

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних технологій

Бакалаврська кваліфікаційна робота

на тему: Розробка програмного додатку для перевірки неоднорідності
рядів за допомогою непараметричних критеріїв

Виконав студент 4 курсу групи К-41
Напряму 6.050101 комп'ютерні науки,
Атанасов Сергій Петрович

Керівник асистент
Шуптар-Пориваєва Наталія Йосипівна

Консультант д.х.н., професор
Кругляк Юрій Олексійович

Рецензент к.георг.н., доцент
Бургаз Олексій Анатолійович

ЗМІСТ

Скорочення та умовні позначки	5
Вступ.....	7
1 Аналітичний огляд предметної області	8
Опис предметної області	8
Характеристика об'єкту розробки	9
Аналіз існуючих аналогів.....	9
Вибір засобів розробки	16
Мова програмування C	16
Microsoft Visual Studio	17
WinAPI.....	18
Он-лайн конструктор Mockingbird	20
2 Проектування об'єкту розробки	21
Ранговий і інверсійний алгоритми	21
Моделювання елементів системи	26
Діаграма діяльності	26
Діаграма варіантів використання	29
Проектування макетів інтерфейсу користувача.....	33
3 Опис програмного продукту	35
Реалізація алгоритму.....	35
3.1.1. Формат зберігання даних і їх завантаження.....	36
Комбінаторика та сортування	37
Конкатенація строки з числом.....	38
WinAPI потоки.....	39
Принцип роботи критерія на інверсіях та рангового критерію.....	39
Зберігання даних	40
Опис графічного інтерфейсу користувача	42
Висновки	48
Перелік джерел посилання	50

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ОДЕКУ – Одеський Державний Екологічний Університет

ПЗ – Програмне Забезпечення

API – Application Programming Interface

IDE – Integrated Development Engine

MFC – Microsoft Foundation Classes

PDF – Portable Document Format

PNG – Portable Network Graphics

UML – Unified Modeling Language

Action state – стан дії

Actor – актор на діаграмі використання

Ci – мова програмування

Concurrent for – розходження на дфіаграмі діяльності

Concurrent join – злиття на діаграмі діяльності

C ++ – мова програмування

C # – компільований статично типізований мова програмування

загального призначення

Drag and drop – спосіб оперування елементами інтерфейсу в інтерфейсах користувача

Java – типізована об'єктно-орієнтована мова програмування

Mockingbird – он-лайн конструктор

Pascal – мова програмування

Linux – операційна система

Objective-C – мова програмування

Visual Studio – інтегроване середовище розробки програмного забезпечення

Use Case Diagram – діаграми варіантів використання

WinAPI – це набір функцій, що працюють під управлінням ОС Windows

Windows Forms – інтерфейс програмування додатків (API), що відповідає за графічний інтерфейс користувача і є частиною Microsoft .NET фреймворк

Windows – операційна система

.NET – програмна платформа

ВСТУП

Критерій Уилкоксона для пов'язаних вибірок (також використовуються назви Т-критерій Уилкоксона, критерій Вілкоксона, критерій знакових рангів Уилкоксона, критерій суми рангів Уилкоксона) – непараметричний статистичний критерій, який використовується для порівняння двох пов'язаних (парних) вибірок за рівнем будь-якого кількісного ознаки, вимірюного в безперервній або в порядковій шкалою [1]¹⁾.

Такого роду задачі виконують студенти ОДЕКУ на декількох напрямках підготовки. Розрахунки займають велику кількість часу, на даний момент в допомогу студентам існує програмний додаток для розрахунків знакових рангів Уилкоксона за вибірками. Цей додаток має обмежені функції розрахунків і формування звітів, тому було прийнято рішення допомогти автоматизувати учбовий процес.

Мета роботи: розробка нового програмного додатку розрахунку Т-критерію Уилкоксона на базі сучасної мови програмування і створення оптимального інтерфейсу для користувача з урахуванням побажань викладачів і студентів.

Загальні характеристики кваліфікаційної роботи:

- повний обсяг сторінок пояснювальної записки – 50
- кількість рисунків – 25
- кількість таблиць – 0
- кількість посилань – 17

¹⁾[1] Інформаційний ресурс Medstatistic. URL: <http://medstatistic.ru/theory/wilcoxon.html>. (дата звернення 02.03.2019).

1 АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

Опис предметної області

Критерій Уїлкоксона можна застосувати в тих випадках, коли ознаки вимірянні принаймні за шкалою порядку, і зрушення між другим і першим вимірами теж можуть бути впорядковані. Для цього вони повинні варіювати в досить широкому діапазоні.

T-критерій Уилкоксона використовується для оцінки відмінностей між двома рядами вимірювань, виконаних для однієї і тієї ж сукупності досліджуваних, але в різних умовах або в різний час. Даний тест здатний виявити спрямованість і вираженість змін - тобто, чи є показники більше зсунутими в одному напрямку, ніж в іншому.

В принципі, можна застосовувати критерій T і в тих випадках, коли зрушення приймають тільки три значення: -1 , 0 і $+1$, але тоді критерій T навряд чи додасть щось нове до тих висновків, які можна було б отримати за допомогою критерію знаків. Ось якщо зрушення змінюються, скажімо, від -30 до $+45$, тоді має сенс їх ранжувати і потім підсумовувати ранги.

Суть методу полягає в зіставленні вираженості зрушень в тому і іншому напрямках по абсолютній величині. Для цього спочатку ранжуються всі абсолютні величини зрушень, а потім сумуються ранги.

Якщо зрушення в позитивну і в негативну сторону відбуваються випадково, то суми рангів абсолютних значень їх будуть приблизно рівні.

