flota.length; i++) {
86         if (flota[i].getPuntos() > flota[maxIdx].getPuntos())
87             maxIdx = i;
88     }
89     return flota[maxIdx];
90 }
91
92 // Crear un metodo que devuelva un nuevo arreglo de objetos con todos los
93 // objetos previamente ingresados
94 // pero aleatoriamente desordenados
95 public static Nave[] desordenar(Nave[] flota) {
96     Nave[] nuevaFlota = new Nave[flota.length];
97     Random r = new Random();
98     System.arraycopy(flota, 0, nuevaFlota, 0, flota.length);
99     for (int idx = 0; idx < nuevaFlota.length; idx++) {
100         int nIdx = r.nextInt(nuevaFlota.length);
101         Nave t = nuevaFlota[idx];
102         nuevaFlota[idx] = nuevaFlota[nIdx];
103         nuevaFlota[nIdx] = t;
104     }
105     return nuevaFlota;
106 }
107 }
```

- Se instancian 10 objetos de la clase nave en el arreglo misNaves, todos los datos son llenados por

el usuario.

- `mostrarNaves()` imprime todas las naves usando su método `toString()`.
- `mostrarPorNombre()` imprime todas las naves con el nombre que el usuario ingrese.
- `mostrarPorPuntos()` imprime todas las naves con menor o igual numero de puntos a lo que el usuario ingrese.
- `mostrarMayorPuntos()` retorna la nave que tiene la mayor cantidad de puntos, que luego es impresa.
- `desordenar()` retorna un nuevo arreglo de naves con las naves desordenadas al azar, que luego es impreso con el método `mostrarNaves()`.

Soldado.java

```
1 public class Soldado {
2     private String nombre;
3     private int vida;
4
5     public Soldado(String nombre, int vida) {
6         setNombre(nombre);
7         setVida(vida);
8     }
9
10    public void setNombre(String nombre) {
11        this.nombre = nombre;
12    }
13
14    public void setVida(int vida) {
15        this.vida = vida;
16    }
17
18    public String getNombre() {
19        return nombre;
20    }
21
22    public int getVida() {
23        return vida;
24    }
25
26    public String toString() {
27        return ("Nombre: " + nombre + ". Vida: " + vida);
28    }
29 }
```

- Clase que guarda nombre y vida del soldado.
- Posee tanto setters como getters para los atributos.
- Posee el metodo toString() para poder imprimir el objeto.

Actividad2.java

```
1 // Laboratorio Nro 3 - Actividad 2
2 // Autor: Christian Mestas
3
4 import java.util.Scanner;
5
6 public class Actividad2 {
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9         Soldado[] soldados = new Soldado[5];
10        for (int i = 0; i < soldados.length; i++) {
11            String nombre = sc.nextLine();
12            int vida = sc.nextInt();
13            sc.nextLine();
14            soldados[i] = new Soldado(nombre, vida);
15        }
16        for (int i = 0; i < soldados.length; i++) {
17            System.out.println("Soldado " + (i + 1) + ":");
18            System.out.println(soldados[i]);
19        }
20    }
21 }
```

- Los 5 nombres y sus respectivas vidas son leídas, para luego instanciar objetos de la clase soldado con sus respectivos atributos.
- Luego los atributos son impresos en el orden en el que se ingresaron.

Actividad3.java

```
1 // Laboratorio Nro 3 - Actividad 3
2 // Autor: Christian Mestas
3
4 import java.util.Random;
5
6 public class Actividad3 {
7     public static void main(String[] args) {
8         Soldado[] ejercito1 = inicializarEjercito();
9         Soldado[] ejercito2 = inicializarEjercito();
10        mostrarEjercito(ejercito1, 1);
11        mostrarEjercito(ejercito2, 2);
12        mostrarGanador(ejercito1, ejercito2);
13    }
14
15    public static Soldado[] inicializarEjercito() {
16        Random random = new Random();
17        Soldado[] ejercito = new Soldado[random.nextInt(5) + 1];
18        for (int i = 0; i < ejercito.length; i++)
19            ejercito[i] = new Soldado("Soldado" + i, 0);
20        return ejercito;
21    }
22
23    public static void mostrarEjercito(Soldado[] ejercito, int numero) {
24        System.out.println("Ejercito " + numero + ":");
25        for (int i = 0; i < ejercito.length; i++)
26            System.out.println("Soldado " + (i + 1) + ": " + ejercito[i].getNombre());
27    }
28
29    public static void mostrarGanador(Soldado[] ejercito1, Soldado[] ejercito2) {
30        if (ejercito1.length > ejercito2.length)
31            System.out.println("Gana el ejercito 1!");
32        else if (ejercito1.length < ejercito2.length)
33            System.out.println("Gana el ejercito 2!");
34        else
35            System.out.println("Hubo un empate!");
36    }
37 }
```

- Se crean 2 ejércitos con el método `inicializarEjercito()`, que crea un arreglo de soldados de tamaño al azar entre 1 y 5.
- `mostrarEjercito()` imprime una lista de los soldados de cada ejército.
- `mostrarGanador()` compara los tamaños de los ejércitos y declara al ganador como el ejército mas grande.

6. Ejecución del código

DemoBatalla.java

