Nguyễn Doãn Tiến Dũng-ITCSIU21173

Lab02

Task 4.1:

Code:

main.go

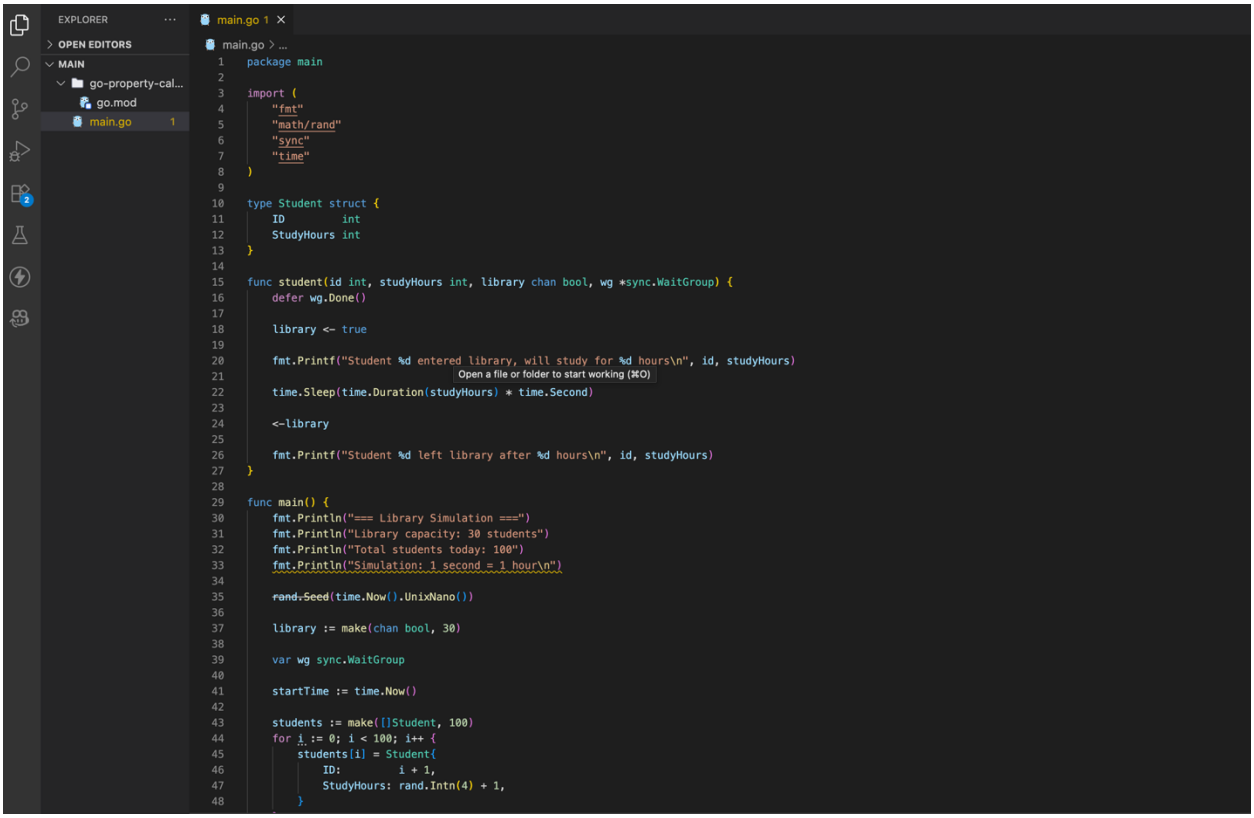
```go
package main

import (
    "fmt"
    "time"
)

func searchEngineA(query string, ch chan string) {
    time.Sleep(300 * time.Millisecond)
    result := fmt.Sprintf("Results from Engine A for '%s'", query)
    ch <- result
}

func searchEngineB(query string, ch chan string) {
    time.Sleep(200 * time.Millisecond)
    result := fmt.Sprintf("Results from Engine B for '%s'", query)
    ch <- result
}

func main() {
    fmt.Println("=== Search Race ===")
    query := "golang concurrency"

    chA := make(chan string)
    chB := make(chan string)

    go searchEngineA(query, chA)
    go searchEngineB(query, chB)

    select {