Якщо ж інтенсивність зсуву в одному з напрямків переважає, то сума рангів абсолютних значень зрушень в протилежну сторону буде значно нижче, ніж це могло б бути при випадкових змінах [2]¹⁾.

Спочатку виходять із припущення про те, що типовим зсувом буде зрушення в більш часто зустрічається напрямку, а нетиповим.

¹⁾ [2] Інформаційний ресурс Math.semestr. URL: <https://math.semestr.ru/group/wilcoxon.php>. (дата звернення 01.03.2019).

Характеристика об'єкту розробки

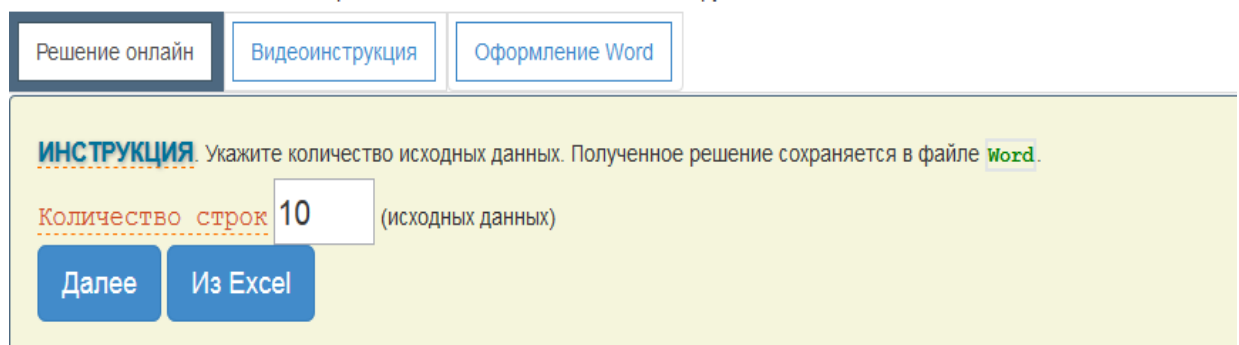
Об'єкт розробки – це програмний додаток для студентів ОДЕКУ, який допоможе у розрахунку критерію суми рангів Уилкоксона для інверсійної і рангової обробки даних.

Основні вимоги до проекту:

- реалізація алгоритму розрахунку даних;
- розробка оптимального інтерфейсу користувача для полегшення вводу інформації;
- реалізація функції «створити звіт розрахунків» для подальшого аналізу результатів;
- розробка додаткових функцій для паралельної обробки великих рядків даних з метою прискорення часу обчислення.

Аналіз існуючих аналогів

В ролі першого аналога розглянемо онлайн калькулятор «Math.semestr» за адресою (<https://math.semestr.ru/group/wilcoxon.php>) (дата звернення 02.03.2019)(рис. 1):



Решение онлайн Видеоинструкция Оформление Word

ИНСТРУКЦИЯ. Укажите количество исходных данных. Полученное решение сохраняется в файле [Word](#).

Количество строк (исходных данных)

Далее Из Excel

Рисунок 1 – Головна форма ресурсу «Math.semestr»

По-перше, необхідно ввести кількість строк для розрахунку критерія Віллоксона. Якщо з'являються питання щодо користування ресурсом, є можливість переглянути відео-інструкцію щодо коректного завдання вхідних даних і формування звіту по закінченню розрахунків.

Після натиску на кнопку «Далее» програма переходить до другої форми (рис. 2).

Т-критерий Виллоксона	
<input type="button" value="Ввод данных"/> <input type="button" value="Видеоинструкция"/>	
До	После
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

Рисунок 2 – Форма для вхідних даних на ресурсі «Math.semestr»

Тут необхідно завантажити ряд даних для перевірки одним із способів (безпосередньо з клавіатури або завантажити з файлу даних) (рис. 3). Далі калькулятор виведе два ряди значень (до і після вибірки для подальшого аналізу).

Після завершення розрахунків програма відображає результат, який можна завантажити у дос-файл як звіт.

До недоліків даного ресурсу слід віднести відсутність завантаження додатку до комп'ютера користувача, швидкість розрахунку залежить від інтернет доступу.

Крім цього, онлайн-калькулятор не дозволяє проводити розрахунки з великою кількістю даних і порівнювати результати між собою [1]¹⁾.

І-критерии Вилкоксона

Ввод данных | Видеоинструкция

До	После
3	6
4,5	3,6
4	4
3,69	5
2	6
5	6,2
8	8,2
7	8
4,5	6
5	5

После эксперимента значения показателей должны быть

Исключать нулевые сдвиги

Для получения решения необходимо отключить блокираторы типа AdBlock (Adguard, uBlock) и перезагрузить страницу. Для DrWeb, Kaspersky, Avast возможно понадобится включение сайта в список исключений

Рисунок 3 – Завдання початкових даних

Другим аналогом розглянуто онлайн ресурс для математичних розрахунків «<https://www.psychol-ok.ru/statistics/wilcoxon>» (дата звернення 03.03.2019).

На даному онлайн ресурсі є можливість провести розрахунки (рис. 4). Для початку роботи необхідно ввести в першу колонку дані першого виміру («До»), а в другу колонку дані другого виміру («Після»).

¹⁾ [1] Інформаційний ресурс Medstatistic. URL: <http://medstatistic.ru/theory/wilcoxon.html>. (дата звернення 02.03.2019).

Дані вводяться по одному числу на рядок, без пробілів, пропусків і т.д. Вводяться тільки цифри. Дробові числа вводяться зі знаком «.», причому підказок для цього випадку не передбачено.

Крім даних, для даного ресурсу слід явно вказати чи є необхідність урахувати нулевий сдвиг. Після заповнення колонок слід натиснути на кнопку «Крок 2», щоб розрахуватися Т-критерію Вілкоксона, або кнопку «Сброс» у разі помилкового введення вхідних даних для розрахунків.

