

## Lab02

### Task 4.1:

Code:

```
main.go ×
main.go > main
1  package main
2
3  import (
4      "fmt"
5      "time"
6  )
7
8  func searchEngineA(query string, ch chan string) {
9      time.Sleep(300 * time.Millisecond)
10     result := fmt.Sprintf("Results from Engine A for '%s'", query)
11     ch <- result
12 }
13
14 func searchEngineB(query string, ch chan string) {
15     time.Sleep(200 * time.Millisecond)
16     result := fmt.Sprintf("Results from Engine B for '%s'", query)
17     ch <- result
18 }
19
20 func main() {
21     fmt.Println("=== Search Race ===")
22     query := "golang concurrency"
23
24     chA := make(chan string)
25     chB := make(chan string)
26
27     go searchEngineA(query, chA)
28     go searchEngineB(query, chB)
29
30     select {
31     case result := <-chA:
32         fmt.Println("Engine A won! (~300ms)")
33         fmt.Println(result)
34     case result := <-chB:
35         fmt.Println("Engine B won! (~200ms)")
36         fmt.Println(result)
37     }
38 }
```

Result:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SPELL CHECKER

go run "/Users/doandung/Downloads/main/main.go"
doandung@MacBook-Air-cua-Doan main % go run "/Users/doandung/Downloads/main/main.go"
=== Search Race ===
Engine B won! (~200ms)
Results from Engine B for 'golang concurrency'
doandung@MacBook-Air-cua-Doan main %
```

## Task 5.1

Code:

```
EXPLORER
> OPEN EDITORS
MAIN
  go-property-cal...
    go.mod
    main.go 1

main.go 1 X
1 package main
2
3 import (
4     "fmt"
5     "math/rand"
6     "sync"
7     "time"
8 )
9
10 type Student struct {
11     ID      int
12     StudyHours int
13 }
14
15 func student(id int, studyHours int, library chan bool, wg *sync.WaitGroup) {
16     defer wg.Done()
17
18     library <- true
19
20     fmt.Printf("Student %d entered library, will study for %d hours\n", id, studyHours)
21     // Open a file or folder to start working (%O)
22     time.Sleep(time.Duration(studyHours) * time.Second)
23
24     <-library
25
26     fmt.Printf("Student %d left library after %d hours\n", id, studyHours)
27 }
28
29 func main() {
30     fmt.Println("=== Library Simulation ===")
31     fmt.Println("Library capacity: 30 students")
32     fmt.Println("Total students today: 100")
33     fmt.Println("Simulation: 1 second = 1 hour\n")
34
35     rand.Seed(time.Now().UnixNano())
36
37     library := make(chan bool, 30)
38
39     var wg sync.WaitGroup
40
41     startTime := time.Now()
42
43     students := make([]Student, 100)
44     for i := 0; i < 100; i++ {
45         students[i] = Student{
46             ID:      i + 1,
47             StudyHours: rand.Intn(4) + 1,
48         }
49     }
```

