

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**  
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**  
**ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

**КАТЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ**

## **Дискретна математика**

### **Лабораторна робота №2**

**«Бінарні відношення та їх основні властивості, операції над відношеннями»**

**Виконав:**  
**студент групи ІО-32**  
**Крадожон М. Р.**  
**Номер у списку групи: 16**  
**Перевірив:**  
**Пономаренко А. М.**

## Лабораторна робота №2

**Тема:** «Бінарні відношення та їх основні властивості, операції над відношеннями».

**Мета:** вивчити основні властивості бінарних відношень та оволодіти операціями над бінарними відношеннями.

### Загальне завдання:

1. Написати в окремому модулі функцію для формування несуперечливих бінарних відношень.
2. Написати в окремому модулі функції виконання логічних операцій над бінарними відношеннями.
3. Пояснити правило формування несуперечливих відношень відповідно до Вашого варіанту.

### Теоретичні основи:

Бінарним (або *двомісним*) відношенням  $R$  називають підмножину впорядкованих пар, тобто множину, кожен елемент якої є впорядкованою парою. Якщо  $R$  є деяким відношенням, це записують як  $x, y \in R$  або  $xRy$ .

Один з типів відношень – це множина всіх таких пар  $x, y$ , що  $x$  є елементом деякої фіксованої множини  $X$ , а  $y$  – елементом деякої фіксованої множини  $Y$ . Таке відношення називають *прямим* або *декартовим добутком*.

Бінарним відношенням  $R$  називають підмножину пар  $x, y \in X \times Y$  прямого добутку  $X \times Y$ , тобто  $R \subseteq X \times Y$ .

### **Операції над відношеннями:**

Оскільки відношення на множині  $A$  задають підмножинами  $R \subseteq A \times A$ , то для них визначні ті ж операції, що й над множинами, а саме:

1. **Об'єднання:**  $R_1 \cup R_2 = \{ \langle a, b \rangle \mid \langle a, b \rangle \in R_1 \text{ або } \langle a, b \rangle \in R_2 \}$ .
2. **Перетин:**  $R_1 \cap R_2 = \{ \langle a, b \rangle \mid \langle a, b \rangle \in R_1 \text{ і } \langle a, b \rangle \in R_2 \}$ .
3. **Різниця:**  $R_1 \setminus R_2 = \{ \langle a, b \rangle \mid \langle a, b \rangle \in R_1 \text{ і } \langle a, b \rangle \notin R_2 \}$ .
4. **Доповнення:**  $\overline{R} = U \setminus R$ , де  $U = A \times A$ .

Крім того, необхідно визначити інші операції над бінарними відношеннями.

#### **5. Обернене відношення $R^{-1}$ .**

Якщо  $\langle a, b \rangle \in R$  – відношення, то відношення  $R^{-1}$  називають **оберненим відношенням** до даного відношення  $R$  тоді й тільки тоді, коли  $R^{-1} = \{ \langle b, a \rangle \mid \langle a, b \rangle \in R \}$ .

## Індивідуальне завдання

### Загальний порядок виконання лабораторної роботи:

Етап 1. Створити програму, яка коректно формує відношення у відповідності з варіантом завдання та виконує операції над цими відношеннями.

Етап 2. Ввести елементи множин А та В . Наприклад: А = {Антоніна, Оксана, Галина, Ольга, Світлана, Петро, Тетяна, Іван, Катерина, Олег}, В = {Борис, Василь, Максим, Ольга, Тетяна, Іван, Аркадій, Артем, Оксана, Петро}.

Етап 3. Задати програмно відношення S і R між елементами множин А і В .

Наприклад, використовуючи варіант: aSb, якщо а сестра b. aRb, якщо а дружина b

Етап 4. Виконати програмно перевірку коректності задавання відношень.

Наприклад:

$S = \{\langle 1,1 \rangle, \langle 2,2 \rangle, \langle 3,3 \rangle, \langle 4,6 \rangle, \langle 5,5 \rangle, \langle 7,7 \rangle, \langle 9,9 \rangle\}$  – елементи відношення коректні,

$R = \{\langle 1,2 \rangle, \langle 2,1 \rangle, \langle 3,8 \rangle, \langle 4,7 \rangle, \langle 5,3 \rangle, \langle 7,6 \rangle, \langle 9,10 \rangle\}$  – елементи відношення коректні.

