Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергетиці

Лабораторна робота № 1

з курсу: «Програмування вебзастосунків»

**Виконав:**  
студент 4-го курсу,  
групи ТВ-з11

Роговський Назар Тарасович

Посилання на GitHub репозиторій: <https://github.com/dragonfir2016/LW1_TB-311_Rohovskyi_Nazar_Tarasovych>

**Перевірив:**

Недашківський О.Л.

Київ 2024/2025

Лабораторна робота № 1

**Завдання:**

Завдання 1

Написати мобільний калькулятор для розрахунку складу сухої та горючої маси палива та нижчої теплоти згоряння для робочої, сухої та горючої маси за заданим складом компонентів палива, що задаються у вигляді значень окремих компонентів типу: H\_P, %; C\_P, %; S\_P, %; N\_P, %;O\_P, %; W\_P, %; A\_P, % (див. табл. 1.3.).

Завдання 2

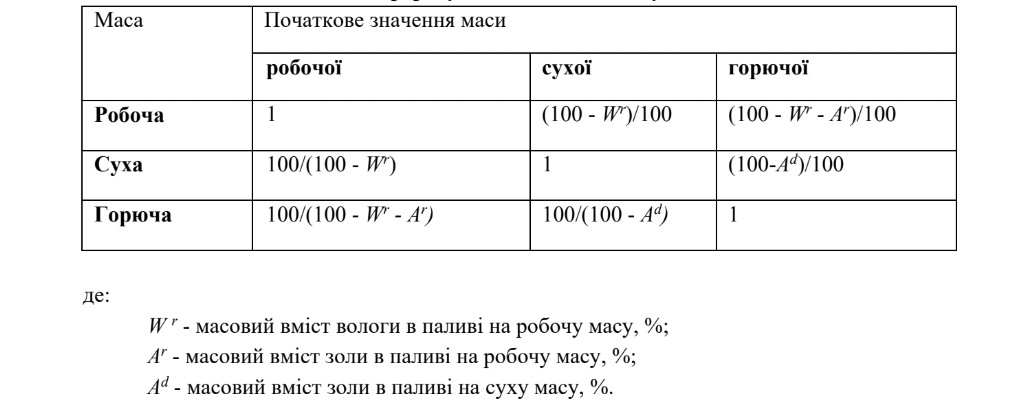
Написати мобільний калькулятор для перерахунку елементарного складу та нижчої теплоти згоряння мазуту на робочу масу для складу горючої маси мазуту, що задається наступними параметрами: вуглець, %; водень, %; кисень, %; сірка, %; нижча теплота згоряння горючої маси мазуту, МДж/кг; вологість робочої маси палива, %; зольність сухої маси, %; вміст

ванадію (V), мг/кг.

**Хід виконання:**

Для написання коду для розрахунку відповідних показників були використані наступні формули:

Для перерахунку масового вмісту складових палива:



Для нижчої теплоти згорання за формулою Мендєлєєва:



Для перерахунку теплоти згоряння:

A white sheet with black text and numbers

Description automatically generated

Завдання 1

Код для Завдання 1 знаходиться у файлі **main.go** у функції task1(). У цьому модулі:

* Користувач вводить значення через консоль.
* func task1() {
* var H\_p, C\_p, S\_p, N\_p, O\_p, W\_p, A\_p float64
* fmt.Println("Введіть значення для завдання 1:")
* fmt.Print("H\_p, C\_p, S\_p, N\_p, O\_p, W\_p, A\_p (через пробіл): ")
* \_, err := fmt.Scanln(&H\_p, &C\_p, &S\_p, &N\_p, &O\_p, &W\_p, &A\_p)
* if err != nil {
* fmt.Println("Помилка введення:", err)
* return
* }
* Після введення даних виконуються обчислення згідно з наведеними формулами.
* K\_PC := 100 / (100 - W\_p)
* K\_PT := 100 / (100 - W\_p - A\_p)
* H\_C := H\_p \* K\_PC
* C\_C := C\_p \* K\_PC
* S\_C := S\_p \* K\_PC
* N\_C := N\_p \* K\_PC
* O\_C := O\_p \* K\_PC
* A\_C := A\_p \* K\_PC
* H\_T := H\_p \* K\_PT
* C\_T := C\_p \* K\_PT
* S\_T := S\_p \* K\_PT
* N\_T := N\_p \* K\_PT
* O\_T := O\_p \* K\_PT
* Q\_P\_H := 339\*C\_p + 1030\*H\_p - 108.8\*(O\_p-S\_p) - 25\*W\_p
* Q\_C\_H := (Q\_P\_H + 0.025\*W\_p) \* (100 / (100 - W\_p))
* Q\_T\_H := (Q\_P\_H + 0.025\*W\_p - A\_p) \* (100 / (100 - W\_p - A\_p))
* Результати обчислень (склад сухої та горючої маси, нижча теплота згоряння) виводяться з округленням до двох десяткових знаків.
* fmt.Println("\nКоефіцієнти переходу:")
* fmt.Printf("K\_PC: %.2f\n", round(K\_PC))
* fmt.Printf("K\_PT: %.2f\n", round(K\_PT))
* fmt.Println("\nСклад сухої маси:")
* fmt.Printf("H\_C: %.2f\n", round(H\_C))
* fmt.Printf("C\_C: %.2f\n", round(C\_C))
* fmt.Printf("S\_C: %.2f\n", round(S\_C))
* fmt.Printf("N\_C: %.2f\n", round(N\_C))
* fmt.Printf("O\_C: %.2f\n", round(O\_C))
* fmt.Printf("A\_C: %.2f\n", round(A\_C))
* fmt.Println("\nСклад горючої маси:")
* fmt.Printf("H\_T: %.2f\n", round(H\_T))
* fmt.Printf("C\_T: %.2f\n", round(C\_T))
* fmt.Printf("S\_T: %.2f\n", round(S\_T))
* fmt.Printf("N\_T: %.2f\n", round(N\_T))
* fmt.Printf("O\_T: %.2f\n", round(O\_T))
* fmt.Println("\nНижча теплота згоряння:")
* fmt.Printf("Q\_P\_H: %.2f кДж/кг\n", round(Q\_P\_H))
* fmt.Printf("Q\_C\_H: %.2f кДж/кг\n", round(Q\_C\_H))
* fmt.Printf("Q\_T\_H: %.2f кДж/кг\n", round(Q\_T\_H))