```
49     }
50
51     for _, s := range students {
52         wg.Add(1)
53         go student(s.ID, s.StudyHours, library, &wg)
54     }
55
56     wg.Wait()
57
58     duration := time.Since(startTime)
59     // Open a file or folder to start working (%O)
60
61     fmt.Println("\n=== Simulation Complete ===")
62     fmt.Printf("Total students served: 100\n")
63     fmt.Printf("Library was open for: %.0f hours\n", duration.Seconds())
64     fmt.Printf("Peak occupancy: 30 students\n")
65 }
```

Result:

Student 16 entered library, will study for 2 hours  
Student 1 left library after 2 hours  
Student 17 entered library, will study for 3 hours  
Student 33 left library after 1 hours  
Student 18 entered library, will study for 4 hours  
Student 41 left library after 3 hours  
Student 78 left library after 3 hours  
Student 20 entered library, will study for 2 hours  
Student 25 left library after 3 hours  
Student 21 entered library, will study for 1 hours  
Student 19 entered library, will study for 1 hours  
Student 63 left library after 3 hours  
Student 22 entered library, will study for 2 hours  
Student 31 left library after 2 hours  
Student 32 left library after 2 hours  
Student 34 entered library, will study for 3 hours  
Student 24 entered library, will study for 2 hours  
Student 23 entered library, will study for 3 hours  
Student 46 left library after 1 hours  
Student 65 left library after 2 hours  
Student 90 entered library, will study for 4 hours  
Student 10 left library after 4 hours  
Student 19 left library after 1 hours  
Student 35 entered library, will study for 1 hours  
Student 3 left library after 4 hours  
Student 80 left library after 4 hours  
Student 82 entered library, will study for 2 hours  
Student 6 left library after 4 hours  
Student 38 entered library, will study for 2 hours  
Student 37 entered library, will study for 1 hours  
Student 51 entered library, will study for 1 hours  
Student 21 left library after 1 hours  
Student 36 entered library, will study for 3 hours  
Student 27 left library after 4 hours  
Student 79 left library after 4 hours  
Student 75 left library after 4 hours  
Student 26 left library after 4 hours  
Student 40 entered library, will study for 1 hours  
Student 48 left library after 2 hours  
Student 50 entered library, will study for 4 hours  
Student 57 entered library, will study for 1 hours  
Student 81 entered library, will study for 3 hours  
Student 52 entered library, will study for 4 hours  
Student 39 entered library, will study for 1 hours  
Student 8 left library after 4 hours  
Student 49 left library after 2 hours  
Student 58 entered library, will study for 2 hours  
Student 16 left library after 2 hours  
Student 53 entered library, will study for 4 hours  
Student 54 entered library, will study for 2 hours  
Student 74 left library after 3 hours  
Student 30 left library after 4 hours  
Student 59 entered library, will study for 2 hours  
Student 45 left library after 3 hours  
Student 60 entered library, will study for 2 hours  
Student 43 left library after 4 hours  
Student 44 left library after 4 hours  
Student 39 left library after 1 hours  
Student 47 left library after 3 hours  
Student 24 left library after 2 hours  
Student 37 left library after 1 hours  
Student 35 left library after 1 hours  
Student 71 entered library, will study for 4 hours  
Student 55 entered library, will study for 3 hours  
Student 61 entered library, will study for 2 hours  
Student 57 left library after 1 hours  
Student 92 entered library, will study for 1 hours

Student 61 entered library, will study for 2 hours  
