**Лабораторна робота № 1**

**РОБОТА З БАЗОВИМИ ТИПАМИ ДАНИХ**

**Мета:** ознайомитись з основними можливостями мови програмування Go, найпростішої структурою програми, з властивостями базових типів. Отримати практичні навички по роботі з базовими типами даних в мові Go.

**Хід роботи:**

**Завдання 1:**

# Лістинг коду:

*/\**

*Многострочный комментарий*

*Структура программы*

*\*/*

package main

*//Однострочный комментарий*

*//Импорт пакетов*

import "fmt"

func main() {

*//Объявление переменной*

    var str string = "Golang!"

*//Вывод в консоль текста*

    fmt.Println("Привіт ", str)

*//Задание.*

*//1. Вывести текст на украинском языке*

}

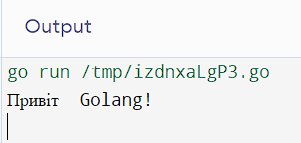


Рис.1. Реалізація завдання.

**Завдання 2:**

# Лістинг коду:

package main

import "fmt"

func main() {

*//Все переменные инициализируются значением по умолчанию, т.е. 0*

    var defaultInt8 int8

    var defaultInt16 int16

    var defaultInt32 int32

    var defaultInt64 int64

    var defaultInt int

    fmt.Println("Default values (signed): ", defaultInt8, defaultInt16, defaultInt32, defaultInt64, defaultInt)

    var defaultuInt8 uint8

    var defaultuInt16 uint16

    var defaultuInt32 uint32

    var defaultuInt64 uint64

    var defaultuInt uint

    fmt.Println("Default values (unsigned): ", defaultuInt8, defaultuInt16, defaultuInt32, defaultuInt64, defaultuInt)

*//Задание.*

*//1. Создать целочисленную переменную (результат не отображать)*

    var customInt int

}

**Завдання 3:**

# Лістинг коду:

package main

import "fmt"

func main() {

*//Инициализация переменных*

    var userinit8 uint8 = 1

    var userinit16 uint16 = 2

    var userinit64 int64 = -3

    var userautoinit = -4 *//Такой вариант инициализации также возможен*

    fmt.Println("Values: ", userinit8, userinit16, userinit64, userautoinit, "\n")

*//Краткая запись объявления переменной*

*//только для новых переменных*

    intVar := 10

    fmt.Printf("Value = %d Type = %T\n", intVar, intVar)

*//Задание.*

*//1. Вывести типы всех переменных*

    fmt.Printf("Type of userinit8: %T\n", userinit8)

    fmt.Printf("Type of userinit16: %T\n", userinit16)

    fmt.Printf("Type of userinit64: %T\n", userinit64)

    fmt.Printf("Type of userautoinit: %T\n", userautoinit)

    fmt.Printf("Type of intVar: %T\n", intVar)

*//2. Присвоить переменной intVar переменные userinit16 и userautoinit. Результат вывести.*

    intVar = int(userinit16)

    fmt.Printf("Updated intVar after assigning userinit16: %d\n", intVar)

    intVar = int(userautoinit)

    fmt.Printf("Updated intVar after assigning userautoinit: %d\n", intVar)

}

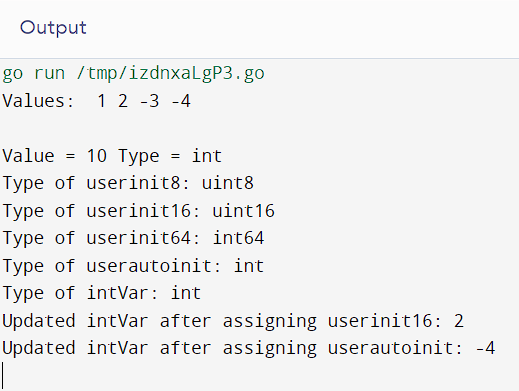


Рис.2. Реалізація завдання.