До	После
<div style="border: 1px solid gray; padding: 5px; min-height: 150px;"> 2 4.5 3 1 3 5 4 2 4 </div>	<div style="border: 1px solid gray; padding: 5px; min-height: 150px;"> 2 4.8 3 1 3.2 5 4.7 2 4 </div>
<input checked="" type="checkbox"/> - Учитывать нулевой сдвиг?	
<div style="display: flex; justify-content: center; gap: 20px;"> <div style="border: 1px solid gray; padding: 5px 15px;">Шаг 2</div> <div style="border: 1px solid gray; padding: 5px 15px;">Сброс</div> </div>	

Рисунок 4 – Головна форма для завдання початкових даних

Після введення даних для аналізу переходимо до другої форми калькулятора (рис. 5), на якій відображаються результати розрахунків у вигляду таблиці.

Крім цього, виводиться критичне значення, і відповідно графік (рис. 6).

Автоматический расчет T- критерия Вилкоксона

Шаг 2

[!] За нетипичный сдвиг было принято «увеличение значения».

N	"До"	"После"	Сдвиг ($t_{\text{после}} - t_{\text{до}}$)	Абсолютное значение сдвига	Ранговый номер сдвига
1	2	2	0	0	3.5
2	4.5	4.8	0.3	0.3	8
3	3	3	0	0	3.5
4	1	1	0	0	3.5
5	3	3.2	0.2	0.2	7
6	5	5	0	0	3.5
7	4	4.7	0.7	0.7	9
8	2	2	0	0	3.5
9	4	4	0	0	3.5
Сумма рангов нетипичных сдвигов:					24

Результат: $T_{\text{Эмп}} = 24$

Рисунок 5 – Форма з результатами розрахунків на ресурсі «psycholok.ru»

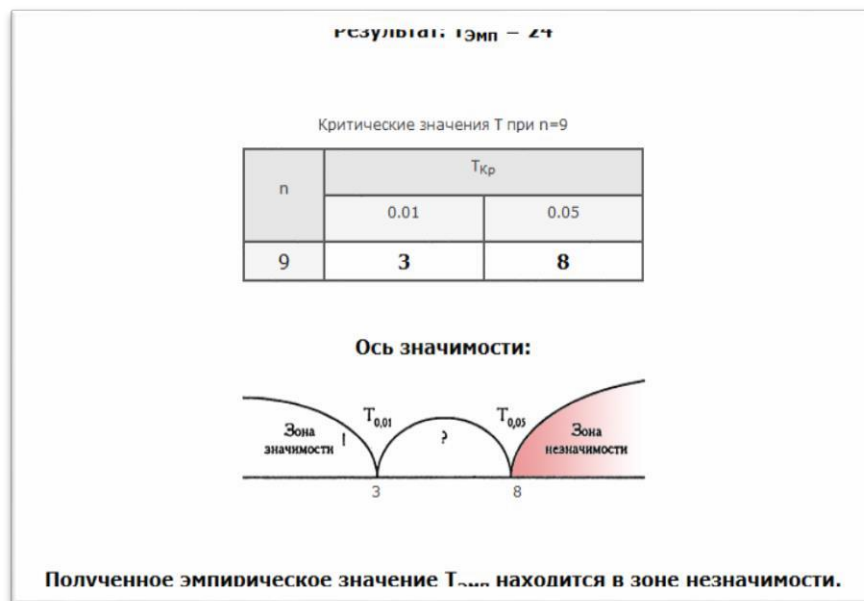


Рисунок 6 – Відображення графіку для емпіричного показника

Взагалі, даний ресурс має багато переваг:

- зручний інтерфейс для користувача;
- є підказки для коректного вводу початкових даних;
- відображення таблиці з результатами розрахунків;

- побудова графіку для емпіричного показника.

Недоліки:

- це он-лайн ресурс, а це означає пряму залежність від доступу до глобальної мережі;
- відсутня функція формування звіту з результатами;
- дані потрібно вводити тільки з клавіатури;
- можливість розрахунку тільки для невеликої послідовності даних;
- не зберігається історія розрахунків для подальшого аналізу.

В ролі останнього аналогу слід розглянути програмний додаток, яким користуються студенти у даний час (рис. 7).

```

C:\Users\Me\Desktop\Vilkoхone_old\Win32\Debug\Project1.exe
Ранговый вариант критерия Вилкоксона.
                                     X Y
12345678910
1002003004007008009001200130023000

Wнабл. =1+2+3+4+5+6+7+8+9+10=55
Wнижн.кр. = (10, 10, 0.025)=78
Wверх.кр. = (10+10+1)*10-78=132
78Ряды X и Y принадлежат к различным генеральным совокупностям.
Повторный расчет данного ряда производить? (Y/N)
  
```

Рисунок 7 – Консольне вікно додатку «Vilkoхone_old»

Цей додаток написано мовою програмування Pascal. Він виводить інформацію щодо розрахунків по ранговому критерію та по інверсійному (рис.8).

```

C:\Users\Me\Desktop\Vilkoхone_old\Win32\Debug\Project1.exe
Критерий Вилкоксона, основанный на подсчёте числа инверсий.
X Y
11
по X :0=0
по Y :1=1
Sigma = 0,5 Mn = 0,5 U = 0 tкр(0.05,0)=1,96
Границы критической области следующие :
-0,48 <=0 <= 1,48
-0,48 <=1 <= 1,48
Ряды X и Y с доверительной вероятностью 95% являются однородными.
Повторный расчет данного ряда производить ? (Y/N)

```

Рисунок 8 – Розрахунок по інверсійному критерію у додатку
«Vilkoхone_old»

Програма має ряд недоліків, які будуть усунені в процесі реалізації дипломного проекту, а саме:

- консольний незручний інтерфейс для звичайного користувача. Вибір дії здійснюється натисканням цифри, яка припадає тієї або іншої дії;
- одночасно можна обробляти тільки один критерій (тобто немає можливості одну і ту ж пару обчислювати за двома критеріями паралельно);
- відсутність можливості паралельно обробляти кілька пар;
- ядро написано на Pascal, де всі змінні включаючи і ітератори оголошені глобально, що загальмовує доступ до цих змінним, а, отже, гальмує роботу програми;
- використовує базовий (навчальний) вид сортування, він більш тривалий.

