

# 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.** Вивести текст українською мовою.

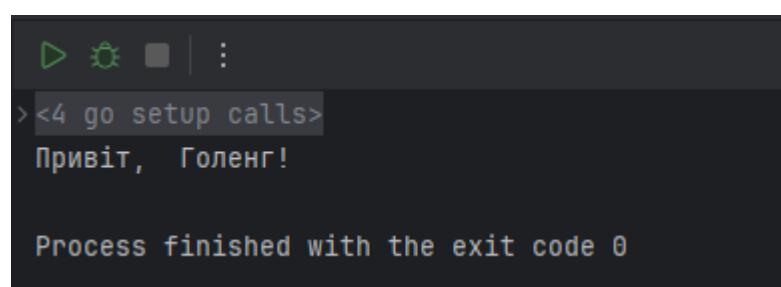
Листинг програми:

```
package main

import "fmt"

func main() {
    var str string = "Голенг!"
    fmt.Println("Привіт, ", str)
}
```

Результат виконання:



```
><4 go setup calls>
Привіт, Голенг!

Process finished with the exit code 0
```

Рис. 1. Завдання 1.

**Завдання 1.** Створити ціличисельну змінну (результат не відображати)

Листинг програми:

```
package main

import "fmt"

func main() {
    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)
```

Змн.	Арк.	№ докум.	Підпис	Дата
Розроб.	Риженко Я.В.			
Перевір.	Петросян Р.В.			
Керівник				
Н. контр.				
Зав. каф.				

ДУ «Житомирська політехніка».25.121.20.000 – Пр 1

Звіт з  
лабораторної роботи

Лім.	Арк.	Аркушів
	1	12
ФІКТ Гр. ІПЗ-23-1		

```

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)

// Завдання.
// 1. Створити цілочисельну змінну (результат не відображати)
var intVariable int
fmt.Printf("Type: %T Value: %v\n", intVariable, intVariable)
}

```

Результат виконання:

```

> <4 go setup calls>
Default values (signed): 0 0 0 0 0
Default values (unsigned): 0 0 0 0 0
Type: int Value: 0

Process finished with the exit code 0

```

Рис. 2. Завдання 2.

### Завдання 3.

1. Вивести типи всіх змінних

2. Присвоїти змінній intVar змінні userinit16 і userautoinit. Результат вивести в консоль.

Листинг програми:

```

package main

import "fmt"

func main() {
    var userinit8 uint8 = 1
    var userinit16 uint16 = 2
    var userinit64 int64 = -3
    var userautoinit = -4

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

    intVar := 10

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

    // Завдання.
    // 1. Вивести типи всіх змінних
    // 2. Присвоїти змінній intVar змінні userinit16 і userautoinit. Результат
    // вивести в консоль.
    fmt.Printf("Value = %d Type = %T\n", userinit8, userinit8)
    fmt.Printf("Value = %d Type = %T\n", userinit16, userinit16)
    fmt.Printf("Value = %d Type = %T\n", userinit64, userinit64)
    fmt.Printf("Value = %d Type = %T\n", userautoinit, userautoinit)
}

```

		Риженко Я.В			ДУ «Житомирська політехніка».25.121.20.000 – Пр1	Арк.
		Петросян Р.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

intVar = int(userinit16)
fmt.Printf("Value = %d Type = %T\n", intVar, intVar)
intVar = userautoinit
}

```

Результат виконання:

```

> <4 go setup calls>
Values: 1 2 -3 -4

Value = 10 Type = int
Value = 1 Type = uint8
Value = 2 Type = uint16
Value = -3 Type = int64
Value = -4 Type = int
Value = 2 Type = int

Process finished with the exit code 0

```

Рис. 3. Завдання 3.

**Завдання 4.** Виконати вираз: fmt.Println("c-- = ", c--)

Листинг програми:

```

package main

import "fmt"

func main() {
    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")

    fmt.Println("c =", c)
    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)

    // Завдання.
    // 1. Виконати вираз: fmt.Println("c--      = ", c--)
    fmt.Println("c =", c)
}

```

		Рижсенко Я.В			ДУ «Житомирська політехніка».25.121.20.000 – Пр1	Арк.
		Петросян Р.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