Student 57 left library after 1 hours  
Student 92 entered library, will study for 1 hours  
Student 62 entered library, will study for 4 hours  
Student 95 entered library, will study for 4 hours  
Student 22 left library after 2 hours  
Student 20 left library after 2 hours  
Student 64 left library after 4 hours  
Student 94 entered library, will study for 4 hours  
Student 56 entered library, will study for 1 hours  
Student 40 left library after 1 hours  
Student 93 entered library, will study for 4 hours  
Student 86 entered library, will study for 1 hours  
Student 91 entered library, will study for 1 hours  
Student 51 left library after 1 hours  
Student 17 left library after 3 hours  
Student 72 entered library, will study for 3 hours  
Student 73 entered library, will study for 3 hours  
Student 83 entered library, will study for 2 hours  
Student 60 left library after 2 hours  
Student 89 entered library, will study for 1 hours  
Student 87 entered library, will study for 1 hours  
Student 82 left library after 2 hours  
Student 34 left library after 3 hours  
Student 38 left library after 2 hours  
Student 97 entered library, will study for 4 hours  
Student 99 entered library, will study for 3 hours  
Student 91 left library after 1 hours  
Student 85 entered library, will study for 2 hours  
Student 96 entered library, will study for 2 hours  
Student 68 entered library, will study for 1 hours  
Student 56 left library after 1 hours  
Student 98 entered library, will study for 3 hours  
Student 59 left library after 2 hours  
Student 67 entered library, will study for 2 hours  
Student 86 left library after 1 hours  
Student 69 entered library, will study for 2 hours  
Student 88 entered library, will study for 4 hours  
Student 92 left library after 1 hours  
Student 54 left library after 2 hours  
Student 84 entered library, will study for 1 hours  
Student 15 left library after 4 hours  
Student 66 entered library, will study for 3 hours  
Student 23 left library after 3 hours  
Student 70 entered library, will study for 3 hours  
Student 58 left library after 2 hours  
Student 18 left library after 4 hours  
Student 36 left library after 3 hours  
Student 81 left library after 3 hours  
Student 83 left library after 2 hours  
Student 61 left library after 2 hours  
Student 90 left library after 4 hours  
Student 89 left library after 1 hours  
Student 87 left library after 1 hours  
Student 68 left library after 1 hours  
Student 84 left library after 1 hours  
Student 73 left library after 3 hours  
Student 55 left library after 3 hours  
Student 53 left library after 4 hours  
Student 72 left library after 3 hours  
Student 52 left library after 4 hours  
Student 50 left library after 4 hours  
Student 96 left library after 2 hours  
Student 85 left library after 2 hours  
Student 67 left library after 2 hours  
Student 69 left library after 2 hours  
Student 71 left library after 4 hours  
Student 66 left library after 3 hours  
Student 70 left library after 3 hours  
Student 98 left library after 3 hours

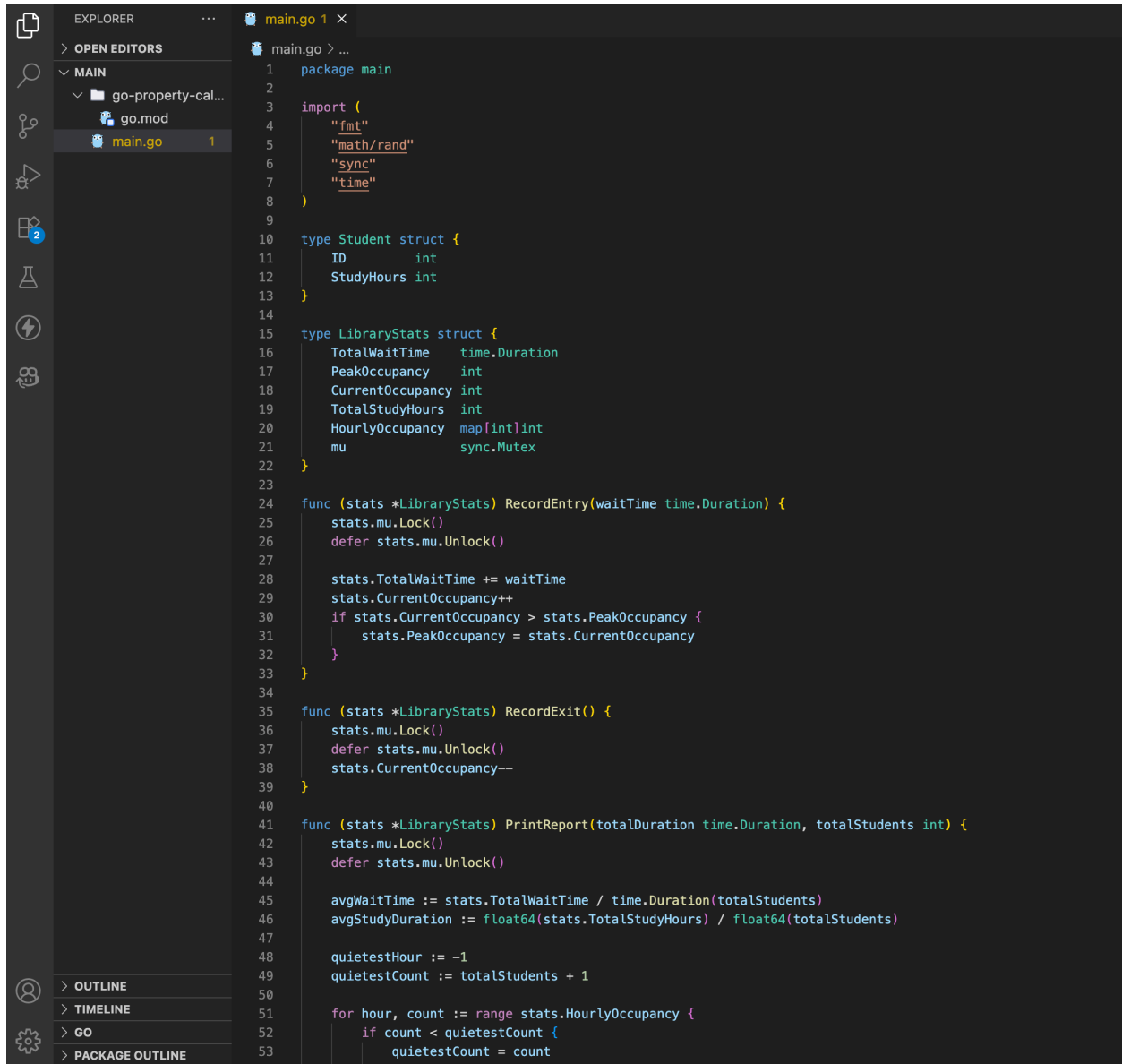
```
Student 69 left library after 2 hours  
Student 71 left library after 4 hours  
Student 66 left library after 3 hours  
Student 70 left library after 3 hours  
Student 98 left library after 3 hours  
Student 93 left library after 4 hours  
Student 62 left library after 4 hours  
Student 95 left library after 4 hours  
Student 99 left library after 3 hours  
Student 94 left library after 4 hours  
Student 88 left library after 4 hours  
Student 97 left library after 4 hours
```