    case result := <-chA:
        fmt.Println("Engine A won! (~300ms)")
        fmt.Println(result)
    case result := <-chB:
        fmt.Println("Engine B won! (~200ms)")
        fmt.Println(result)
    }
}
```

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:




```go
package main

import (
    "fmt"
    "math/rand"
    "sync"
    "time"
)

type Student struct {
    ID         int
    StudyHours int
}

func student(id int, studyHours int, library chan bool, wg *sync.WaitGroup) {
    defer wg.Done()

    library <- true

    fmt.Printf("Student %d entered library, will study for %d hours\n", id, studyHours)

    time.Sleep(time.Duration(studyHours) * time.Second)

    <-library

    fmt.Printf("Student %d left library after %d hours\n", id, studyHours)
}

func main() {
    fmt.Println("=== Library Simulation ===")
    fmt.Println("Library capacity: 30 students")
    fmt.Println("Total students today: 100")
    fmt.Println("Simulation: 1 second = 1 hour\n")

    rand.Seed(time.Now().UnixNano())

    library := make(chan bool, 30)

    var wg sync.WaitGroup

    startTime := time.Now()

    students := make([]Student, 100)
    for i := 0; i < 100; i++ {
        students[i] = Student{
            ID:         i + 1,
            StudyHours: rand.Intn(4) + 1,
        }
    }

    for _, s := range students {
        wg.Add(1)
        go student(s.ID, s.StudyHours, library, &wg)
    }

    wg.Wait()

    duration := time.Since(startTime)

    fmt.Println("\n=== Simulation Complete ===")