```

    c--
    fmt.Println("c-- (after) =", c)
}

```

Результат виконання:

```

><4 go setup calls>
a + b      = 5
a - b      = -1
a * b      = 6
int(a / b) = 0

c = 10
c--      = 9
c++      = 10
c += 10 = 20
c -= 5  = 15
c *= 3   = 45
c /= 7   = 6
c = 6
c-- (after) = 5

Process finished with the exit code 0

```

Рис. 4. Завдання 4.

### Завдання 5. Визначити розрядність платформи

Листинг програми:

```

package main

import (
    "fmt"
    "strconv"
)

func main() {
    fmt.Println("Синоніми цілих типів\n")

    fmt.Println("byte     - int8")
    fmt.Println("rune    - int32")
    fmt.Println("int      - int32 або int64, в залежності від платформи")
    fmt.Println("uint     - uint32 або uint64, в залежності від платформи", "\n")

    // Завдання.
    // 1. Визначити розрядність платформи
    fmt.Println("Platform bitness:", strconv.IntSize, "bit")
}

```

		<i>Рижсенко Я.В</i>		
		<i>Петросян Р.В.</i>		
Змн.	Арк.	№ докум.	Підпис	Дата

Результат виконання:

```
> <4 go setup calls>
Синоніми цілих типів

byte     - int8
rune    - int32
int      - int32 або int64, в залежності від платформи
uint     - uint32 або uint64, в залежності від платформи

Platform bitness: 64 bit

Process finished with the exit code 0
```

Рис. 5. Завдання 5.

### Завдання 6. Пояснити результати операцій

Листинг програми:

```
package main

import "fmt"

func main() {
    var x, y, z uint8

    x = 9 // 9 = 00001001
    y = 28 // 28 = 00011100
    z = x

    fmt.Println("Бітові операції")

    // ^x – побітове NOT: інвертує всі біти (0→1, 1→0)
    // 00001001 -> 11110110 = 246 (бо uint8: 255 - 9 = 246)
    fmt.Printf("^x      = ^(%d)      = ^(%.8b)      = %.8b = %d\n",
        x, x, ^x, ^x)

    // x << 2 – зсув вліво на 2 біти (множення на 4)
    // 00001001 << 2 = 00100100 = 36
    fmt.Printf("x << 2 = (%d << 2) = (%.8b << 2)      = %.8b = %d\n",
        x, x, x<<2, x<<2)

    // x >> 2 – зсув вправо на 2 біти (ділення на 4)
    // 00001001 >> 2 = 00000010 = 2
    fmt.Printf("x >> 2 = (%d >> 2) = (%.8b >> 2)      = %.8b = %d\n",
        x, x, x>>2, x>>2)

    // x & y – AND: біт = 1 лише якщо 1 в обох
    // 00001001
    // 00011100
    // ----- AND
    // 00001000 = 8
    fmt.Printf("x & y = (%d & %d) = (%.8b & %.8b) = %.8b = %d\n",
        x, y, x, y, x&y, x&y)
```

		Рижсенко Я.В		
		Петросян Р.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

// x | y - OR: біт = 1, якщо хоч один = 1
// 00001001
// 00011100
// ----- OR
// 00011101 = 29
fmt.Printf("x | y = (%d | %d) = (%.8b | %.8b) = %.8b = %d\n",
           x, y, x, y, x|y, x|y)

// x ^ y - XOR: біт = 1, якщо біти різні
// 00001001
// 00011100
// ----- XOR
// 00010101 = 21
fmt.Printf("x ^ y = (%d ^ %d) = (%.8b ^ %.8b) = %.8b = %d\n",
           x, y, x, y, x^y, x^y)

// x &^ y - AND NOT: зберігає 1 тільки там, де x=1 і y=0
// 00001001
// 00011100
// ----- AND-NOT
// 00000001 = 1
fmt.Printf("x &^ y = (%d &^ %d) = (%.8b &^ %.8b) = %.8b = %d\n",
           x, y, x, y, x&^y, x&^y)

// x % y - остача від ділення
// 9 < 28, тому остача = саме 9
fmt.Printf("x %% y = (%d %% %d) = (%.8b %% %.8b) = %.8b = %d\n",
           x, y, x, y, x%y, x%y)

fmt.Println("\nБітові операції з присвоюванням")

// x &= y - те саме, що x = x & y
// результат такий самий, як AND вище: 8
x = z
x &= y
fmt.Printf("x &= y = (%d &= %d) = (%.8b &= %.8b) = %.8b = %d\n",
           z, y, z, y, x, x)

// x |= y - те саме, що x = x | y
// результат такий самий, як OR вище: 29
x = z
x |= y
fmt.Printf("x |= y = (%d |= %d) = (%.8b |= %.8b) = %.8b = %d\n",
           z, y, z, y, x, x)

// x ^= y - те саме, що x = x ^ y
// результат такий самий, як XOR: 21
x = z
x ^= y
fmt.Printf("x ^= y = (%d ^= %d) = (%.8b ^= %.8b) = %.8b = %d\n",
           z, y, z, y, x, x)

// x &^= y - те саме, що x = x &^ y
// результат: 1
x = z
x &^= y
fmt.Printf("x &^= y = (%d &^= %d) = (%.8b &^= %.8b) = %.8b = %d\n",
           z, y, z, y, x, x)

// x %= y - остача від ділення
// 9 % 28 = 9
x = z