За результатами аналізу аналогів поставлена мета для об'єкту розробки: створити програмний додаток для розрахунку критерію Уилкоксона, який ліквідує усі вищезазначені недоліки аналогів, а саме зручний інтерфейс користувача, зручний ввід даних за допомогою завантаження файлу а збереження результатів у окремому файлі для подальшої обробки.

Крім цього, слід звернути увагу на оптимізацію програмного алгоритму для прискорення розрахунків, так як на практиці будуть обчислюватись великий об'єм інформації.

Вибір засобів розробки

Мова програмування C

Мова Cі в цілому стандартизований і компілятори для нього існують практично під будь-яку платформу. Велика частина системного програмного забезпечення пишеться на мові Cі. Ядро операційної системи Linux, та й багатьох інших, повністю написано на мові Cі. Розробники системного ПЗ намагаються зводити до мінімуму переносних залежні вставки на асемблері, що дозволяє повторно використовувати більшу частину коду [3]¹⁾.

По суті, мова Cі це стерпний макроасемблер сучасності. Мова Cі дозволяє вирішувати завдання на рівні абстракції, який максимально наближений до заліза. У ньому немає ніякого прихованого поведінки за винятком макропроцесора.

Кожна інструкція мови транслюється в машинний код прямим і зрозумілим чином. Виповнюється рівно той код, який програміст бачить на екрані. Це природним чином сприяє написанню швидкого коду без зайвих накладних витрат. Адже для того, щоб код написаний на Cі працював

¹⁾ [3] Інформаційний блог з програмування. URL: <https://elizarov.livejournal.com/27125.html>. (дата звернення 10.03.2019).

повільно, виконуючи якісь зайві операції, виклики цих зайвих операцій явно запрограмувати.

Всі перші п'ять найбільш популярних зараз мов програмування так чи інакше засновані на мові Сі. На додаток до самого Сі на першому місці, С ++ і Objective-C є його розширеннями в тому чи іншому вигляді, а Java і С # використовують синтаксис заснований на мові Сі.

Microsoft Visual Studio

За останні кілька років продуктова лінійка Visual Studio була значно розширена безліччю нових можливостей.

Найголовнішим компонентом є Visual Studio IDE (рис. 9), яке використовують мільйони розробників у всьому світі і створюють за допомогою цього інструменту чудові додатки.

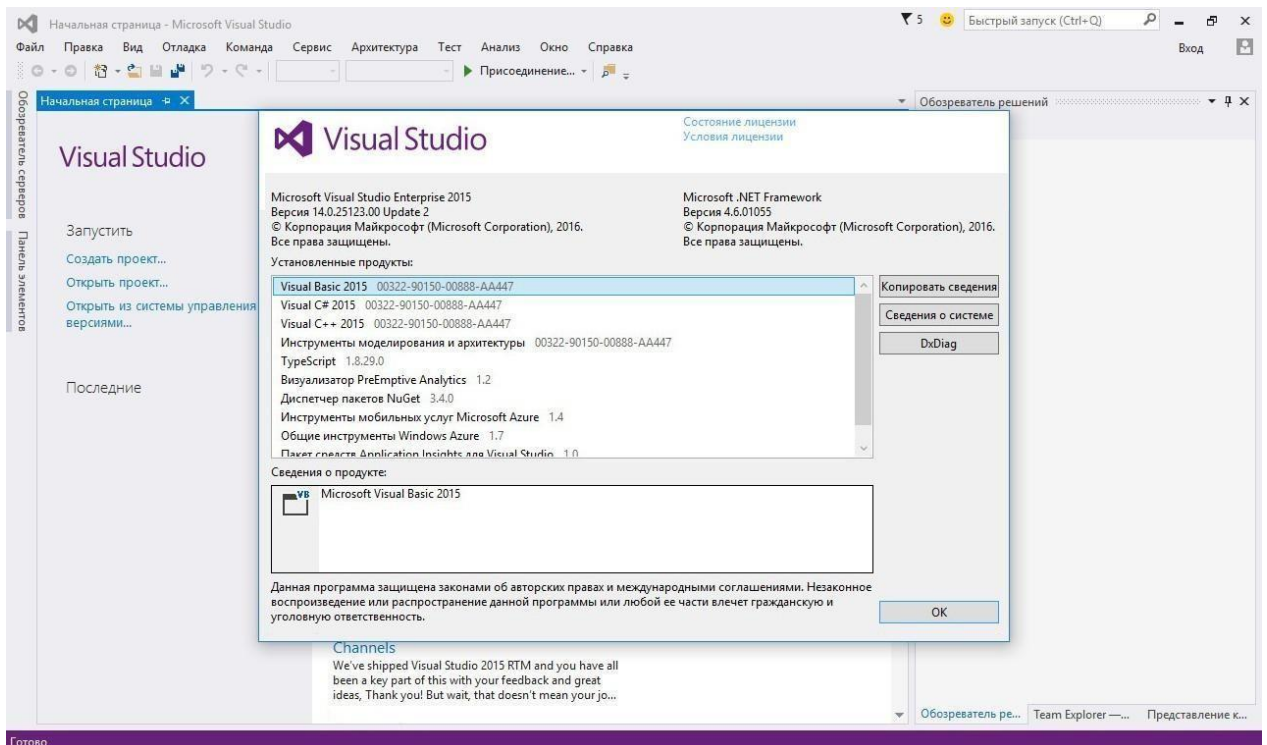


Рисунок 9 – Середя розробки Microsoft Visual Studio

Розробники витрачають основну частину свого робочого часу використовуючи Visual Studio.