**Завдання 4:**

Лістинг коду:

package main

import "fmt"

func main() {

*// Инициализация переменных*

    var a int16 = 2

    var b int16 = 3

    var c int64 = 10

    fmt.Println("a + b        = ", a+b)

    fmt.Println("a - b        = ", a-b)

    fmt.Println("a \* b        = ", a\*b)

    fmt.Println("int(a / b)   = ", int(a/b), "\n")

    c--

    fmt.Println("c--     = ", c)

    c++

    fmt.Println("c++     = ", c)

    c += 10

    fmt.Println("c += 10 = ", c)

    c -= 5

    fmt.Println("c -= 5  = ", c)

    c \*= 3

    fmt.Println("c \*= 3  = ", c)

    c /= 7

    fmt.Println("c /= 7  = ", c)

*// Задание.*

*// 1. Выполнить выражение: c = c - 1*

    c = c - 1

    fmt.Println("c = c - 1  = ", c)

}

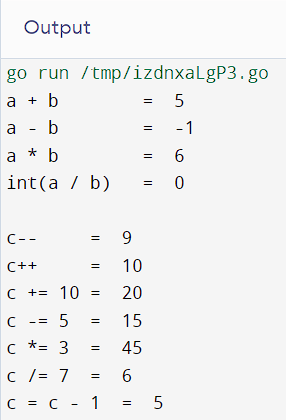


Рис.3. Реалізація завдання.

**Завдання 5:**

Лістинг коду:

package main

import (

    "fmt"

    "runtime"

)

func main() {

    fmt.Println("Синонимы целых типов\n")

    fmt.Println("byte    - int8")

    fmt.Println("rune    - int32")

    fmt.Println("int     - int32, или int64, в зависимости от ОС")

    fmt.Println("uint    - uint32, или uint64, в зависимости от ОС")

*// Задание.*

*// 1. Определить разрядность ОС*

    bits := 8 \* runtime.GOARCH *// визначаємо розрядність ОС у бітах*

    fmt.Printf("Разрядность ОС: %d бит\n", bits)

}

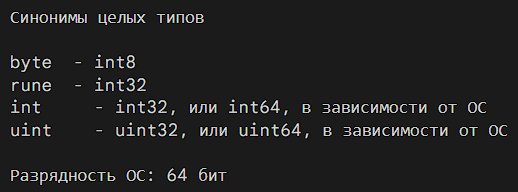


Рис.4. Реалізація завдання.

**Завдання 6:**

Лістинг коду:

package main

import "fmt"

func main() {

    var x, y, z uint8

    x = 9

    y = 28

    z = x

    fmt.Println("Битовые операции")

    fmt.Printf("^x      = ^(%d)      = ^(%.8b)            = %.8b = %d\n", x, x, ^x, ^x)

    fmt.Printf("x << 2  = (%d << 2)  = (%.8b << 2)        = %.8b = %d\n", x, x, x<<2, x<<2)

    fmt.Printf("x >> 2  = (%d >> 2)  = (%.8b >> 2)        = %.8b = %d\n", x, x, x>>2, x>>2)

    fmt.Printf("x & y   = (%d & %d)  = (%.8b & %.8b)  = %.8b = %d\n", x, y, x, y, x&y, x&y)

    fmt.Printf("x | y   = (%d | %d)  = (%.8b | %.8b)  = %.8b = %d\n", x, y, x, y, x|y, x|y)

    fmt.Printf("x ^ y   = (%d ^ %d)  = (%.8b ^ %.8b)  = %.8b = %d\n", x, y, x, y, x^y, x^y)

    fmt.Printf("x &^ y  = (%d &^ %d) = (%.8b &^ %.8b) = %.8b = %d\n", x, y, x, y, x&^y, x&^y)

    fmt.Printf("x %% y   = (%d %% %d)  = (%.8b %% %.8b)  = %.8b = %d\n", x, y, x, y, x%y, x%y)

    fmt.Println("\nБитовые операции с присваиванием")

    x = z

    x &= y

    fmt.Printf("x &= y   = (%d &= %d)  = (%.8b &= %.8b)  = %.8b = %d\n", z, y, z, y, x, x)

    x = z

    x |= y

    fmt.Printf("x |= y   = (%d |= %d)  = (%.8b |= %.8b)  = %.8b = %d\n", z, y, z, y, x, x)

    x = z

    x ^= y

    fmt.Printf("x ^= y   = (%d ^= %d)  = (%.8b ^= %.8b)  = %.8b = %d\n", z, y, z, y, x, x)

    x = z

    x &^= y

    fmt.Printf("x &^= y  = (%d &^= %d) = (%.8b &^= %.8b) = %.8b = %d\n", z, y, z, y, x, x)

    x = z

    x %= y

    fmt.Printf("x %%= y   = (%d %%= %d)  = (%.8b %%= %.8b)  = %.8b = %d\n", z, y, z, y, x, x)

1. ^x - Побітове не (^) для числа x. Ця операція інвертує всі біти числа.

