

ЛАБОРАТОРНА РОБОТА № 4
Робота з графічним інтерфейсом. Застосування
компонентів Label, CheckBox, ComboBox, Entry, Button

Мета: ознайомитися з основними графічними компонентами. Отримати практичні навички роботи з графічним інтерфейсом, використовувати компоненти Label, CheckBox, ComboBox тощо.

Хід роботи:

Завдання 1. Написати програму (у двох варіантах) для обчислення вартості склопакета в залежності від введених і обраних даних.

Ціни за 1 см² склопакета:

- Однокамерний, дерев'яний – 2.5 грн;
- Двокамерний, дерев'яний – 3 грн;
- Однокамерний, металевий – 0.5 грн;
- Двокамерний, металевий – 1 грн;
- Однокамерний, металопластиковий – 1.5 грн;
- Двокамерний, металопластиковий - 2 грн;
- Вартість підвіконня – 350 грн.

Перший варіант – реалізувати розрахунок з використанням функцій мовою Go.
Другий варіант – реалізувати розрахунок з використанням функцій мовою С, для яких створити функції-обгортки мовою Go.

Листинг програми (Варіант 1):

```
package main

import (
    "fmt"
    "strconv"

    "fyne.io/fyne/v2"
    "fyne.io/fyne/v2/app"
    "fyne.io/fyne/v2/container"
    "fyne.io/fyne/v2/widget"
)

func calcPrice(width, height float64, chambers int, material string, hasSill bool)
```

Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25.121.20.000 – Пр4		
Розроб.		Риженко Я.В.					
Перевір.		Петросян Р.В.					
Керівник							
Н. контр.							
Зав. каф.							
Звіт з лабораторної роботи					Lім.	Арк.	Аркушів
						1	9
					ФІКТ Гр. ІПЗ-23-1		

```

float64 {
    area := width * height

    priceTable := map[string]map[int]float64{
        "Дерев'яний": {
            1: 2.5,
            2: 3,
        },
        "Металевий": {
            1: 0.5,
            2: 1,
        },
        "Металопластиковий": {
            1: 1.5,
            2: 2,
        },
    }
}

price := area * priceTable[material][chambers]

if hasSill {
    price += 350
}

return price
}

func main() {
    a := app.New()
    w := a.NewWindow("Розрахунок склопакета")
    w.Resize(fyne.NewSize(500, 350))

    // Підписи та поля введення
    widthLabel := widgetNewLabel("Ширина (см) :")
    widthEntry := widget.NewEntry()
    widthEntry.SetPlaceholder("Введіть ширину склопакета в см")

    heightLabel := widgetNewLabel("Висота (см) :")
    heightEntry := widget.NewEntry()
    heightEntry.SetPlaceholder("Введіть висоту склопакета в см")

    chambersLabel := widgetNewLabel("Кількість камер:")
    chambers := widget.NewSelect([]string{"Однокамерний", "Двокамерний"}, func(string) {})

    chambers.SetSelected("Однокамерний")

    materialLabel := widgetNewLabel("Матеріал рами:")
    material := widget.NewSelect([]string{
        "Дерев'яний",
        "Металевий",
        "Металопластиковий",
    }, func(string) {})

    material.SetSelected("Дерев'яний")

    sill := widget.NewCheck("Додати підвіконня (350 грн)", nil)

    result := widget.NewLabel("")

    button := widget.NewButton("Розрахувати", func() {
        wVal, _ := strconv.ParseFloat(widthEntry.Text, 64)
        hVal, _ := strconv.ParseFloat(heightEntry.Text, 64)
    })
}

```

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

```
    var ch int
    if chambers.Selected == "Однокамерный" {
        ch = 1
    } else {
        ch = 2
    }

    mat := material.Selected

    total := calcPrice(wVal, hVal, ch, mat, sill.Checked)
    result.SetText(fmt.Sprintf("Вартість: %.2f грн", total))
}

w.SetContent(container.NewVBox(
    widthLabel,
    widthEntry,
    heightLabel,
    heightEntry,
    chambersLabel,
    chambers,
    materialLabel,
    material,
    sill,
    button,
    result,
))
w.ShowAndRun()
}
```

Результат виконання:

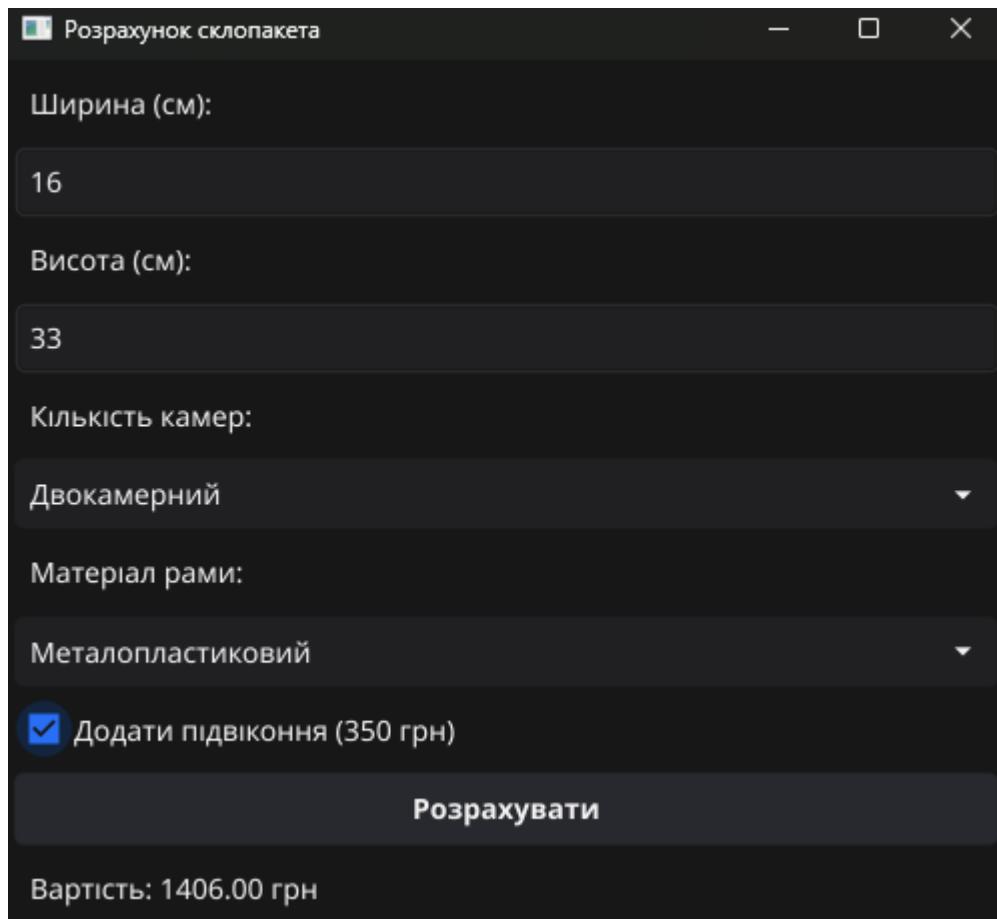


Рис. 1. Завдання 1, варіант 1.

		<i>Риженко Я.В</i>				
		<i>Петросян Р.В.</i>				
Змн.	Арк.	№ докум.	<i>Підпись</i>	<i>Дата</i>	ДУ «Житомирська політехніка».25.121.20.000 – Пр4	Арк. 3

Листинг програми (Варіант 2):

```
#include <stdbool.h>

double calc(double width, double height, int chambers, int material, bool sill) {
    double area = width * height;
    double price = 0;

    double table[3][3] = {
        {0, 2.5, 3},
        {0, 0.5, 1},
        {0, 1.5, 2}
    };

    price = area * table[material][chambers];

    if (sill) price += 350;

    return price;
}
```

```
package main

/*
#include <stdbool.h>
double calc(double width, double height, int chambers, int material, bool sill);
*/
import "C"

import (
    "fmt"
    "strconv"

    "fyne.io/fyne/v2"
    "fyne.io/fyne/v2/app"
    "fyne.io/fyne/v2/container"
    "fyne.io/fyne/v2/widget"
)

func main() {
    a := app.New()
    w := a.NewWindow("Розрахунок склопакета")
    w.Resize(fyne.NewSize(500, 350))

    widthLabel := widget.NewLabel("Ширина (см):")
    widthEntry := widget.NewEntry()
    widthEntry.SetPlaceHolder("Введіть ширину в см")

    heightLabel := widget.NewLabel("Висота (см):")
    heightEntry := widget.NewEntry()
    heightEntry.SetPlaceHolder("Введіть висоту в см")

    chambersLabel := widget.NewLabel("Кількість камер:")
    chambers := widget.NewSelect([]string{"Однокамерний", "Двокамерний"}, func(string) {})
    chambers.SetSelected("Однокамерний")

    materialLabel := widget.NewLabel("Матеріал рами:")
    material := widget.NewSelect([]string{
        "Дерев'яний",
        "Металевий",
        "Металопластиковий",
    })
}
```