```
=== Simulation Complete ===  
Total students served: 100  
Library was open for: 10 hours  
Peak occupancy: 30 students
```

```
❖ doandung@MacBook-Air-cua-Doan main %
```

Task 5.2:

Code:



The screenshot shows a Go IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a 'MAIN' folder containing 'go.mod' and 'main.go'. The code editor displays the contents of 'main.go', which is a Go program simulating a library system. The code includes imports for 'fmt', 'math/rand', 'sync', and 'time'. It defines two structs: 'Student' with fields 'ID' and 'StudyHours', and 'LibraryStats' with fields for wait times, occupancy, study hours, and a mutex. The program includes functions for recording library entries and exits, and a function to print a report. The report function calculates average wait and study durations and includes a quiet test count.

```
1 package main
2
3 import (
4     "fmt"
5     "math/rand"
6     "sync"
7     "time"
8 )
9
10 type Student struct {
11     ID        int
12     StudyHours int
13 }
14
15 type LibraryStats struct {
16     TotalWaitTime    time.Duration
17     PeakOccupancy    int
18     CurrentOccupancy int
19     TotalStudyHours  int
20     HourlyOccupancy  map[int]int
21     mu               sync.Mutex
22 }
23
24 func (stats *LibraryStats) RecordEntry(waitTime time.Duration) {
25     stats.mu.Lock()
26     defer stats.mu.Unlock()
27
28     stats.TotalWaitTime += waitTime
29     stats.CurrentOccupancy++
30     if stats.CurrentOccupancy > stats.PeakOccupancy {
31         stats.PeakOccupancy = stats.CurrentOccupancy
32     }
33 }
34
35 func (stats *LibraryStats) RecordExit() {
36     stats.mu.Lock()
37     defer stats.mu.Unlock()
38     stats.CurrentOccupancy--
39 }
40
41 func (stats *LibraryStats) PrintReport(totalDuration time.Duration, totalStudents int) {
42     stats.mu.Lock()
43     defer stats.mu.Unlock()
44
45     avgWaitTime := stats.TotalWaitTime / time.Duration(totalStudents)
46     avgStudyDuration := float64(stats.TotalStudyHours) / float64(totalStudents)
47
48     quietestHour := -1
49     quietestCount := totalStudents + 1
50
51     for hour, count := range stats.HourlyOccupancy {
52         if count < quietestCount {
53             quietestCount = count
```

EXPLORER

main.go 1 X

main.go > ...

89 func main() {

99

100 library := make(chan bool, libraryCapacity)

101 var wg sync.WaitGroup

102

103 stats := LibraryStats{

104 HourlyOccupancy: make(map[int]int),

105 }

106

107 students := make([]Student, totalStudents)

108 for i := 0; i < totalStudents; i++ {

109 studyHours := rand.Intn(4) + 1

110 students[i] = Student{

111 ID: i + 1,

112 StudyHours: studyHours,

113 }

114 stats.TotalStudyHours += studyHours

115 }

116

117 startTime := time.Now()

118

119 for \_, s := range students {

120 wg.Add(1)

121 go student(s, library, &wg, &stats)

122 }

123

124 done := make(chan struct{})

125 go func() {

126 wg.Wait()

127 close(done)

128 }()

129

130 hour := 0

131 ticker := time.NewTicker(1 \* time.Second)

132 running := true

133 for running {

134 select {

135 case <-ticker.C:

136 hour++

137 stats.mu.Lock()

138 stats.HourlyOccupancy[hour] = stats.CurrentOccupancy

139 stats.mu.Unlock()

140 case <-done:

141 ticker.Stop()

142 running = false

143 }

144 }

145

146 duration := time.Since(startTime)

147 stats.PrintReport(duration, totalStudents)

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 }

1001 }

1002 }

1003 }

1004 }

1005 }

1006 }

1007 }

1008 }

1009 }

1010 }

1011 }

1012 }

1013 }

1014 }

1015 }

1016 }

1017 }

1018 }

1019 }

1020 }

1021 }

1022 }

1023 }

1024 }

1025 }

1026 }

1027 }

1028 }

1029 }

1030 }

1031 }

1032 }

1033 }

1034 }

1035 }

1036 }

1037 }

1038 }

1039 }

1040 }

1041 }

1042 }

1043 }

1044 }

1045 }

1046 }

1047 }

1048 }

1049 }

1050 }

1051 }

1052 }

1053 }

1054 }

1055 }

1056 }

1057 }

1058 }

1059 }

1060 }

1061 }

1062 }

1063 }

1064 }

1065 }

1066 }

1067 }

1068 }

1069 }

1070 }

1071 }

1072 }

1073 }

1074 }

1075 }

1076 }

1077 }

1078 }

1079 }

1080 }

1081 }

1082 }

1083 }

1084 }

1085 }

1086 }

1087 }

1088 }

1089 }

1090 }

1091 }

1092 }

1093 }

1094 }