    fmt.Printf("Total students served: 100\n")
    fmt.Printf("Library was open for: %.0f hours\n", duration.Seconds())
    fmt.Printf("Peak occupancy: 30 students\n")
}
```

Result:

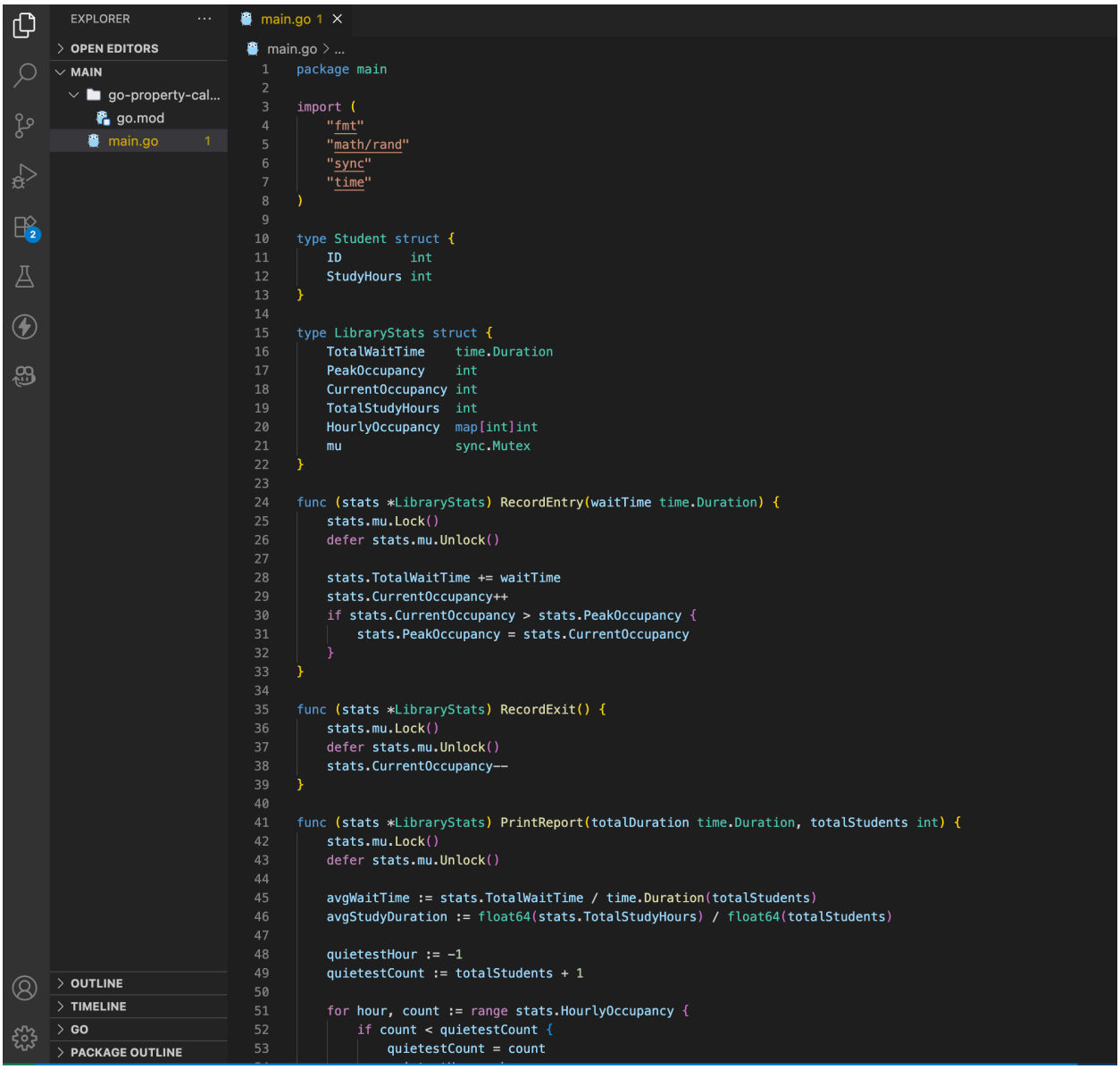
```
Student 16 entered library, will study for 2 hours
Student 1 left library after 2 hours
Student 17 entered library, will study for 3 hours
Student 33 left library after 1 hours
Student 18 entered library, will study for 4 hours
Student 41 left library after 3 hours
Student 78 left library after 3 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
```

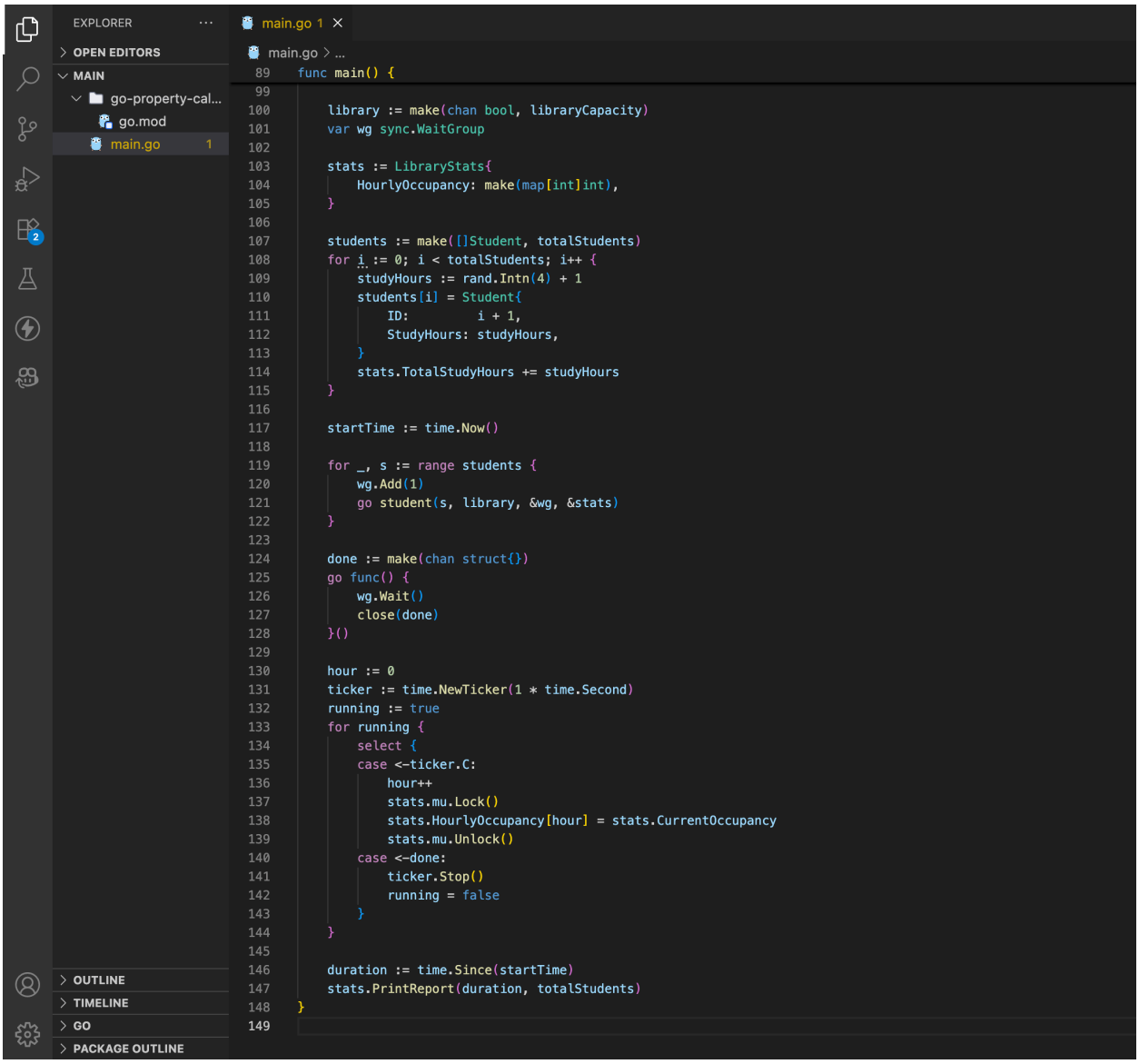
```
Student 61 entered library, will study for 2 hours
Student 57 left library after 1 hours