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

```

    }, func(string) {})
material.SetSelected("Дерев'яний")

sill := widget.NewCheck("Додати підвіконня (350 грн)", nil)

result := widgetNewLabel("")

button := widget.NewButton("Розрахувати", func() {
    wVal, _ := strconv.ParseFloat(widthEntry.Text, 64)
    hVal, _ := strconv.ParseFloat(heightEntry.Text, 64)

    var ch int
    if chambers.Selected == "Однокамерний" {
        ch = 1
    } else {
        ch = 2
    }

    var mat int
    switch material.Selected {
    case "Дерев'яний":
        mat = 0
    case "Металевий":
        mat = 1
    case "Металопластиковий":
        mat = 2
    }

    total := C.calc(
        C.double(wVal),
        C.double(hVal),
        C.int(ch),
        C.int(mat),
        C.bool(sill.Checked),
    )

    result.SetText(fmt.Sprintf("Вартість: %.2f грн", float64(total)))
})

w.SetContent(container.NewVBox(
    widthLabel,
    widthEntry,
    heightLabel,
    heightEntry,
    chambersLabel,
    chambers,
    materialLabel,
    material,
    sill,
    button,
    result,
))
w.ShowAndRun()
}

```

		<i>Риженко Я.В</i>			<i>ДУ «Житомирська політехніка».25.121.20.000 – Пр4</i>	Арк.
		<i>Петросян Р.В.</i>				
Змн.	Арк.	№ докум.	Підпис	Дата		5

Результат виконання:

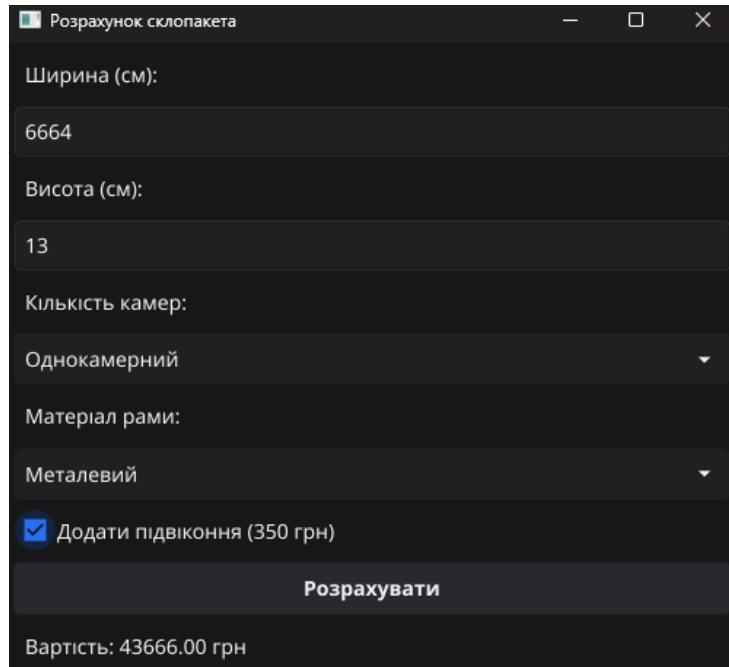


Рис. 2. Завдання 1, варіант 2.

Завдання 2. Аналогічно створити проект для обчислення вартості замовлення туру на відпочинок.

Ціни за 1 день подорожі:

Болгарія, літо – \$100;

Болгарія, зима – \$150;

Німеччина, літо – \$160;

Німеччина, зима – \$200;

Польща, літо – \$120;

Польща, зима – \$180;

Вартість індивідуального гіда – \$50 в день на всю кількість путівок;

Врахувати вартість націнки в розмірі 20% за проживання в номері люкс.

Листинг програми:

```
package main

import (
    "fmt"
    "strconv"

    "fyne.io/fyne/v2"
    "fyne.io/fyne/v2/app"
    "fyne.io/fyne/v2/container"
    "fyne.io/fyne/v2/widget"
)
```

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

```

func calcTourPrice(days int, country, season string, guide bool, luxury bool)
float64 {
    priceTable := map[string]map[string]float64{
        "Болгарія": {
            "літо": 100,
            "зима": 150,
        },
        "Німеччина": {
            "літо": 160,
            "зима": 200,
        },
        "Польща": {
            "літо": 120,
            "зима": 180,
        },
    }
}

price := float64(days) * priceTable[country][season]

if guide {
    price += float64(days) * 50
}

if luxury {
    price *= 1.2
}

return price
}

func main() {
    a := app.New()
    w := a.NewWindow("Розрахунок туру")
    w.Resize(fyne.NewSize(500, 350))

    daysLabel := widgetNewLabel("Кількість днів:")
    daysEntry := widgetNewEntry()
    daysEntry.SetPlaceHolder("Введіть кількість днів подорожі")

    countryLabel := widgetNewLabel("Країна подорожі:")
    country := widgetNewSelect([]string{"Болгарія", "Німеччина", "Польща"}, func(string) {})
    country.SetSelected("Болгарія")

    seasonLabel := widgetNewLabel("Сезон поїздки:")
    season := widgetNewSelect([]string{"літо", "зима"}, func(string) {})
    season.SetSelected("літо")

    guide := widgetNewCheck("Індивідуальний гід (+50$/день)", nil)
    luxury := widgetNewCheck("Номер люкс (+20%)", nil)

    result := widgetNewLabel("")

    button := widgetNewButton("Розрахувати", func() {
        days, _ := strconv.Atoi(daysEntry.Text)
        c := country.Selected
        s := season.Selected
        g := guide.Checked
        l := luxury.Checked

        total := calcTourPrice(days, c, s, g, l)
        result.SetText(fmt.Sprintf("Вартість: %.2f $", total))
    })
}