Visual Studio 2015 об'єднує налагодження і профілювання в єдиний Діагностичний хаб, який дозволяє перевірити коректність реалізації і параметри продуктивності вашого додатка в режимі контекстної отладки.

Як б розробка не проводилась, веб, хмара або мобільні пристрої, важливим є гнучкість і можливість вибору. Visual Studio і .NET надають такий вибір, і дозволяють використовувати різні платформи і мови програмування. Visual Studio це найпотужніша IDE (IDE - Integrated Development Engine (Інтегроване середовище розробки)) під .NET зручний, надійний, інтуїтивно зрозумілий, легко розширюваний інструмент [4]¹⁾.

Microsoft Visual Studio 2015 має зручний аналізатор коду з «лампочками», що відображають різні підказки для розробника; призначені для користувача макети інтерфейсу IDE для кожного з варіантів розробки з можливістю швидко і зручно перемикатися між ними; вдосконалені інструменти налагодження і діагностики.

WinAPI

WinAPI – це набір функцій (API – Application Programming Interface), що працюють під управлінням ОС Windows. Вони містяться в бібліотеці «windows.h» [5]²⁾.

За допомогою WinAPI можна створювати різні віконні процедури, діалогові вікна, програми і навіть ігри. Ця бібліотека є базовою в освоєнні програмування Windows Forms, MFC, тому що ці інтерфейси є надбудовами цієї бібліотеки. Освоївши її, буде легко створювати форми, і розуміти, як це

¹⁾ [4] Інформаційний ресурс Shiftoffproblem з роботи у Visual Studio. URL://shift-offproblem.com/visual-studio-community-2015-review/. (дата звернення 11.03.2019).

²⁾ [5] Інформаційний ресурс з програмування. URL: <http://cppstudio.com/post/9384/>. (11.03.2019).

відбувається. Використовуючи Windows API, можна розробляти додатки, які успішно працюють у всіх версіях Windows, використовуючи переваги функцій і можливостей, унікальних для кожної версії.

Він створює і використовує вікна для відображення виводу, запиту введення даних користувачем і виконання інших завдань, що підтримують взаємодію з користувачем (рис. 10).

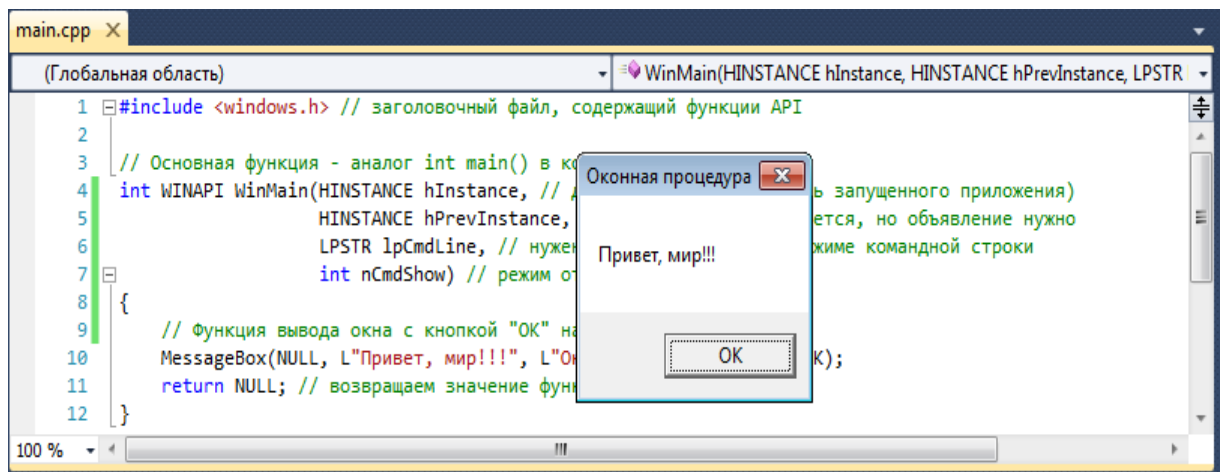


Рисунок 10 – Приклад діалогового вікна за допомогою WinAPI

Більшість додатків створюють як мінімум одне вікно. Функції API дозволяють організувати інтерфейс між прикладною програмою і середовищем, в якій працює ця програма. Виклик функцій API дозволяє програмі отримувати доступ до ресурсів середовища і управляти її роботою. Як правило, API задає стандарт взаємодії середовища і прикладної програми [6]¹⁾.

API – це інтерфейс програмування додатків, термін, часто згадуваний розробниками програмного забезпечення. Якщо об'єкт розробки має функцію, що дозволяє звертатися до нього з інших додатків, то це – API застосування.

¹⁾ [6] Інформаційний портал з програмування. URL: <https://habr.com/ru/post/352096/>. (дата звернення 13.03.20190).

Параметри, які приймає функція, утворюють її API, так як вони є засобом, за допомогою якого інші програми взаємодіють з цією функцією L URL [7]²⁾.

Он-лайн конструктор Mockingbird

Mockingbird – це сервіс для побудови прототипів додатків, щоб показати найважливіше: ідею, інформацію, взаємодії (рис. 11). До можливостей Mockingbird можна віднести наступне:

- створення прототипів для користувача інтерфейсів за короткий час з функцією Drag and drop;
- додавання кількох сторінок в макет;
- перехід між сторінками використовуючи посилання;
- редагування макета співробітниками або клієнтами в режимі реального часу;
- експорт в форматах PNG і PDF.