Student 92 entered library, will study for 1 hours
Student 62 entered library, will study for 4 hours
Student 95 entered library, will study for 4 hours
Student 22 left library after 2 hours
Student 20 left library 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
```

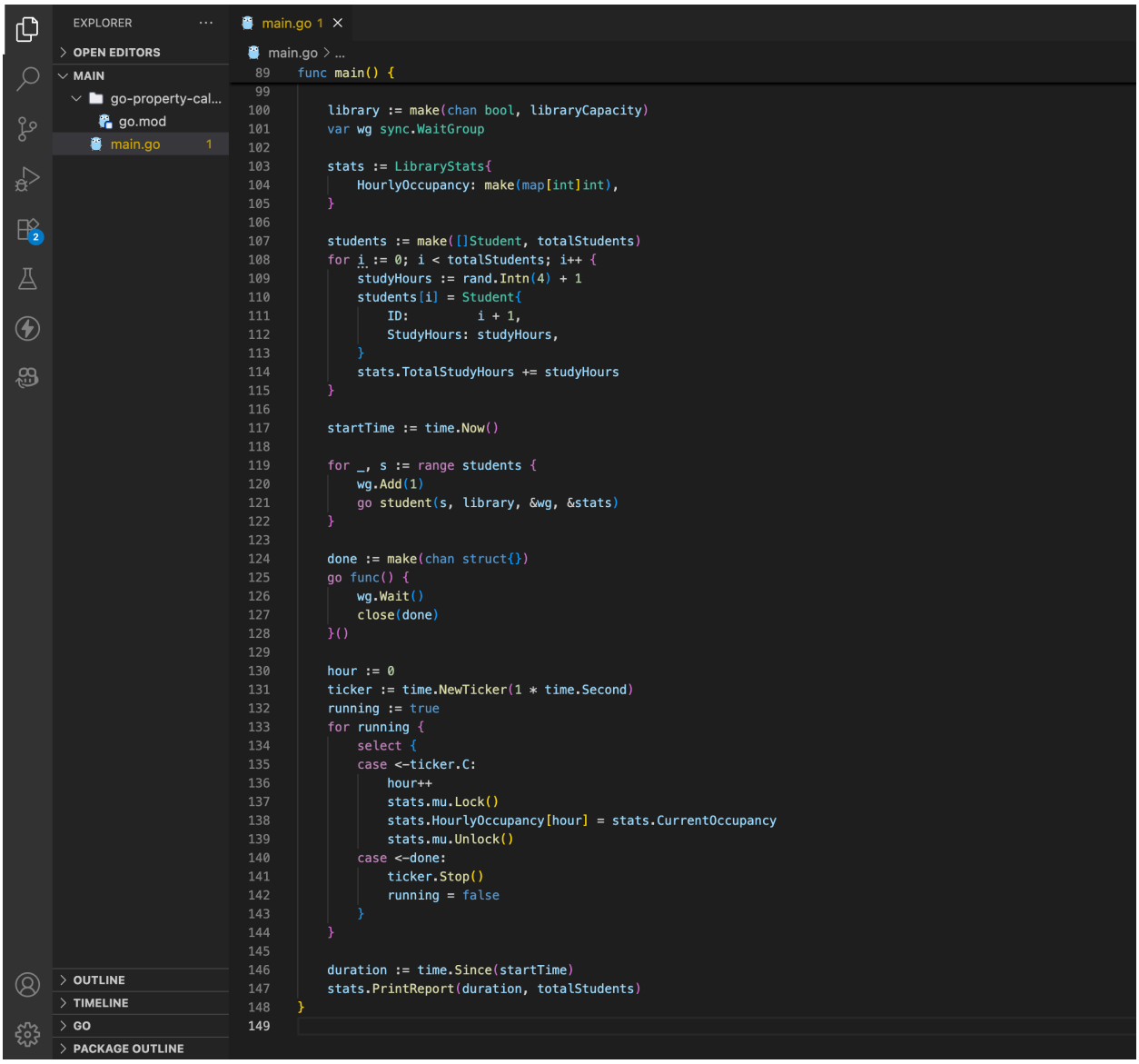
```
Student 69 left library after 2 hours
Student 71 left library after 4 hours
Student 66 left library after 3 hours
Student 70 left library after 3 hours
Student 98 left library after 3 hours
Student 93 left library after 4 hours
Student 62 left library after 4 hours
Student 95 left library after 4 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:

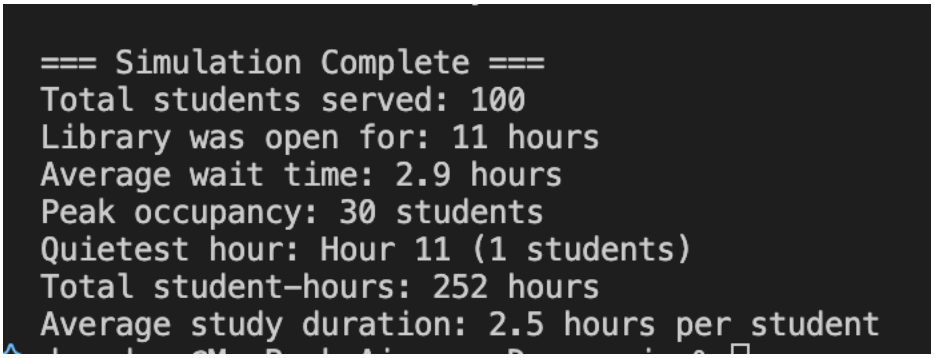
Code:


EXPLORER
OPEN EDITORS
MAIN
go-property-cal...