2. x << 2 - Зсув вліво (<<) для числа x на 2 позиції.

3. x >> 2 - Зсув вправо (>>) для числа x на 2 позиції.

4. x & y - Побітове і (&) для чисел x і y.

5. x | y - Побітове або (|) для чисел x і y.

6. x ^ y - Побітове виключне або (^) для чисел x і y.

7. `x &^ y` - Побітове і не (`&^`) для чисел `x` і `y`.

8. `x % y` - Остача від ділення числа `x` на `y`.

Операції з присвоєнням:

1. x &= y - Побітове і з присвоєнням.

2. x |= y - Побітове або з присвоєнням.

3. x ^= y – Побітове виключне або з присвоєнням.

4. x &^= y - Побітове і не з присвоєнням.

5. x %= y - Остача від ділення з присвоєнням.

**Завдання 7:**

Лістинг коду:

package main

import "fmt"

func main() {

    variable8 := int8(127)

    variable16 := int16(16383)

    fmt.Println("Приведение типов\n")

    fmt.Printf("variable8         = %-5d = %.16b\n", variable8, variable8)

    fmt.Printf("variable16        = %-5d = %.16b\n", variable16, variable16)

    fmt.Printf("uint16(variable8) = %-5d = %.16b\n", uint16(variable8), uint16(variable8))

    fmt.Printf("uint8(variable16) = %-5d = %.16b\n", uint8(variable16), uint8(variable16))

*// Задание.*

*// 1. Создайте 2 переменные разных типов. Выполните арифметические операции. Результат вывести*

*// Створення двох змінних різних типів*

    var numberInt int = 42

    var numberFloat float64 = 3.14

*// Арифметичні операції*

    sum := numberInt + int(numberFloat)

    difference := numberInt - int(numberFloat)

    product := numberInt \* int(numberFloat)

    quotient := numberInt / int(numberFloat)

*// Вивід результатів*

    fmt.Printf("\nАрифметичні операції:\n")

    fmt.Printf("Сума: %d\n", sum)

    fmt.Printf("Різниця: %d\n", difference)

    fmt.Printf("Добуток: %d\n", product)

    fmt.Printf("Частка: %d\n", quotient)

}

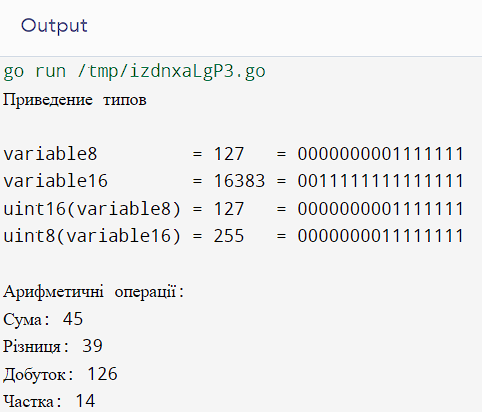


Рис.7. Реалізація завдання.

**Завдання 8:**

Лістинг коду:

package main

*// Імпорт нескольких пакетів*

import (

    "fmt"

    "math"

)

func main() {

    var defaultFloat float32

    var defaultDouble float64 = 5.5

    fmt.Println("defaultFloat       = ", defaultFloat)

    fmt.Printf("defaultDouble (%T) = %f\n\n", defaultDouble, defaultDouble)

    fmt.Println("MAX float32        = ", math.MaxFloat32)

    fmt.Println("MIN float32        = ", math.SmallestNonzeroFloat32, "\n")

    fmt.Println("MAX float64        = ", math.MaxFloat64)

    fmt.Println("MIN float64        = ", math.SmallestNonzeroFloat64, "\n")

*// Задание.*

*// 1. Создайте переменные разных типов, используя краткую запись и инициализацию по-умолчанию. Результат вывести*

*// Створення змінних різних типів*

    shortFloat := float32(3.14)

    shortDouble := 42.0

*// Вивід результатів*

    fmt.Printf("shortFloat (%T) = %f\n", shortFloat, shortFloat)

    fmt.Printf("shortDouble (%T) = %f\n", shortDouble, shortDouble)

}

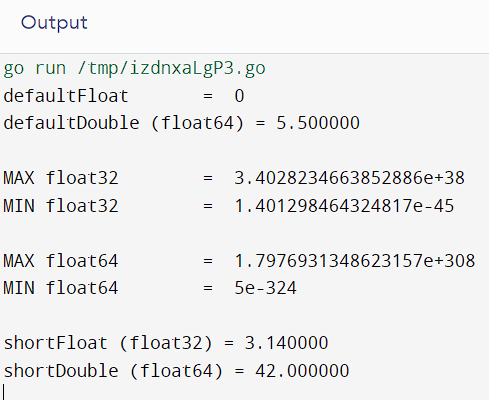


Рис.8. Реалізація завдання.