Приклади некоректних елементів відношень:

$\langle 1,1 \rangle \notin R$  – Антоніна не може бути дружиною Бориса, оскільки вона – його сестра.

$\langle 3,4 \rangle \notin R$  – Ольга – особа жіночої статі, тому не може бути дружиною Галини.

$\langle 6,9 \rangle \notin R$  – Петро – особа чоловічої статі, тому не може бути дружиною Оксани.

$\langle 4,4 \rangle \notin S$  – Ольга не може бути сестрою сама собі, і т. д.

### Варіант 19:

19	aSb, якщо а внучка b.	aRb, якщо а дружина b.
----	-----------------------	------------------------

### Роздруківка коду:

#### data.py

# For fourth window

```
coord_dict1 = {('Валерія', 'Олексій'): (210, 55, 410, 175),  
               ('Анастасія', 'Юрій'): (410, 55, 210, 175),  
               ('Вікторія', 'Іван'): (110, 55, 10, 175),  
               ('Марія', 'Володимир'): (310, 55, 310, 175),  
               ('Вікторія', 'Юрій'): (110, 55, 210, 175),  
               ('Анна', 'Іван'): (10, 55, 10, 175),  
               ('Анна', 'Олег'): (10, 55, 110, 175),  
               ('Валерія', 'Юрій'): (210, 55, 210, 175)}
```

```
coord_dict2 = {('Валерія', 'Олексій'): (710, 55, 910, 175),  
               ('Анастасія', 'Юрій'): (910, 55, 710, 175),  
               ('Вікторія', 'Іван'): (610, 55, 510, 175),  
               ('Марія', 'Володимир'): (810, 55, 810, 175),  
               ('Вікторія', 'Юрій'): (610, 55, 710, 175),  
               ('Анна', 'Іван'): (510, 55, 510, 175),  
               ('Анна', 'Олег'): (510, 55, 610, 175),
```

```

('Валерія', 'Юрій'): (710, 55, 710, 175)}

coord_dict3 = {'Валерія', 'Олексій'): (205, 305, 405, 425),
               ('Анастасія', 'Юрій'): (405, 305, 205, 425),
               ('Вікторія', 'Іван'): (105, 305, 5, 425),
               ('Марія', 'Володимир'): (305, 305, 305, 425),
               ('Вікторія', 'Юрій'): (105, 305, 205, 425),
               ('Анна', 'Іван'): (5, 305, 5, 425),
               ('Анна', 'Олег'): (5, 305, 105, 425),
               ('Валерія', 'Юрій'): (205, 305, 205, 425)}

coord_dict4 = {'Валерія', 'Олексій'): (710, 305, 910, 425),
               ('Анастасія', 'Юрій'): (910, 305, 710, 425),
               ('Вікторія', 'Іван'): (610, 305, 510, 425),
               ('Марія', 'Володимир'): (810, 305, 810, 425),
               ('Вікторія', 'Юрій'): (610, 305, 710, 425),
               ('Анна', 'Іван'): (510, 305, 510, 425),
               ('Анна', 'Олег'): (510, 305, 610, 425),
               ('Валерія', 'Юрій'): (710, 305, 710, 425)}

coord_dict5 = {'Валерія', 'Олексій'): (210, 555, 410, 675),
               ('Анастасія', 'Юрій'): (410, 555, 210, 675),
               ('Вікторія', 'Іван'): (110, 555, 10, 675),
               ('Марія', 'Володимир'): (310, 555, 310, 675),
               ('Вікторія', 'Юрій'): (110, 555, 210, 675),
               ('Анна', 'Іван'): (10, 555, 10, 675),
               ('Анна', 'Олег'): (10, 555, 110, 675),
               ('Валерія', 'Юрій'): (210, 555, 210, 675)}

# For third window
coord_dict_for_set_s = {'Марія', 'Володимир'): (705, 55, 705, 175),
                        ('Вікторія', 'Юрій'): (505, 55, 605, 175),
                        ('Анна', 'Іван'): (405, 55, 405, 175),
                        ('Анна', 'Олег'): (405, 55, 505, 175),
                        ('Валерія', 'Юрій'): (605, 55, 605, 175)}

coord_dict_for_set_r = {'Валерія', 'Олексій'): (605, 305, 805, 425),
                        ('Анастасія', 'Юрій'): (805, 305, 605, 425),
                        ('Вікторія', 'Іван'): (505, 305, 405, 425)}

relationships_for_granddaughter = {
    ('Марія', 'Володимир'): ('Марія', 'Володимир'),
    ('Валерія', 'Юрій'): ('Валерія', 'Юрій'),
    ('Вікторія', 'Юрій'): ('Вікторія', 'Юрій'),
    ('Анна', 'Іван'): ('Анна', 'Іван'),
    ('Анна', 'Олег'): ('Анна', 'Олег')
}

relationships_for_wife = {
    ('Анастасія', 'Юрій'): ('Анастасія', 'Юрій'),
    ('Валерія', 'Олексій'): ('Валерія', 'Олексій'),
    ('Вікторія', 'Іван'): ('Вікторія', 'Іван')
}

# General
set_u = {'Анастасія', 'Юрій'),
         ('Валерія', 'Олексій'),
         ('Вікторія', 'Іван'),
         ('Анна', 'Олег'),
         ('Вікторія', 'Юрій'),