```
1 Nave 1
2 Nombre: naveuno
3 Fila: 5
4 Columna: G
5 Estado: true
6 Puntos: 6
7
8 Nave 2
9 Nombre: navedos
10 Fila: 2
11 Columna: A
12 Estado: false
13 Puntos: 7
14
15 Nave 3
16 Nombre: navetres
17 Fila: 8
18 Columna: J
19 Estado: true
20 Puntos: 11
21
22 Nave 4
23 Nombre: navecuatro
24 Fila: 8
25 Columna: D
26 Estado: true
27 Puntos: 9
28
29 Nave 5
30 Nombre: navecinco
31 Fila: 6
32 Columna: F
33 Estado: false
34 Puntos: 11
35
36 Nave 6
37 Nombre: naveseis
38 Fila: 2
39 Columna: B
40 Estado: false
41 Puntos: 7
42
43 Nave 7
44 Nombre: navesiete
45 Fila: 3
46 Columna: F
47 Estado: true
48 Puntos: 2
49
50 Nave 8
51 Nombre: naveocho
52 Fila: 4
```

```
53 Columna: A
54 Estado: true
55 Puntos: 10
56
57 Nave 9
58 Nombre: navenueve
59 Fila: 3
60 Columna: I
61 Estado: false
62 Puntos: 13
63
64 Nave 10
65 Nombre: navediez
66 Fila: 2
67 Columna: E
68 Estado: true
69 Puntos: 7
70
71 Naves creadas:
72 naveuno: (5, G). Estado: true. Puntos: 6
73 navedos: (2, A). Estado: false. Puntos: 7
74 navetres: (8, J). Estado: true. Puntos: 11
75 navecuatro: (8, D). Estado: true. Puntos: 9
76 navecinco: (6, F). Estado: false. Puntos: 11
77 naveseis: (2, B). Estado: false. Puntos: 7
78 navesiete: (3, F). Estado: true. Puntos: 2
79 naveocho: (4, A). Estado: true. Puntos: 10
80 navenueve: (3, I). Estado: false. Puntos: 13
81 navediez: (2, E). Estado: true. Puntos: 7
82
83 Ingrese el nombre a buscar:
84 navecinco
85 Naves con el nombre navecinco:
86 navecinco: (6, F). Estado: false. Puntos: 11
87
88 Ingrese la cantidad de puntos maximos: 8
89 Naves con 8 puntos como maximo:
90 naveuno: (5, G). Estado: true. Puntos: 6
91 navedos: (2, A). Estado: false. Puntos: 7
92 naveseis: (2, B). Estado: false. Puntos: 7
93 navesiete: (3, F). Estado: true. Puntos: 2
94 navediez: (2, E). Estado: true. Puntos: 7
95
96 Nave con mayor numero de puntos: navenueve: (3, I). Estado: false. Puntos: 13
97
98 Naves desordenadas:
99 navecinco: (6, F). Estado: false. Puntos: 11
100 naveseis: (2, B). Estado: false. Puntos: 7
101 naveocho: (4, A). Estado: true. Puntos: 10
102 navesiete: (3, F). Estado: true. Puntos: 2
103 navediez: (2, E). Estado: true. Puntos: 7
104 navetres: (8, J). Estado: true. Puntos: 11
105 naveuno: (5, G). Estado: true. Puntos: 6
106 navenueve: (3, I). Estado: false. Puntos: 13
107 navedos: (2, A). Estado: false. Puntos: 7
108 navecuatro: (8, D). Estado: true. Puntos: 9
```

Actividad2.java

```
1 Luis Pancho
2 15
3 Marco Marco
4 20
5 Jorge Mandela
6 16
7 Jose Rufs
8 18
9 Esqueche Almino
10 21
11 Soldado 1:
12 Nombre: Luis Pancho. Vida: 15
13 Soldado 2:
14 Nombre: Marco Marco. Vida: 20
15 Soldado 3:
16 Nombre: Jorge Mandela. Vida: 16
17 Soldado 4:
18 Nombre: Jose Rufs. Vida: 18
19 Soldado 5:
20 Nombre: Esqueche Almino. Vida: 21
```

Actividad3.java

```
1 Ejercito 1:
2 Soldado 1: Soldado0
3 Soldado 2: Soldado1
4 Soldado 3: Soldado2
5 Soldado 4: Soldado3
6 Soldado 5: Soldado4
7 Ejercito 2:
8 Soldado 1: Soldado0
9 Gana el ejercito 1!
```

7. Estructura de laboratorio 03

- El contenido que se entrega en este laboratorio es el siguiente:

```
lab03/  
|--- Nave.java  
|--- DemoBatalla.java  
|--- Soldado.java  
|--- Actividad2.java  
|--- Actividad3.java  
|--- ejec01.bash  
|--- ejec02.bash  
|--- ejec03.bash  
|--- Informe.tex  
|--- Informe.pdf  
|--- img  
    |--- logo_abet.png  
    |--- logo_episunsa.png  
    |--- logo_unsa.jpg  
    |--- commit01.jpg  
    |--- commit02.jpg  
    |--- commit03.jpg  
    |--- commit04.jpg  
    |--- commit05.jpg  
    |--- commit06.jpg  
    |--- commit07.jpg  
    |--- commit08.jpg  
    |--- commit09.jpg
```

8. Rúbricas

8.1. Entregable Informe

Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y fácil de leer.

8.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobatoria, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Niveles de desempeño

Puntos	Nivel			
	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	3	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	1.5	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	1.5	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente estan dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	1.5	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	3	
Total		20		16.5	

9. Referencias

- Aedo, M. y Castro, E. (2021). FUNDAMENTOS DE PROGRAMACIÓN 2 - Tópicos de Programación Orientada a Objetos. Editorial UNSA.