

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки
Дискретна математика
Лабораторна робота №2
«Множини: основні властивості та операції над ними, діаграми Венна»

Виконав:
студент групи ІВ-71
Мазан Ян Владиславович
Залікова книжка:
№7109
Перевірив:
Саверченко Василь Григорович

Київ-2018

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.

Варіант виразу відповідно до індивідуального завдання:

$i = 9$

$G = 71$

$Z = (9 + 71 \bmod 60) \bmod 30 + 1 = 21$

21	aSb, якщо a хрещена мати b.	aRb, якщо a кума b.
----	-----------------------------	---------------------

Теоретичні відомості:

Упорядкована пара предметів – це сукупність, що складається із двох предметів, розташованих у деякому певному порядку.

Бінарним (або *двомісним*) відношенням R називають підмножину впорядкованих пар, тобто множину, кожен елемент якої є впорядкованою парою

Декартовим добутком $X * Y$ множин X і $Y \in$ множина $\{(x, y) \mid x \in X, y \in Y\}$.

Бінарним відношенням R називають підмножину пар $(x, y) \in R$ прямого добутку $X * Y$. У силу визначення бінарних відношень, як **спосіб їх задавання** можуть бути використані будь-які способи задавання множин. Відношення, визначені на скінченних множинах, зазвичай задають списком чи матрицею.

Операції над відношеннями:

1. **Об'єднання:** $R_1 \cup R_2 = \{(a, b) \mid (a, b) \in R_1 \text{ або } (a, b) \in R_2\}$.

2. **Перетин:** $R_1 \cap R_2 = \{(a, b) \mid (a, b) \in R_1 \text{ і } (a, b) \in R_2\}$.

3. **Різниця:** $R_1 \setminus R_2 = \{(a, b) \mid (a, b) \in R_1 \text{ і } (a, b) \notin R_2\}$.

4. **Доповнення:** $\overline{R} = U \setminus R$, де $U = A \times B$.

Крім того, необхідно визначити інші операції над бінарними відношеннями.

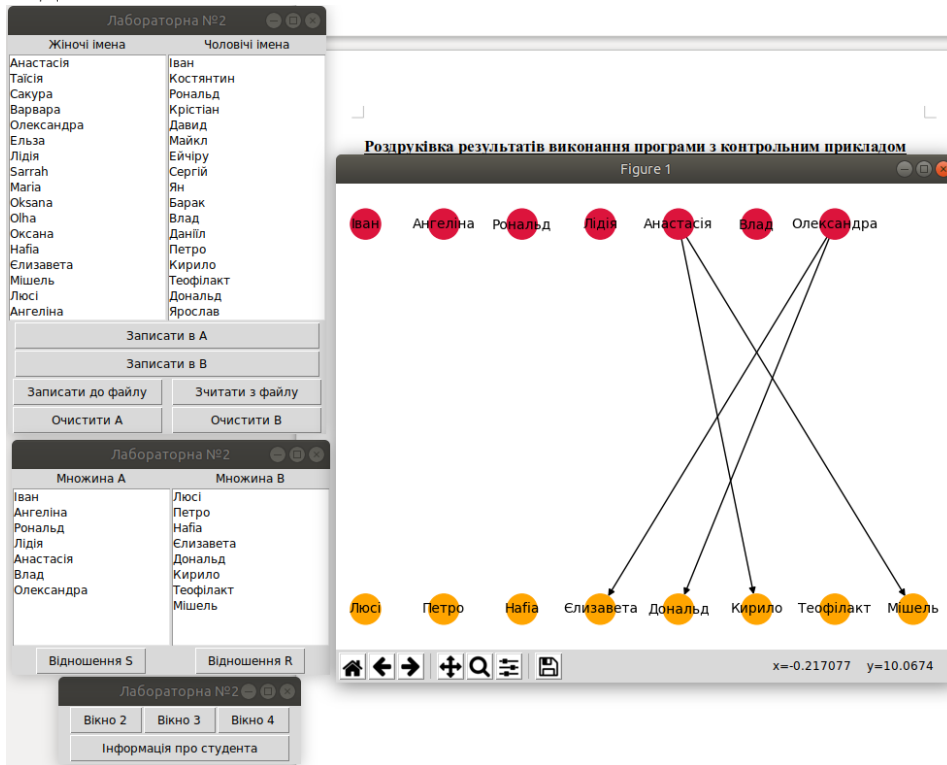
5. **Обернене відношення** R^{-1} .