```

```
('Валерія', 'Юрій'),  
( 'Марія', 'Володимир'),  
( 'Анна', 'Іван'),  
( 'Катерина', 'Микита')}]
```

```
womenlist = ["Анна", "Вікторія", "Валерія", "Марія", "Анастасія"]  
menlist = ["Іван", "Олег", "Юрій", "Володимир", "Олексій"]
```

## main.py

```
import data  
import os  
from tkinter import *  
from data import *  
from sencitive_data import *  
  
def second_window():  
    def set_adder():  
        selected_listbox = women_listbox if set_num.get() == 1 else men_listbox  
        selected_set = set_a if set_num.get() == 1 else set_b  
        print(selected_listbox.get(ANCHOR))  
        selected_set.add(selected_listbox.get(ANCHOR))  
  
    def save_set_to_file(set_to_save, file_name):  
        with open(file_name, "w") as f:  
            f.write("\n".join(set_to_save))  
  
    def read_set_from_file(set_to_update, file_name):  
        with open(file_name, "r") as f:  
            set_to_update.clear()  
            temp = f.read().strip().split("\n")  
            set_to_update.update(temp)  
  
            print(temp)  
            print(set_to_update)  
  
    def clear_set_and_file(set_to_clear, file_name):  
        set_to_clear.clear()  
        os.remove(file_name)  
  
    def create_listbox(x_pos, y_pos, items):  
        listbox = Listbox(root2)  
        listbox.place(x=x_pos, y=y_pos)  
        listbox.insert(END, *items)  
        return listbox  
  
    root2 = Tk()  
    root2.title("Вікно 2")  
    root2.geometry("500x350")  
  
    set_num = IntVar(root2)  
    women_listbox = create_listbox(50, 10, items=womenlist)  
    men_listbox = create_listbox(300, 10, items=menlist)  
  
    Radiobutton(root2, text="Множина А", variable=set_num, value=1).place(x=60, y=180)  
    Radiobutton(root2, text="Множина В", variable=set_num, value=2).place(x=320, y=180)  
    Button(root2, width=8, text="Додати", font="Arial 10", command=set_adder).place(x=200, y=120)  
    Button(root2, width=15, text="Зберегти А у файл", font="Arial 10",  
            command=lambda: save_set_to_file(set_a, "Adata.txt")).place(x=0 + 50, y=240)  
    Button(root2, width=15, text="Зберегти В у файл", font="Arial 10",
```

```

        command=lambda: save_set_to_file(set_b, "Bdata.txt")).place(x=0 + 50, y=270)
Button(root2, width=10, text="Зчитати А", font="Arial 10",
        command=lambda: read_set_from_file(set_a, "Adata.txt")).place(x=140 + 50, y=240)
Button(root2, width=10, text="Зчитати В", font="Arial 10",
        command=lambda: read_set_from_file(set_b, "Bdata.txt")).place(x=140 + 50, y=270)
Button(root2, width=10, text="Очистити А", font="Arial 10",
        command=lambda: clear_set_and_file(set_a, "Adata.txt")).place(x=320, y=240)
Button(root2, width=10, text="Очистити В", font="Arial 10",
        command=lambda: clear_set_and_file(set_b, "Bdata.txt")).place(x=320, y=270)