```

		Риженко Я.В		
		Петросян Р.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

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

```

Результат виконання:

```

> <4 go setup calls>
Бітові операції
^x      = ^(9)      = ^(00001001)      = 11110110 = 246
x << 2  = (9 << 2) = (00001001 << 2)      = 00100100 = 36
x >> 2  = (9 >> 2) = (00001001 >> 2)      = 00000010 = 2
x & y   = (9 & 28) = (00001001 & 00011100) = 00001000 = 8
x | y   = (9 | 28) = (00001001 | 00011100) = 00011101 = 29
x ^ y   = (9 ^ 28) = (00001001 ^ 00011100) = 00010101 = 21
x &^ y  = (9 &^ 28) = (00001001 &^ 00011100) = 00000001 = 1
x % y   = (9 % 28) = (00001001 % 00011100) = 00001001 = 9

Бітові операції з присвоюванням
x &= y  = (9 &= 28) = (00001001 &= 00011100) = 00001000 = 8
x |= y  = (9 |= 28) = (00001001 |= 00011100) = 00011101 = 29
x ^= y  = (9 ^= 28) = (00001001 ^= 00011100) = 00010101 = 21
x &^= y = (9 &^= 28) = (00001001 &^= 00011100) = 00000001 = 1
x %= y  = (9 %= 28) = (00001001 %= 00011100) = 00001001 = 9

Process finished with the exit code 0

```

Рис. 6. Завдання 6.

**Завдання 7.** Створіть 2 змінні різних типів. Виконайте арифметичні операції.

Результат вивести в консоль

Листинг програми:

```

package main

import (
    "fmt"
    "strconv"
)

func main() {
    variable8 := int8(127)
    variable16 := int16(16383)

    fmt.Println("Приведення типів\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))
}

```

		Рижсенко Я.В			Арк.
		Петросян Р.В.			
Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25.121.20.000 – Пр1

```

uint8(variable16)

    // Завдання. //
    //1. Створіть 2 змінні різних типів. Виконайте арифметичні операції. Результат
вивести в консоль
    variableAuto := 57
    variableInt16 := int16(27)
    variableString := "Check"

    // 1. Арифметика між сумісними типами – ОК
    sum1 := variableAuto + int(variableInt16)
    fmt.Println("\nРезультат додавання двох Int різних розрядів з перетворенням
int16 на Int(auto):", sum1)

    // 2. Спроба додати int + string (імітація помилки)
    fmt.Println("\nСпроба додати int i string:")

    // Це неможливо → виводимо помилку самостійно
    fmt.Println("Помилка: неможливо додати 'int' i 'string'")

    // 3. Коректний варіант – конкатенація
    sum3 := strconv.Itoa(variableAuto) + variableString
    fmt.Println("\nРезультат конкатенації:", sum3)
}

```

Результат виконання:

```

><4 go setup calls>
Приведення типів

variable8      = 127      = 0000000011111111
variable16     = 16383   = 0011111111111111
uint16(variable8) = 127      = 0000000011111111
uint8(variable16) = 255      = 0000000111111111

Результат додавання двох Int різних розрядів з перетворенням int16 на Int(auto): 84

Спроба додати int i string:
Помилка: неможливо додати 'int' i 'string'

Результат конкатенації: 57Check

```

Рис. 7. Завдання 7.

**Завдання 8.** Створіть змінні різних типів, використовуючи короткий запис та ініціалізацію за замовчуванням. Результат вивести в консоль

Листинг програми:

```

package main

//Імпорт декількох пакетов
import (
    "fmt"
    "math"
)

func main() {

```

		Рижсенко Я.В					
		Петросян Р.В.					
Змн.	Арк.	№ докум.	Підпис	Дата		ДУ «Житомирська політехніка».25.121.20.000 – Пр1	Арк.

```

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")

// Завдання.
// 1. Створіть змінні різних типів, використовуючи короткий запис та ініціалізацію за замовчуванням. Результат вивести в консоль
intVar := 42
floatVar := 3.14159
boolVar := true
stringVar := "Hello, Go!"
complexVar := complex(2, 3)

fmt.Println("Змінні різних типів:")
fmt.Printf("intVar      = %v (%T)\n", intVar, intVar)
fmt.Printf("floatVar    = %v (%T)\n", floatVar, floatVar)
fmt.Printf("boolVar     = %v (%T)\n", boolVar, boolVar)
fmt.Printf("stringVar   = %v (%T)\n", stringVar, stringVar)
fmt.Printf("complexVar  = %v (%T)\n", complexVar, complexVar)
}

```

Результат виконання:

```

><4 go setup calls>
defaultfloat      = 0
defaultDouble (float64) = 5.500000

MAX float32      = 3.4028234663852886e+38
MIN float32      = 1.401298464324817e-45

MAX float64      = 1.7976931348623157e+308
MIN float64      = 5e-324

Змінні різних типів:
intVar      = 42 (int)
floatVar    = 3.14159 (float64)
boolVar     = true (bool)
stringVar   = Hello, Go! (string)
complexVar  = (2+3i) (complex128)

Process finished with the exit code 0

```

Рис. 8. Завдання 8.

		<i>Рижсенко Я.В</i>		
		<i>Петросян Р.В.</i>		
Змн.	Арк.	№ докум.	Підпис	Дата

## Завдання 9. Пояснити результати операцій

Листинг програми:

```
package main

import "fmt"

func main() {
    // Створення булевих змінних
    var first, second bool // за замовчуванням false
    var third bool = true // явно true
    fourth := !third // NOT third → false
    var fifth = true // короткий запис

    fmt.Println("Булеві змінні:")
    fmt.Println("first = ", first, " // false, за замовчуванням")
    fmt.Println("second = ", second, " // false, за замовчуванням")
    fmt.Println("third = ", third, " // true, явно задано")
    fmt.Println("fourth = ", fourth, " // !true → false")
    fmt.Println("fifth = ", fifth, " // true, короткий запис\n")

    // Логічне заперечення (!)
    fmt.Println("Запереченння (!):")
    fmt.Println("!true = ", !true, " // false")
    fmt.Println("!false = ", !false, " // true\n")

    // Логічне AND (&&)
    fmt.Println("Логічне AND (&&):")
    fmt.Println("true && true = ", true && true, " // обидва true → true")
    fmt.Println("true && false = ", true && false, " // один false → false")
    fmt.Println("false && false = ", false && false, " // обидва false → false\n")

    // Логічне OR (||)
    fmt.Println("Логічне OR (||):")
    fmt.Println("true || true = ", true || true, " // хоча б один true → true")
    fmt.Println("true || false = ", true || false, " // хоча б один true → true")
    fmt.Println("false || false = ", false || false, " // обидва false → false\n")