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

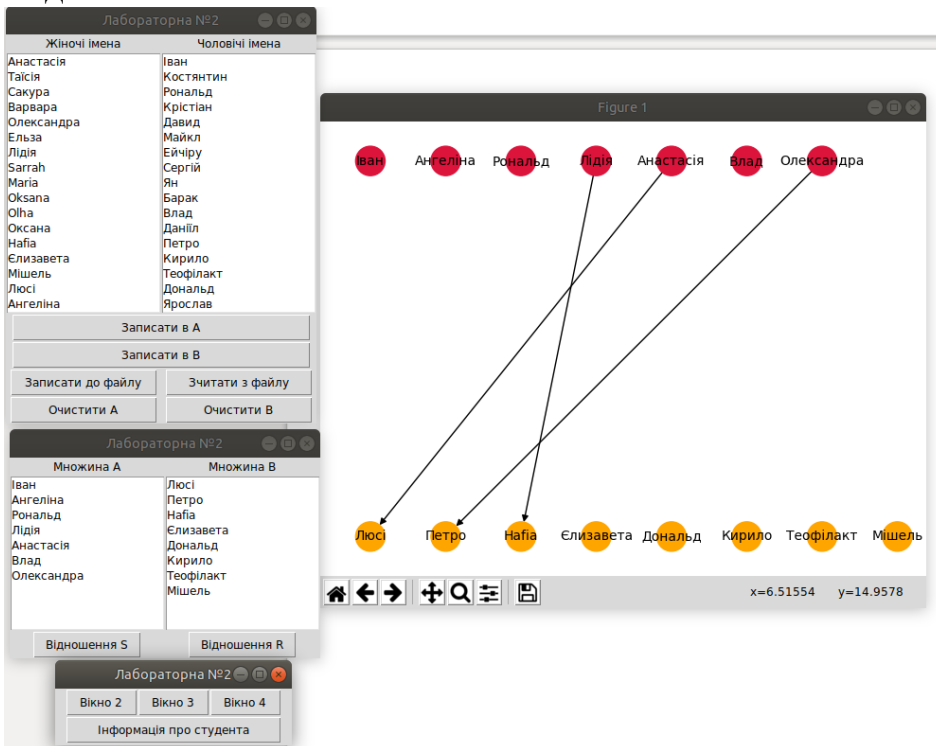
Роздруківка того фрагменту тексту програми, який написаний індивідуально

```
class algorithms:
    def __init__(self, A:"list", B:"list"):
        self.A = A
        self.B = B
        self.universal = self.decart_multiplication()
        self.male_names = names().male_names
        self.fem_names = names().fem_names
        self.A_males = []
        self.A_females = []
        self.relation_S = []
        self.children = []
        self.godmothers = []
        self.godmother()
        self.compadre()
    #basic operations
    def decart_multiplication(self):
        multiplication = [(i,j) for i in self.A for j in self.B]
        return multiplication
    def union(self, relation_1, relation_2):
        return list(set(relation_1).union(set(relation_2)))
    def intersection(self, relation_1, relation_2):
        return list(set(relation_1).intersection(set(relation_2)))
    def difference(self, relation_1, relation_2):
        return list(set(relation_1).difference(set(relation_2)))
    def complement(self, universal, relation):
        return list(set(universal).difference(set(relation)))
    def inverted_relation(self, relation):
        return [(i[1], i[0]) for i in relation]
    #operations according to my variant
    def godmother(self):
        self.A_males = list(set(self.A).intersection(set(self.male_names)))
        self.A_females = list(set(self.A).intersection(set(self.fem_names)))
        if len(self.A_females) < len(self.B):
            self.godmothers =
sample(self.A_females, randint(1, len(self.A_females) - 1))
        else:
            self.godmothers = sample(self.A_females, randint(1, len(self.B) - 1))
        self.relation_S = list(zip(self.godmothers, self.B))
        self.children = [i[1] for i in self.relation_S]
        return self.relation_S
    def compadre(self):
        self.unmarked_in_B = list(set(self.B).difference(set(self.children)))
        self.B_males = list(set(self.B).intersection(set(self.male_names)))
        self.B_females = list(set(self.B).intersection(set(self.fem_names)))
        self.fathers = list(set(self.B_males).intersection(self.unmarked_in_B))
        self.mothers =
list(set(self.B_females).intersection(self.unmarked_in_B))
        self.pairs = list(zip(self.mothers, self.fathers))
        alpha_R = list(zip(self.godmothers, self.pairs))
        self.relation_R = []
        #alpha_R represents relation of godmother to pair,
        # not relation of godmother to distinct parent
        for i in alpha_R:
            self.relation_R.append((i[0], i[1][0]))
            self.relation_R.append((i[0], i[1][1]))
```

Роздруківка результатів виконання програми з контрольним прикладом Відношення R:



Відношення S:



Аналіз результатів та висновки

Під час виконання даної лабораторної роботи я вивчив операції над відношеннями, зокрема виконав індивідуальне завдання, відповідно до мого варіанту. Під час виконання лабораторної роботи проблем не виникло.