1095 }

1096 }

1097 }

1098 }

1099 }

1100 }

1101 }

1102 }

1103 }

1104 }

1105 }

1106 }

1107 }

1108 }

1109 }

1110 }

1111 }

1112 }

1113 }

1114 }

1115 }

1116 }

1117 }

1118 }

1119 }

1120 }

1121 }

1122 }

1123 }

1124 }

1125 }

1126 }

1127 }

1128 }

1129 }

1130 }

1131 }

1132 }

1133 }

1134 }

1135 }

1136 }

1137 }

1138 }

1139 }

1140 }

1141 }

1142 }

1143 }

1144 }

1145 }

1146 }

1147 }

1148 }

1149 }

1150 }

1151 }

1152 }

1153 }

1154 }

1155 }

1156 }

1157 }

1158 }

1159 }

1160 }

1161 }

1162 }

1163 }

1164 }

1165 }

1166 }

1167 }

1168 }

1169 }

1170 }

1171 }

1172 }

1173 }

1174 }

1175 }

1176 }

1177 }

1178 }

1179 }

1180 }

1181 }

1182 }

1183 }

1184 }

1185 }

1186 }

1187 }

1188 }

1189 }

1190 }

1191 }

1192 }

1193 }

1194 }

1195 }

1196 }

1197 }

1198 }

1199 }

1200 }

1201 }

1202 }

1203 }

1204 }

1205 }

1206 }

1207 }

1208 }

1209 }

1210 }

1211 }

1212 }

1213 }

1214 }

1215 }

1216 }

1217 }

1218 }

1219 }

1220 }

1221 }

1222 }

1223 }

1224 }

1225 }

1226 }

1227 }

1228 }

1229 }

1230 }

1231 }

1232 }

1233 }

1234 }

1235 }

1236 }

1237 }

1238 }

1239 }

1240 }

1241 }

1242 }

1243 }

1244 }

1245 }

1246 }

1247 }

1248 }

1249 }

1250 }

1251 }

1252 }

1253 }

1254 }

1255 }

1256 }

1257 }

1258 }

1259 }

1260 }

1261 }

1262 }

1263 }

1264 }

1265 }

1266 }

1267 }

1268 }

1269 }

1270 }

1271 }

1272 }

1273 }

1274 }

1275 }

1276 }

1277 }

1278 }

1279 }

1280 }

1281 }

1282 }

1283 }

1284 }

1285 }

1286 }

1287 }

1288 }

1289 }

1290 }

1291 }

1292 }

1293 }

1294 }

1295 }

1296 }

1297 }

1298 }

1299 }

1300 }

1301 }

1302 }

1303 }

1304 }

1305 }

1306 }

1307 }

1308 }

1309 }

1310 }

1311 }

1312 }

1313 }

1314 }

1315 }

1316 }

1317 }

1318 }

1319 }

1320 }

1321 }

1322 }

1323 }

1324 }

1325 }

1326 }

1327 }

1328 }

1329 }

1330 }

1331 }

1332 }

1333 }

1334 }

1335 }

1336 }

1337 }

1338 }

1339 }

1340 }

1341 }

1342 }

1343 }

1344 }

1345 }

1346 }

1347 }

1348 }

1349 }

1350 }

1351 }

1352 }

1353 }

1354 }

1355 }

1356 }

1357 }

1358 }

1359 }

1360 }

1361 }

1362 }

1363 }

1364 }

1365 }

1366 }

1367 }

1368 }

1369 }

1370 }

1371 }

1372 }

1373 }

1374 }

1375 }

1376 }

1377 }

1378 }

1379 }

1380 }

1381 }

1382 }

1383 }

1384 }

1385 }

1386 }

1387 }

1388 }

1389 }

1390 }

1391 }

1392 }

1393 }

1394 }

1395 }

1396 }

1397 }

1398 }

1399 }

1400 }

1401 }

1402 }

1403 }

1404 }

1405 }

1406 }

1407 }

1408 }

1409 }

1410 }

1411 }

1412 }

1413 }

1414 }

1415 }

1416 }

1417 }

1418 }

1419 }

1420 }

1421 }

1422 }

1423 }

1424 }

1425 }

1426 }

1427 }

1428 }

1429 }

1430 }

1431 }

1432 }

1433 }

1434 }

1435 }

1436 }

1437 }

1438 }

1439 }

1440 }

1441 }

1442 }

1443 }

1444 }

1445 }

1446 }

1447 }

1448 }

1449 }

1450 }

1451 }

1452 }

1453 }

1454 }

1455 }

1456 }

1457 }

1458 }

1459 }

1460 }

1461 }

1462 }

1463 }

1464 }

1465 }

1466 }

1467 }

1468 }

1469 }

1470 }

1471 }

1472 }

1473 }

1474 }

1475 }

1476 }

1477 }

1478 }

1479 }

1480 }

1481 }

1482 }

1483 }

1484 }

1485 }

1486 }

1487 }

1488 }

1489 }

1490 }

1491 }

1492 }

1493 }

1494 }

1495 }

1496 }

1497 }

1498 }

1499 }

1500 }

1501 }

1502 }

1503 }

1504 }

1505 }

1506 }

1507 }

1508 }

1509 }

1510 }

1511 }

1512 }

1513 }

1514 }

1515 }

1516 }

1517 }

1518 }

1519 }

1520 }

1521 }

1522 }

1523 }

1524 }

1525 }

1526 }

1527 }

1528 }

1529 }

1530 }

1531 }

1532 }

1533 }

1534 }

1535 }

1536 }

1537 }

1538 }

1539 }

1540 }

1541 }

1542 }

1543 }

15

The screenshot shows the Visual Studio Code editor with a Go file named `main.go`. The code implements a simulation of a library system. It starts by creating a `Library` struct with a channel for students and a `WaitGroup`. Then, it creates a slice of `Student` structs, each with a unique ID and a random study duration. A `done` channel is used to signal when all students have finished. A `ticker` is used to simulate the passage of time, updating the library's occupancy and wait times. The simulation runs until all students are served, and then prints a report of the results.