²⁾ [7] Інформаційний блог з програмування. URL: <https://eas.me/tag/winapi/>. (дата звернення 13.03.2019).

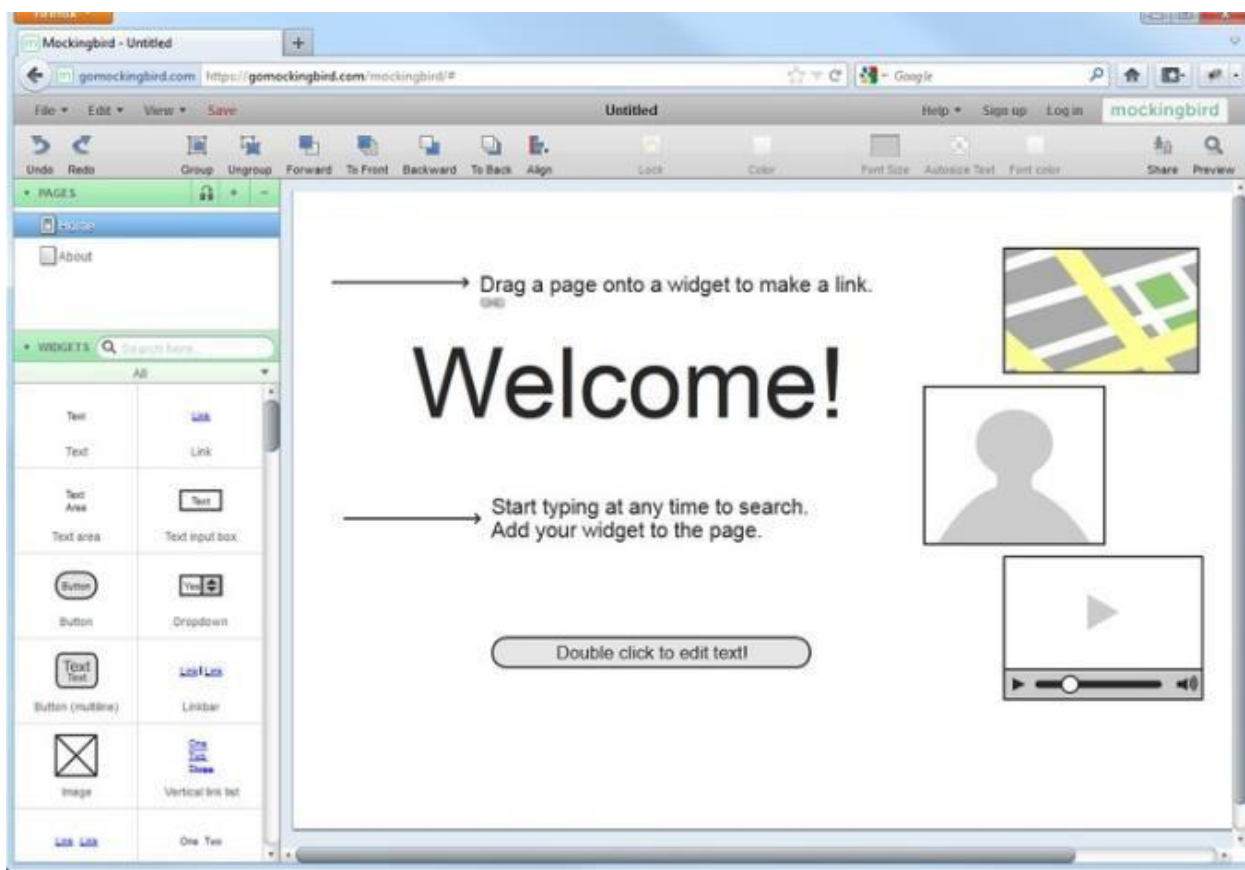


Рисунок 11 – Интерфейс онлайн конструктора Mockingbird

2 ПРОЄКТУВАННЯ ОБ'ЄКТУ РОЗРОБКИ

Ранговий і інверсійний алгоритми

Статистичний критерій – це вирішальне правило, що забезпечує математично обґрунтоване прийняття істинної і відхилення помилкової гіпотези. Статистичні критерії будуються на основі статистики $\Psi(x_1, x_2, \dots, x_n)$ – деякої функції від результатів спостережень x_1, x_2, \dots, x_n . Статистика Ψ є випадковою величиною з певним законом розподілу.

Серед значень статистики Ψ виділяють критичну область $\Psi_{кр}$ з властивістю: якщо емпіричне значення статистики $\Psi_{емп}$ належать області $\Psi_{кр}$, то нульову гіпотезу відхиляють (відкидають), інакше - приймають. Статистичні критерії визначають у практичній діяльності метод розрахунку певного числа, яке позначається як емпіричне значення критерію, наприклад, $t_{емп}$ для t-критерію Стьюдента [8]¹⁾.

Співвідношення емпіричного і критичного значень критерію є підставою для підтвердження чи спростовування гіпотези. Наприклад, у разі застосування t-критерію Стьюдента, якщо $|t_{емп}| \geq |t_{кр}|$, то значення статистики належать критичній області і нульова гіпотеза H_0 відхиляється (приймається альтернативна гіпотеза H_1). Правила прийняття статистичного рішення обумовлюються для кожного критерію.

Відповідно до статистичних гіпотез статистичні критерії діляться на параметричні й непараметричні [9]²⁾.