```

```
def third_window():
```

```
def populate_listbox(listbox, some_set, x_position, y_position):
```

```
listbox.place(x=x_position, y=y_position)
```

```
for item in some_set:
```

```
listbox.insert(END, item)
```

```
def create_labels_and_ovals(title, names, x_offset, y_offset):
```

```
Label(root3, text=title, font='Arial 12').place(x=400 + x_offset, y=y_offset)
```

```
for i, name in enumerate(names):
```

```
y_position = 40 + y_offset if y_offset in [150, 400] else 25 + y_offset
```

```
oval_y = 25 + y_offset if y_offset in [150, 400] else 50 + y_offset
```

```
Label(root3, text=name, font='Arial 10').place(x=401 + i * 100 + x_offset, y=y_position)
```

```
canvas.create_oval(401 + i * 100 + x_offset, oval_y, 410 + i * 100 + x_offset, oval_y + 10, fill="black")
```

```
for i in set_r:
```

```
if i in coord_dict_for_set_r:
```

```
canvas.create_line(coord_dict_for_set_r[i], arrow="last")
```

```
for i in set_s:
```

```
if i in coord_dict_for_set_s:
```

```
canvas.create_line(coord_dict_for_set_s[i], arrow="last")
```

```
def relations(letter, relate):
```

```
for condition, elements in relate.items():
```

```
if elements[0] in set_a and elements[1] in set_b:
```

```
letter.add(condition)
```

```
print(letter)
```

```
root3 = Tk()
```

```
root3.title("Вікно 3")
```

```
root3.geometry("1000x600")
```

```
canvas = Canvas(root3, width=1000, height=500)
```

```
canvas.place(x=0, y=0)
```

```
relations(letter=set_s, relate=data.relationships_for_granddaughter)
```

```
relations(letter=set_r, relate=data.relationships_for_wife)
```

```
populate_listbox(Listbox(root3), set_a, 50, 100)
```

```
populate_listbox(Listbox(root3), set_b, 200, 100)
```

```
create_labels_and_ovals('А онука В', womenlist, 0, 0)
```

```
create_labels_and_ovals('А дружина В', womenlist, 0, 250)
```

```
create_labels_and_ovals(None, menlist, 0, 150)
```

```
create_labels_and_ovals(None, menlist, 0, 400)
```

```
def fourth_window():
```

```
def create_labels_and_ovals(title, x, y):
```

```

Label(root4, text=title, font='Arial 12').place(x=x, y=y)
for i, name in enumerate(womenlist):
    Label(root4, text=name, font='Arial 10').place(x=x + i * 100, y=y + 20)