При запуску контрольного прикладу виходять наступні результати:

A screenshot of a computer

AI-generated content may be incorrect.

Та при запуску варіанту 7 з таблиці з варіантами наступні результати:

A screenshot of a computer

AI-generated content may be incorrect.

Завдання 2

Код для Завдання 2 знаходиться у файлі **main.go** у функції task2(). Алгоритм роботи наступний:

* Зчитування значень, введених користувачем, з консолі.
* var H\_g, C\_g, S\_g, O\_g, A\_g, W\_p\_mazut, Q\_g float64
* fmt.Println("Введіть значення для завдання 2:")
* fmt.Print("H\_g, C\_g, S\_g, O\_g, A\_g, W\_p\_mazut, Q\_g (через пробіл): ")
* \_, err := fmt.Scanln(&H\_g, &C\_g, &S\_g, &O\_g, &A\_g, &W\_p\_mazut, &Q\_g)
* if err != nil {
* fmt.Println("Помилка введення:", err)
* return
* }
* Обчислення коефіцієнта переходу до робочої маси:
* K\_GC := 100 / (100 - W\_p\_mazut - A\_g)
* Перерахунок складу робочої маси та розрахунок нижчої теплоти згоряння:
* H\_p := H\_g \* K\_GC
* C\_p := C\_g \* K\_GC
* S\_p := S\_g \* K\_GC
* O\_p := O\_g \* K\_GC
* A\_p := A\_g \* (100 / (100 - W\_p\_mazut))
* Q\_p := Q\_g \* (100 - W\_p\_mazut - A\_g) / 100
* Результати виводяться на екран.
* fmt.Println("\nКоефіцієнт переходу до робочої маси:")
* fmt.Printf("K\_GC: %.2f\n", round(K\_GC))
* fmt.Println("\nСклад робочої маси:")
* fmt.Printf("H\_p: %.2f\n", round(H\_p))
* fmt.Printf("C\_p: %.2f\n", round(C\_p))
* fmt.Printf("S\_p: %.2f\n", round(S\_p))
* fmt.Printf("O\_p: %.2f\n", round(O\_p))
* fmt.Printf("A\_p: %.2f\n", round(A\_p))
* fmt.Println("\nНижча теплота згоряння:")
* fmt.Printf("Q\_p: %.2f МДж/кг\n", round(Q\_p))

При запуску контрольного прикладу виходять наступні результати:

A computer screen with white text

AI-generated content may be incorrect.

Та при введенні довільних даних наступні результати:

A screenshot of a computer

AI-generated content may be incorrect.

**Висновок**

В результаті виконання лабораторної роботи №1 було засвоєно базові арифметичні операції на мові Go. Реалізація дозволила:

* Освоїти процес зчитування даних із консолі.
* Виконувати математичні обчислення та використовувати функції для округлення значень.
* Застосувати принципи розробки консольних додатків для вирішення завдань, які спочатку були реалізовані в мові Kotlin.

Лабораторна робота сприяла закріпленню навичок роботи з базовими конструкціями мови Go, а також продемонструвала можливості цієї мови для реалізації обчислювальних задач.

Попередня реалізація на Kotlin була орієнтована на мобільний застосунок із графічним інтерфейсом, а Go-версія реалізована як консольна програма. Обидві версії виконують ті ж обчислення, проте Go-рішення простіше у налаштуванні та запуску з командного рядка.