Параметричні критерії використовуються в завданнях перевірки параметричних гіпотез і включають у свій розрахунок показники розподілу, наприклад, середні, дисперсії тощо. Це такі відомі класичні критерії, як z-критерій, t-критерій Стьюдента, F-критерій Фішера та ін. Непараметричні критерії перевірки гіпотез засновані на операціях з іншими даними, зокрема, частотами, рангами тощо. Це λ -критерій Колмогорова-Смірнова, U-критерій Вілкоксона-Манна-Вітні та багато інших.

Параметричні критерії дозволяють прямо оцінити рівень основних параметрів генеральних сукупностей, різниці середніх і відмінності в

дисперсіях. Критерії спроможні виявити тенденції зміни ознаки при переході від умови до умови, оцінити взаємодію двох і більш факторів у впливі на зміни ознаки. Параметричні критерії вважаються дещо більш потужними, ніж непараметричні, за умов, якщо ознака виміряна за інтервальною шкалою і нормально розподілена.

¹⁾ [8] Інформаційний ресурс електронної бібліотеки Pidruchniki. URL: https://pidruchniki.com/12590605/statistika/statistichni_kriteriyi. (дата звернення 15.03.2019).

²⁾ [9] Лоева І. Д., Бургаз О.А. Збірник методичних вказівок до практичних занять з дисципліни «Обробка і аналіз інформації» для студентів денної форми навчання, напрям підготовки 6.040106 «Екологія, охорона навколишнього середовища та збалансоване природокористування». Одеса: ОДЕКУ, 2010. 65 с.

Проте з інтервальною шкалою можуть виникнути певні проблеми, якщо дані, представлено не в стандартизованих оцінках. До того ж перевірка розподілу «на нормальність» вимагає досить складних розрахунків, результат яких заздалегідь невідомий. Найчастіше розподіли ознак відрізняються від нормального, тоді доводиться звертатися до непараметричних критеріїв [10]¹⁾.

Непараметричні критерії позбавлені перерахованих вище обмежень. Проте вони не дозволяють здійснити пряму оцінку рівня таких важливих параметрів, як середнє або дисперсія, з їхньою допомогою неможливо оцінити взаємодію двох і більше умов або факторів, що впливають на зміну ознаки.

Непараметричні критерії дозволяють вирішити деякі важливі завдання, які супроводжують дослідження в психології і педагогіці: виявлення відмінностей у рівні досліджуваної ознаки, оцінка зсуву значень досліджуваної ознаки, виявлення відмінностей у розподілах ознак [9]²⁾.

Застосування критеріїв для прийняття (відхилення) статистичних гіпотез завжди здійснюються з довірчою ймовірністю, інакше кажучи, на певному рівні значущості

Одним із непараметричних критеріїв є критерій Вілкоксона. Цей критерій використовується в двох варіантах: перший базується на підрахунках числа інверсій, другий – ранговий.

Якщо ми маємо прості статистичні ряди випадкових величин, то дослідження однорідності таких вибірок треба починати з операції ранжування. Тобто випадкові величини, що належать до цих вибірок, розташовують у загальну послідовність у порядку збільшення (або зменшення) їх значень.

¹⁾ [10] Електронна бібліотека Studopedia/ URL: https://studopedia.com.ua/1_47878_statistichniy-kriteriy-perevirki-osnovnoi-gipotezi.html. (дата звернення 20.03.2019).

²⁾ [9] Лоева І. Д., Бургаз О.А. Збірник методичних вказівок до практичних занять з дисципліни «Обробка і аналіз інформації» для студентів денної форми навчання, напрям підготовки 6.040106 «Екологія, охорона навколишнього середовища та збалансоване природоко-ристування». Одеса:ОДЕКУ, 2010. 65 с.

Якщо якому-небудь значенню x у загальній послідовності передують деяке значення y , то кажуть, що ця пара утворює інверсію. Позначимо сумарну кількість інверсій через U [9]¹⁾.

Відомо, що в однорідних рядах, кожний з котрих має не менше 10 членів, число інверсій розподіляється приблизно за нормальним законом з параметрами:

$$m_U = \frac{m \cdot n}{2} \quad (1)$$

і

$$\sigma_U^2 = \frac{m \cdot n}{12} (m + n + 1), \quad (2)$$

де

n і m – кількість членів у першій та другій вибірках.

Нульову гіпотезу H_0 сформулюємо так: ряди випадкових величин X і Y належать до однієї генеральної сукупності на рівні значущості α .

Для перевірки цієї гіпотези необхідно знайти межі допустимих значень U , що відділяють область прийняття гіпотези H_0 від критичної області, яка, як зрозуміло, є двосторонньою. Якщо значення критерію, що отримане за даними спостережень, попаде до області прийняття гіпотези, то гіпотеза H_0 на цьому рівні значущості не відхиляється. У протилежному випадку приймається альтернативна гіпотеза H_1 [11]¹⁾.

H_0 визначається виразом:

Область прийняття гіпотези

$$m_U - t_{кр}(\alpha, \nu) \sigma_U \leq U \leq m_U + t_{кр}(\alpha, \nu) \sigma_U, \quad (3)$$

¹⁾ [9] Лоева І. Д., Бургаз О.А. Збірник методичних вказівок до практичних занять з дисципліни «Обробка і аналіз інформації» для студентів денної форми навчання, напрям підготовки 6.040106 «Екологія, охорона навколишнього середовища та збалансоване природоко-ристування». Одеса: ОДЕКУ, 2010. 65 с.

¹⁾ [11] Електронна бібліотека Westudents. URL: <https://westudents.com.ua/glavy/88649-parametrichn-neparametrichn-kriter.html>. (дата звернення 23.03.2019).

а критична – нерівностями:

$$U_{кр1} < m_U - t_{кр}(\alpha, \nu) \sigma_U, \quad (4)$$

$$U_{кр2} > m_U + t_{кр}(\alpha, \nu) \sigma_U. \quad (5)$$

У нерівностях (3 - 5) $\sigma_U = \sqrt{\sigma_U^2}$ – середній квадратичний відхил числа інверсій; $t_{кр}(\alpha, \nu$ – критерій Стьюдента для рівня значущості α і числа)

степенів вільності $\nu = m + n - 2$ [2].