```

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

```

    }

    w.SetContent(container.NewVBox(
        daysLabel,
        daysEntry,
        countryLabel,
        country,
        seasonLabel,
        season,
        guide,
        luxury,
        button,
        result,
    ) )

    w.ShowAndRun()
}

func GenerateIntSequence(seed uint64) []uint64 {
    x := seed
    seq := make([]uint64, K)

    for i := 0; i < K; i++ {
        x = (a*x + c) % m
        seq[i] = x % 100
    }
    return seq
}

func GenerateFloatSequence(seed uint64) []float64 {
    x := seed
    seq := make([]float64, K)

    for i := 0; i < K; i++ {
        x = (a*x + c) % m
        seq[i] = (float64(x) / float64(m)) * 100
    }
    return seq
}

func Analyze(seq []uint64) {
    freq := make([]int, 100)

    for _, v := range seq {
        freq[v]++
    }
    prob := make([]float64, 100)
    for i := 0; i < 100; i++ {
        prob[i] = float64(freq[i]) / float64(len(seq))
    }

    var mean float64
    for i := 0; i < 100; i++ {
        mean += float64(i) * prob[i]
    }

    var dispersion float64
    for i := 0; i < 100; i++ {
        dispersion += math.Pow(float64(i)-mean, 2) * prob[i]
    }

    std := math.Sqrt(dispersion)
}

```

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

```

fmt.Println("==== Результати аналізу ===")
fmt.Printf("Математичне сподівання: %.6f\n", mean)
fmt.Printf("Дисперсія: %.6f\n", dispersion)
fmt.Printf("СКВ: %.6f\n\n", std)

fmt.Println("==== Ймовірності появи значень ===")
for i := 0; i < 100; i++ {
    fmt.Printf("%2d : P = %.6f\n", i, prob[i])
}
}

func main() {
    seq := GenerateIntSequence(1)
    Analyze(seq)

    floatSeq := GenerateFloatSequence(1)
    fmt.Println("Перші 5 дійсних значень [0;100]:", floatSeq[:5])
}

```

Результат виконання:

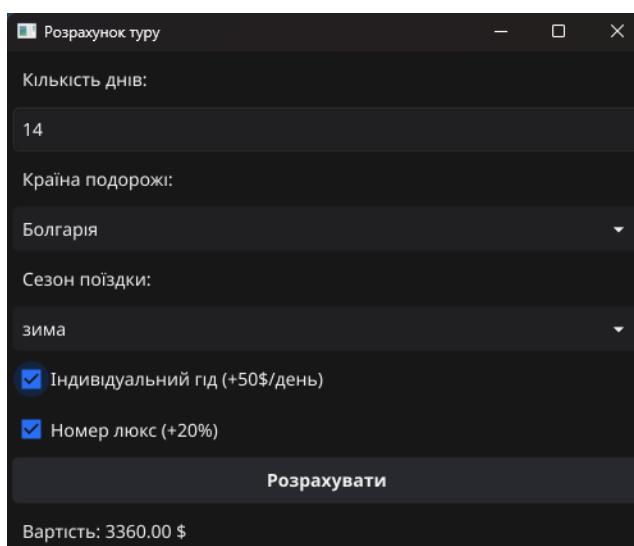


Рис. 3. Завдання 2.

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

Висновок: В ході виконання роботи було отримано практичні навички створення графічного інтерфейсу користувача з використанням мови Go та бібліотеки Fyne. Було освоєно застосування основних компонентів, таких як Label для відображення текстової інформації, Entry для введення даних користувачем, CheckBox для вибору опцій, ComboBox (Select) для вибору зі списку та Button для запуску обчислень або дій. Створені програми демонструють інтерактивну роботу з користувачем, відображення результатів обчислень та зміни стану інтерфейсу залежно від введених даних. Практичне застосування цих компонентів дозволило зрозуміти принципи побудови інтуїтивно зрозумілого інтерфейсу та взаємодії користувача з програмою, що є важливим для розробки сучасних GUI-додатків.

		Рижсенко Я.В			ДУ «Житомирська політехніка».25.121.20.000 – Пр4	Арк.
		Петросян Р.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		9