    canvas.create_oval(x + 5 + i * 100, y + 45, x + 15 + i * 100, y + 55, fill="black")
for i, name in enumerate(menlist):
    Label(root4, text=name, font='Arial 10').place(x=x + i * 100, y=y + 190)
    canvas.create_oval(x + 5 + i * 100, y + 175, x + 15 + i * 100, y + 185, fill="black")

root4 = Tk()
root4.title("Вікно 4")
root4.geometry("1000x750")

canvas = Canvas(root4, bg="white", width=1000, height=1000)
canvas.place(x=0, y=0)

set_dict = {
    "rs": (set_r.union(set_s), coord_dict1),
    "rs_intersect": (set_r.intersection(set_s), coord_dict3),
    "r_diff_s": (set_r.difference(set_s), coord_dict2),
    "u_diff_r": (set_u.difference(set_r), coord_dict4),
    "s": (set_s, coord_dict5)
}

for key, (s, coord_dict) in set_dict.items():
    for i in s:
        if i in coord_dict:
            if key == "s":
                canvas.create_line(coord_dict[i], coord_dict[i], arrow="first", arrowshape=(8, 10, 3))
            else:
                canvas.create_line(coord_dict[i], coord_dict[i], arrow="last", arrowshape=(8, 10, 3))

create_labels_and_ovals('R ∪ S', 0, 0)
create_labels_and_ovals('R ∩ S', 0, 250)
create_labels_and_ovals('R \ S', 500, 0)
create_labels_and_ovals('U \ R', 500, 250)
create_labels_and_ovals('S^1', 0, 500)

variant = (senc_data['number_in_list'] + senc_data['group'] % 60) % 30 + 1
set_a, set_b, set_s, set_r = set(), set(), set(), set()

root = Tk()
root.title("Вікно 1")
root.geometry("500x200")

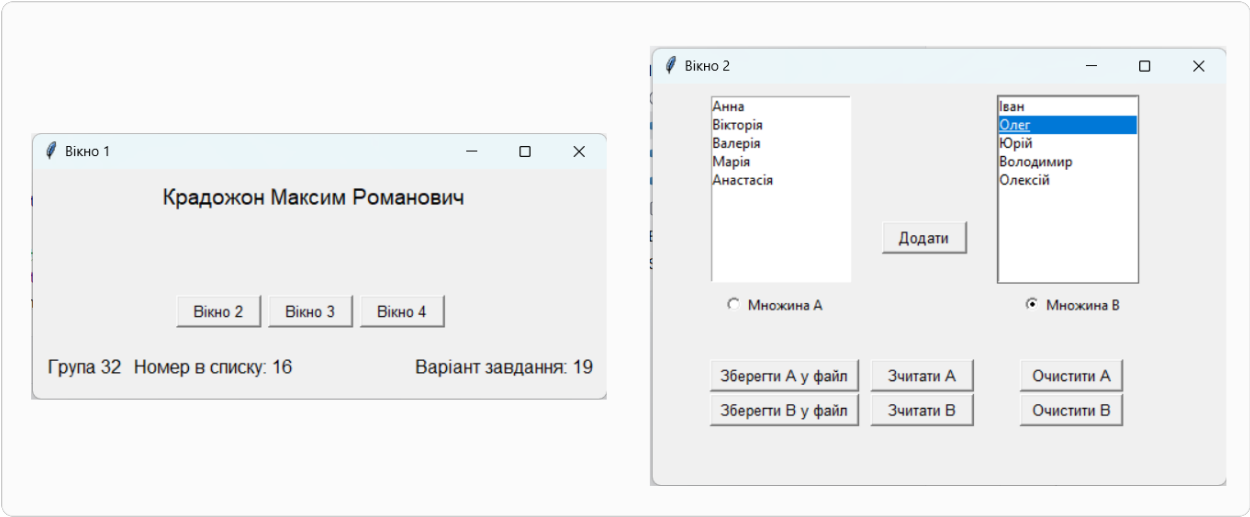
Label(root, text=senc_data['name'], font='Arial 14').place(x=110, y=10)
Label(root, text=f"Група {senc_data['group']}", font='Arial 12').place(x=10, y=160)
Label(root, text=f"Номер в списку: {senc_data['number_in_list']}", font='Arial 12').place(x=85, y=160)
Label(root, text=f"Варіант завдання: {variant}", font='Arial 12').place(x=330, y=160)

Button(root, width=8, text="Вікно 2", font="Arial 10", command=second_window).place(x=125, y=110)
Button(root, width=8, text="Вікно 3", font="Arial 10", command=third_window).place(x=205, y=110)
Button(root, width=8, text="Вікно 4", font="Arial 10", command=fourth_window).place(x=285, y=110)

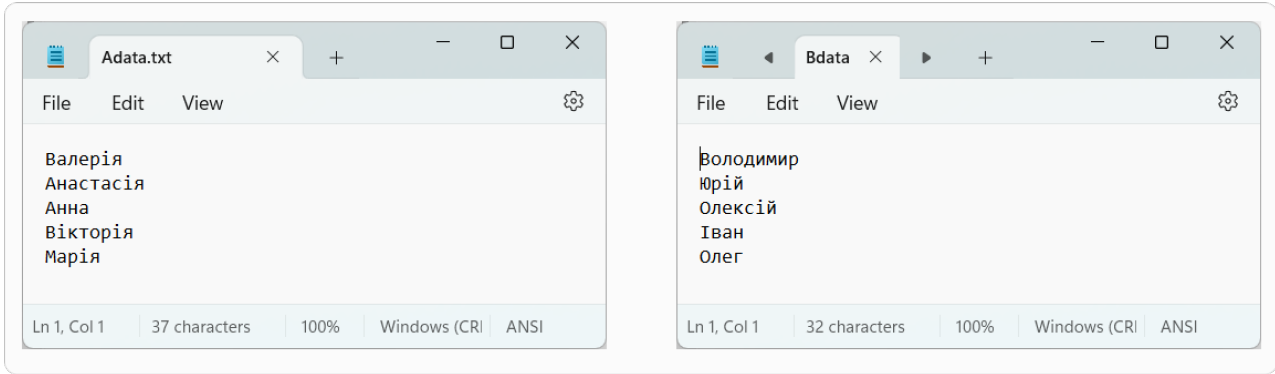
root.mainloop()