    // Порівняння чисел
    fmt.Println("Порівняння чисел:")
    fmt.Println("2 < 3 = ", 2 < 3, " // 2 менше 3 → true")
    fmt.Println("2 > 3 = ", 2 > 3, " // 2 не більше 3 → false")
    fmt.Println("3 < 3 = ", 3 < 3, " // 3 не менше 3 → false")
    fmt.Println("3 <= 3 = ", 3 <= 3, " // 3 менше або дорівнює 3 → true")
    fmt.Println("3 > 3 = ", 3 > 3, " // 3 не більше 3 → false")
    fmt.Println("3 >= 3 = ", 3 >= 3, " // 3 більше або дорівнює 3 → true")
    fmt.Println("2 == 3 = ", 2 == 3, " // 2 не дорівнює 3 → false")
    fmt.Println("3 == 3 = ", 3 == 3, " // 3 дорівнює 3 → true")
    fmt.Println("2 != 3 = ", 2 != 3, " // 2 не дорівнює 3 → true")
    fmt.Println("3 != 3 = ", 3 != 3, " // 3 дорівнює 3 → false")
}
```

		Рижсенко Я.В		
		Петросян Р.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

Результат виконання:

```
<4 go setup calls>
Булеві змінні:
first = false // false, за замовчуванням
second = false // false, за замовчуванням
third = true // true, явно задано
fourth = false // !true → false
fifth = true // true, короткий запис

Заперечення (!):
!true = false // false
!false = true // true

Логічне AND (&&):
true && true = true // обидва true → true
true && false = false // один false → false
false && false = false // обидва false → false

Логічне OR (||):
true || true = true // хоча б один true → true
true || false = true // хоча б один true → true
false || false = false // обидва false → false

Порівняння чисел:
2 < 3 = true // 2 менше 3 → true
2 > 3 = false // 2 не більше 3 → false
3 < 3 = false // 3 не менше 3 → false
3 <= 3 = true // 3 менше або дорівнює 3 → true
3 > 3 = false // 3 не більше 3 → false
3 >= 3 = true // 3 більше або дорівнює 3 → true
2 == 3 = false // 2 не дорівнює 3 → false
3 == 3 = true // 3 дорівнює 3 → true
2 != 3 = true // 2 не дорівнює 3 → true
3 != 3 = false // 3 дорівнює 3 → false

Process finished with the exit code 0
```

Рис. 9. Завдання 9.

## Завдання 10.

1. Виведення української літери 'Ї'

2. Пояснення типу "rune"

Листинг програми:

```
package main

import "fmt"

func main() {
    var chartype int8 = 'R'
    fmt.Printf("Code '%c' - %d\n", chartype, chartype)
    // Виводимо: Code 'R' - 82
    // 82 - це код символу 'R' в ASCII
```

		Рижсенко Я.В		
		Петросян Р.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

// 1. Виведення української літери 'Ї'
var ukrainianRune rune = 'Ї'
fmt.Printf("Code '%c' - %d\n", ukrainianRune, ukrainianRune)
// Виведе: Code 'Ї' - 1025
// 1025 – це Unicode код для 'Ї'

// 2. Пояснення типу "rune"
/*
    rune – це int32, який використовується для зберігання одного
    Unicode-символу. Це дозволяє працювати з літерами будь-яких мов,
    включно з українськими, китайськими, арабськими тощо.
    Змінні типу rune зручно застосовувати для роботи з текстом у форматі
Unicode.
*/
}

```

Результат виконання:

```

><4 go setup calls>
Code 'R' - 82
Code 'Ї' - 1031

Process finished with the exit code 0

```

Рис. 10. Завдання 10.

**Посилання на репозиторій** - <https://github.com/JanRizhenko/GoLang-sigma-practice>

**Висновок:** у ході роботи було розглянуто основні властивості базових типів даних мови Go та отримано практичне розуміння їх використання. Ознайомлення з простою структурою програми та базовими конструкціями дозволило закріпити навички оголошення змінних, виконання простих операцій та обробки значень різних типів. Практична частина сприяла формуванню початкового розуміння принципів роботи Go, що створює основу для подальшого вивчення мови та розробки складніших програм.

		Рижсенко Я.В			ДУ «Житомирська політехніка».25.121.20.000 – Пр1	Арк.
		Петросян Р.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		12