ITMD:469/569 - Week 3 Homework

Travis Smith

Homework Summary

This homework will be a continuation of Labs 1 and 2, demonstrating the use of interfaces to create type agnostic functions.

Program Specifications

Remember Lab 2? Your employer from that assignment has overlooked another expense source: the government! The government will take a 21 percent cut of all earnings after operating expenses have been subtracted. Update your Lab 2 to incorporate this government tax when displaying the total company cost.

Steps

Follow these steps to complete the assignment:

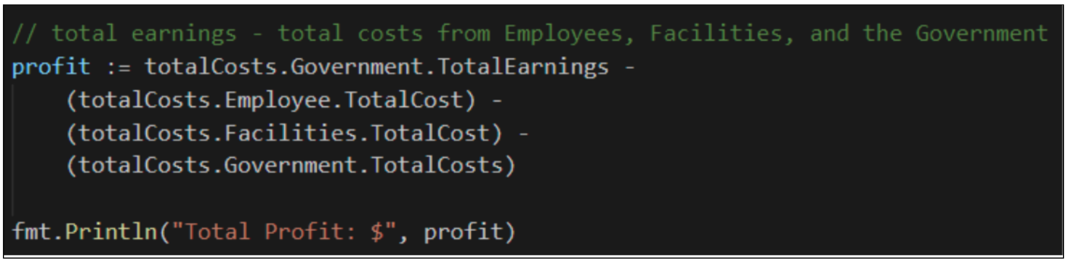
1. Begin by extending Lab 2. Create a new type named 'Government' with the following fields: CorporateTax, TotalEarnings, and TotalCosts, all of type float64.
2. Establish a method on the 'Government' type, named 'calcYearlyCosts', much like you did for the 'Employee' and 'Facilities' types. Use this to calculate and return the taxable amount, derived from total earnings and total costs.
3. We will now utilize Embedded types to organize all our expense-incurring types into a single struct.
4. Before proceeding, add a field named 'TotalCost' to both the 'Employee' and 'Facilities' structs. In your program, after calculating the total cost to the company using the 'calcYearlyCosts' interface method, set the 'TotalCost' field for each type.
5. Develop a type named 'TotalCosts' that embeds the 'Employee', 'Facilities', and 'Government' types. When you instantiate this type, you will have access to and be able to call all the fields and methods of the embedded types.
6. In the main function, following the printout of the Total expenses, instantiate the 'Government' type. Ensure the corporate tax is set to 21 percent, and you can select the total earnings of the company. Set the total costs equal to the total expenses you calculated in

Lab 2. Use the ’calcYearlyCosts’ interface method to calculate the total government cost to the company.

7. Generate an instance of the ’TotalCosts’ struct, setting the fields to match the ’Employee’, ’Facilities’, and ’Government’ instances you have created throughout the lab. Utilize this instance to display the total government expenses and the company’s total profit. Do not use the individual instances of each struct for this task, to demonstrate your understanding of how embedded types function.

8. In summary, within the main function, you will be doing the following: Creating instances of ’Employee’, ’Facility’, and ’Government’, and utilizing the interface function to compute their total costs. Adding these total costs to the newly created ’TotalCosts’ field in each struct, as shown in Step 4b. Use the ’TotalCosts’ field to print out the total company profit.

9. Your implementation should allow you to perform the following at the end of the main function:



```
// total earnings - total costs from Employees, Facilities, and the Government
profit := totalCosts.Government.TotalEarnings -
    (totalCosts.Employee.TotalCost) -
    (totalCosts.Facilities.TotalCost) -
    (totalCosts.Government.TotalCosts)

fmt.Println("Total Profit: $", profit)
```

Submission

Please compress your source code into a ZIP file and upload it to Blackboard. Additionally, attach a separate screenshot demonstrating your code running successfully. week