go.mod
main.go 1
OUTLINE
TIMELINE
GO
PACKAGE OUTLINE
main.go 1
main.go > ...
package main
import (
"fmt"
"math/rand"
"sync"
"time"
)
type Student struct {
ID int
StudyHours int
}
type LibraryStats struct {
TotalWaitTime time.Duration
PeakOccupancy int
CurrentOccupancy int
TotalStudyHours int
HourlyOccupancy map[int]int
mu sync.Mutex
}
func (stats *LibraryStats) RecordEntry(waitTime time.Duration) {
stats.mu.Lock()
defer stats.mu.Unlock()
stats.TotalWaitTime += waitTime
stats.CurrentOccupancy++
if stats.CurrentOccupancy > stats.PeakOccupancy {
stats.PeakOccupancy = stats.CurrentOccupancy
}
}
func (stats *LibraryStats) RecordExit() {
stats.mu.Lock()
defer stats.mu.Unlock()
stats.CurrentOccupancy--
}
func (stats *LibraryStats) PrintReport(totalDuration time.Duration, totalStudents int) {
stats.mu.Lock()
defer stats.mu.Unlock()
avgWaitTime := stats.TotalWaitTime / time.Duration(totalStudents)
avgStudyDuration := float64(stats.TotalStudyHours) / float64(totalStudents)
quietestHour := -1
quietestCount := totalStudents + 1
for hour, count := range stats.HourlyOccupancy {
if count < quietestCount {
quietestCount = count


EXPLORER
OPEN EDITORS
MAIN
go-property-cal...
go.mod
main.go 1
OUTLINE
TIMELINE
GO
PACKAGE OUTLINE
main.go 1
main.go > ...
func main() {
library := make(chan bool, libraryCapacity)
var wg sync.WaitGroup
stats := LibraryStats{
HourlyOccupancy: make(map[int]int),
}
students := make([]Student, totalStudents)
for i := 0; i < totalStudents; i++ {
studyHours := rand.Intn(4) + 1
students[i] = Student{
ID: i + 1,
StudyHours: studyHours,
}
stats.TotalStudyHours += studyHours
}
startTime := time.Now()
for _, s := range students {
wg.Add(1)
go student(s, library, &wg, &stats)
}
done := make(chan struct{})
go func() {
wg.Wait()
close(done)
}()
hour := 0
ticker := time.NewTicker(1 * time.Second)
running := true
for running {
select {
case <-ticker.C:
hour++
stats.mu.Lock()
stats.HourlyOccupancy[hour] = stats.CurrentOccupancy
stats.mu.Unlock()
case <-done:
ticker.Stop()
running = false
}
}
duration := time.Since(startTime)
stats.PrintReport(duration, totalStudents)
}


EXPLORER
main.go 1
OPEN EDITORS
MAIN
go-property-cal...
go.mod
main.go 1
OUTLINE
TIMELINE
GO
PACKAGE OUTLINE
main.go > ...
func main() {
library := make(chan bool, libraryCapacity)
var wg sync.WaitGroup
stats := LibraryStats{
HourlyOccupancy: make(map[int]int),
}
students := make([]Student, totalStudents)
for i := 0; i < totalStudents; i++ {
studyHours := rand.Intn(4) + 1
students[i] = Student{
ID: i + 1,
StudyHours: studyHours,
}
stats.TotalStudyHours += studyHours
}
startTime := time.Now()
for _, s := range students {
wg.Add(1)
go student(s, library, &wg, &stats)
}
done := make(chan struct{})
go func() {
wg.Wait()
close(done)
}()
hour := 0
ticker := time.NewTicker(1 * time.Second)
running := true
for running {
select {
case <-ticker.C:
hour++
stats.mu.Lock()
stats.HourlyOccupancy[hour] = stats.CurrentOccupancy
stats.mu.Unlock()
case <-done:
ticker.Stop()
running = false
}
}
duration := time.Since(startTime)
stats.PrintReport(duration, totalStudents)
}

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