```

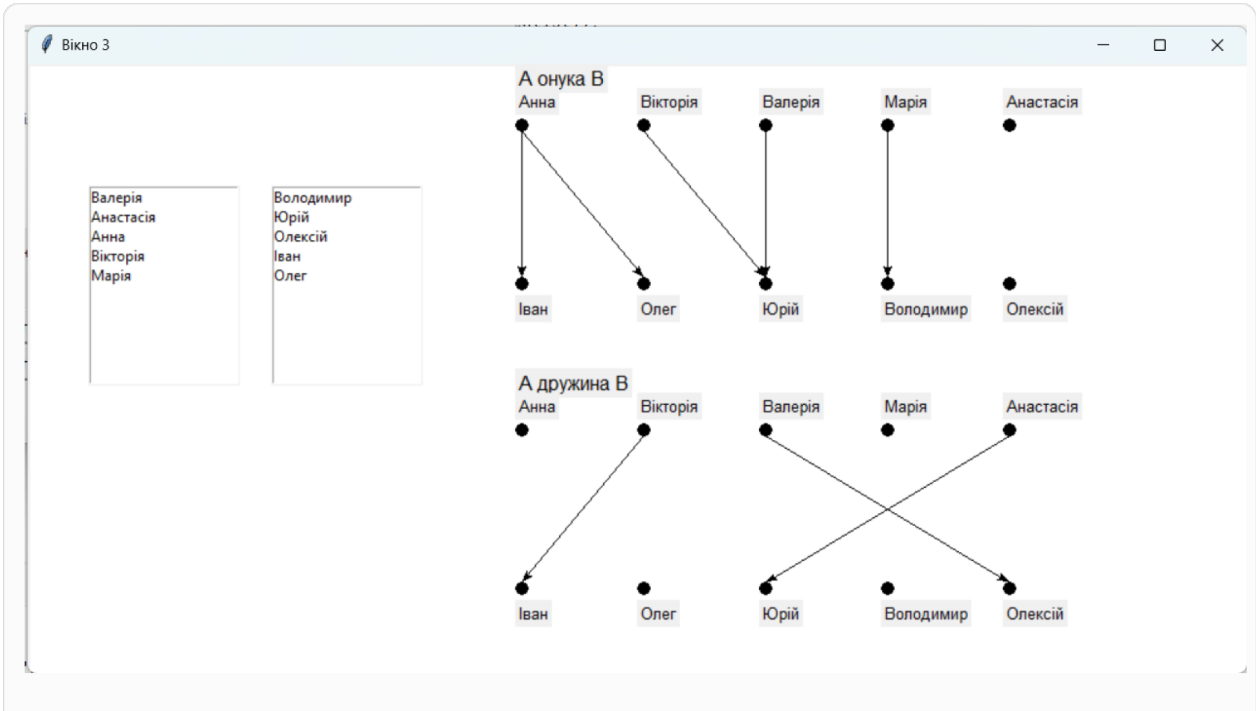
Скриншоти:



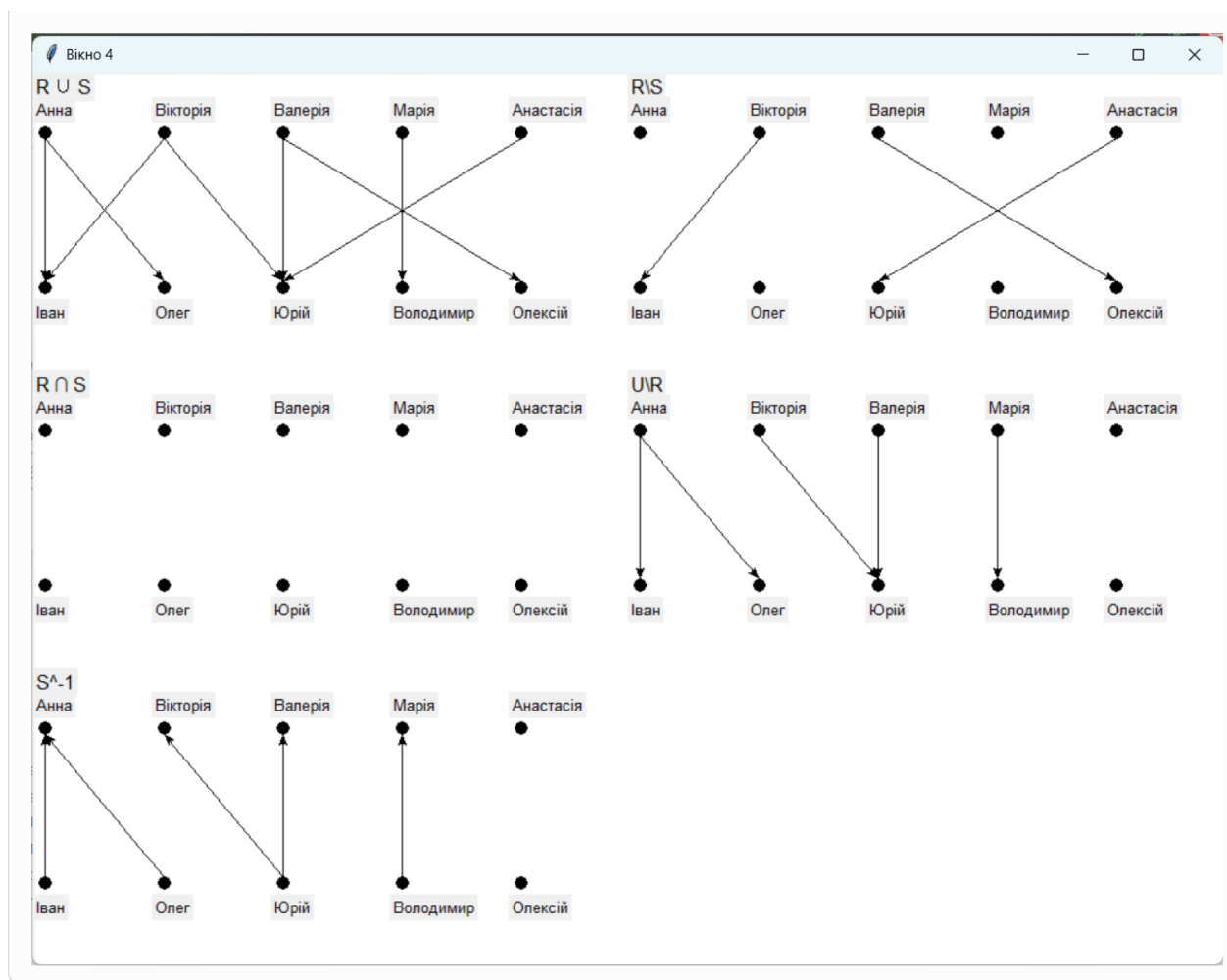
Скриншоти



Section 1







**Висновок:** Виконавши цю лабораторну роботу, я зміг здобути відповідні навички в бінарних відношеннях та їхні основні властивості, операції над відношеннями. Під час виконання лабораторної роботи проблем не виникало, а складність була в структуруванні коду та приведенні його до більш гарного вигляду.