```
89 func main() {
90
91     library := make(chan bool, libraryCapacity)
92     var wg sync.WaitGroup
93
94     stats := LibraryStats{
95         HourlyOccupancy: make(map[int]int),
96     }
97
98     students := make([]Student, totalStudents)
99     for i := 0; i < totalStudents; i++ {
100         studyHours := rand.Intn(4) + 1
101         students[i] = Student{
102             ID: i + 1,
103             StudyHours: studyHours,
104         }
105         stats.TotalStudyHours += studyHours
106     }
107
108     startTime := time.Now()
109
110     for _, s := range students {
111         wg.Add(1)
112         go student(s, library, &wg, &stats)
113     }
114
115     done := make(chan struct{})
116     go func() {
117         wg.Wait()
118         close(done)
119     }()
120
121     hour := 0
122     ticker := time.NewTicker(1 * time.Second)
123     running := true
124     for running {
125         select {
126             case <-ticker.C:
127                 hour++
128                 stats.mu.Lock()
129                 stats.HourlyOccupancy[hour] = stats.CurrentOccupancy
130                 stats.mu.Unlock()
131             case <-done:
132                 ticker.Stop()
133                 running = false
134         }
135     }
136
137     duration := time.Since(startTime)
138     stats.PrintReport(duration, totalStudents)
139 }
```

Result:

```
=== Simulation Complete ===
Total students served: 100
Library was open for: 11 hours
Average wait time: 2.9 hours
Peak occupancy: 30 students
Quietest hour: Hour 11 (1 students)
Total student-hours: 252 hours
Average study duration: 2.5 hours per student
```