**Завдання 9:**

Лістинг коду:

package main

import "fmt"

func main() {

    var first, second bool

    var third bool = true

    fourth := !third

    var fifth = true

    fmt.Println("first  = ", first)       *// false*

    fmt.Println("second = ", second)      *// false*

    fmt.Println("third  = ", third)       *// true*

    fmt.Println("fourth = ", fourth)      *// false*

    fmt.Println("fifth  = ", fifth, "\n") *// true*

    fmt.Println("!true  = ", !true)        *// false*

    fmt.Println("!false = ", !false, "\n") *// true*

    fmt.Println("true && true   = ", true && true)         *// true*

    fmt.Println("true && false  = ", true && false)        *// false*

    fmt.Println("false && false = ", false && false, "\n") *// false*

    fmt.Println("true || true   = ", true || true)         *// true*

    fmt.Println("true || false  = ", true || false)        *// true*

    fmt.Println("false || false = ", false || false, "\n") *// false*

    fmt.Println("2 < 3  = ", 2 < 3)        *// true*

    fmt.Println("2 > 3  = ", 2 > 3)        *// false*

    fmt.Println("3 < 3  = ", 3 < 3)        *// false*

    fmt.Println("3 <= 3 = ", 3 <= 3)       *// true*

    fmt.Println("3 > 3  = ", 3 > 3)        *// false*

    fmt.Println("3 >= 3 = ", 3 >= 3)       *// true*

    fmt.Println("2 == 3 = ", 2 == 3)       *// false*

    fmt.Println("3 == 3 = ", 3 == 3)       *// true*

    fmt.Println("2 != 3 = ", 2 != 3)       *// true*

    fmt.Println("3 != 3 = ", 3 != 3, "\n") *// false*

*//Задание.*

*//1. Пояснить результаты операций*

!true - Логічне НЕ (!) для true дає false.

!false - Логічне НЕ для false дає true.

Логічне І (&&):

true && true - Логічне І для двох true дає true.

true && false - Логічне І для true та false дає false.

false && false - Логічне І для двох false дає false.

Логічне АБО (||):

true || true - Логічне АБО для двох true дає true.

true || false - Логічне АБО для true та false дає true.

false || false - Логічне АБО для двох false дає false.

Оператори порівняння:

2 < 3 - Перевірка, чи 2 менше 3 (true).

2 > 3 - Перевірка, чи 2 більше 3 (false).

3 < 3 - Перевірка, чи 3 менше 3 (false).

3 <= 3 - Перевірка, чи 3 менше або дорівнює 3 (true).

3 > 3 - Перевірка, чи 3 більше 3 (false).

3 >= 3 - Перевірка, чи 3 більше або дорівнює 3 (true).

2 == 3 - Перевірка, чи 2 дорівнює 3 (false).

3 == 3 - Перевірка, чи 3 дорівнює 3 (true).

2 != 3 - Перевірка, чи 2 не дорівнює 3 (true).

3 != 3 - Перевірка, чи 3 не дорівнює 3 (false).

**Завдання 10:**

Лістинг коду:

package main

import "fmt"

func main() {

    var chartype int8 = 'R'

    fmt.Printf("Code '%c' - %d\n", chartype, chartype)

*// Задание.*

*// 1. Вывести украинскую букву 'Ї'*

    ukrainianChar := 'Ї'

    fmt.Printf("Code '%c' - %d\n", ukrainianChar, ukrainianChar)

*// 2. Пояснить назначение типа "rune"*

    var runeVar rune = 'Я'

    fmt.Printf("Code '%c' - %d (Type: %T)\n", runeVar, runeVar, runeVar)

}

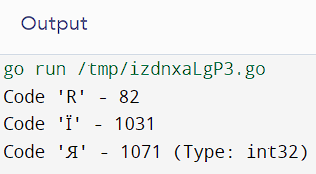


Рис.10. Реалізація завдання.

**Висновок:** У ході виконання завдань ми отримали практичні навички створення програм на Go, ознайомились з базовою структурою програм, використанням коментарів, імпортом пакетів, та оголошенням та ініціалізацією змінних.