За допомогою рангового критерію Вілкоксона перевірка статистичної гіпотези H_0 про однорідність двох рядів випадкових величин починається з того, що:

- розташовують члени обох вибірок в один об'єднаний ряд (в порядку збільшення їх значень);
- далі знаходять в об'єднаному ряду суму порядкових номерів (рангів) членів вибірки, об'єм якої менший. Позначимо її через $W_{виб.}$;
- потім визначають нижню критичну точку за формулою:

$$W_{кр.н}(m, n, \alpha/2) = \frac{(m+n+1) \cdot m - 1}{2} - Z_{кр} \sqrt{\frac{m \cdot n \cdot (m+n+1)}{12}}, \quad (6)$$

де m і n – об'єми вибірок випадкових величин X і Y ;

α – рівень значущості;

$Z_{кр}$ – визначається по таблицях функції Лапласа (використовуємо

додаток Е) за правилом:

$$\Phi(Z_{кр}) = \frac{1-\alpha}{2}. \quad (7)$$

Верхня критична точка розраховується за формулою:

$$W_{кр.в} = (m + n + 1) \cdot m - W_{кр.н}. \quad (8)$$

Гіпотеза H_0 на заданому рівні значущості α не відкидається, якщо

$$W_{кр.н} < W_{виб.} < W_{кр.в}. \quad (9)$$

За умов $W_{виб.} < W_{кр.н}$ або $W_{виб.} > W_{кр.в}$ приймається альтернативна гіпотеза H_1 про те, що ряди X та Y є неоднорідними [9]¹⁾.

Моделювання елементів системи

Діаграма діяльності

При моделюванні поведінки проекрованої або аналізованої системи виникає необхідність не тільки представити процес зміни її станів, але і деталізувати особливості алгоритмічної і логічної реалізації виконуваних системою операцій. Традиційно для цієї мети використовувалися блок-схеми або структурні схеми алгоритмів. Кожна така схема акцентує увагу на послідовності виконання певних дій або елементарних операцій, які в сукупності призводять до отримання бажаного результату.

Для моделювання процесу виконання операцій в мові UML використовуються так звані діаграми діяльності. Застосовувана в них графічна нотація багато в чому схожа на нотацію діаграми станів, оскільки на діаграмах діяльності також присутні позначення станів і переходів. Відмінність полягає

¹⁾ [9] Лоева І. Д., Бургаз О.А. Збірник методичних вказівок до практичних занять з дисципліни «Обробка і аналіз інформації» для студентів денної форми навчання, напрям підготовки 6.040106 «Екологія, охорона навколишнього середовища та збалансоване природоко-ристування». Одеса:ОДЕКУ, 2010. 65 с.

в семантиці станів, які використовуються для уявлення не діяльностей, а дій, і у відсутності на переходах сигнатури подій [12]¹⁾.

Кожне стан на діаграмі діяльності відповідає виконанню деякої елементарної операції, а перехід в наступний стан спрацьовує тільки при завершенні цієї, операції в попередньому стані. Графічно діаграма діяльності видається у формі графа діяльності, вершинами якого є стани дії, а дугами - переходи від одного стану дії до іншого.

Стан дії (action state) є спеціальним випадком стану з деяким вхідним дією і принаймні одним виходять зі стану переходом. Цей перехід неявно передбачає, що вхідний дію вже завершилося. Стан дії не може мати внутрішніх переходів, оскільки воно є елементарним. Звичайне використання стану дії полягає в моделюванні одного кроку виконання алгоритму (процедури) або потоку управління.

Один з найбільш значущих недоліків звичайних блок-схем або структурних схем алгоритмів пов'язаний з проблемою зображення паралельних гілок окремих обчислень.

Оскільки розпаралелювання обчислень істотно підвищує загальну швидкодію програмних систем, необхідні графічні примітиви для подання паралельних процесів.

У мові UML для цієї мети використовується спеціальний символ для розділення і злиття паралельних обчислень або потоків управління. Таким символом є пряма риска, аналогічно позначенню переходу в формалізмі мереж Петрі [12]¹⁾.

Як правило, така риска зображується відрізком горизонтальної лінії, товщина якої трохи ширше основних суцільних ліній діаграми діяльності. При цьому поділ (concurrent fork) має один вхідний перехід і кілька виходять. Злиття (concurrent join), навпаки, має кілька вхідних переходів і один виходить.

¹⁾ [12] Інформаційний ресурс Github. URL: https://flexberry.github.io/ru/fd_activity-diagram.html. (дата звернення 26.03.2019).

¹⁾ [12] Офіційний сайт UML/ URL: <https://www.uml.org/>. (дата звернення 28.03.2019).

При розробці проекту нової системи, процес функціонування якої базується на нових технологічних рішеннях, ситуація видається більш складною. А саме, до початку роботи над проектом можуть бути невідомі не тільки деталі реалізації окремих діяльностей, а й сам зміст цих діяльностей стає предметом розробки [13]²⁾.

В даному випадку домінуючим буде спадний процес розробки від більш загальних схем до уточнюючих їх діаграм. При цьому досягнення такого рівня деталізації всіх діаграм, який достатній для розуміння особливостей реалізації всіх дій і діяльностей, може служити ознакою завершення окремих етапів роботи над проектом.

Для алгоритму роботи програми побудуємо діаграму діяльності (рис. 12).

²⁾ [13] Інформаційний ресурс Github. URL: https://flexberry.github.io/ru/fd_activity-diagram.html. (дата звернення 29.03.2019).

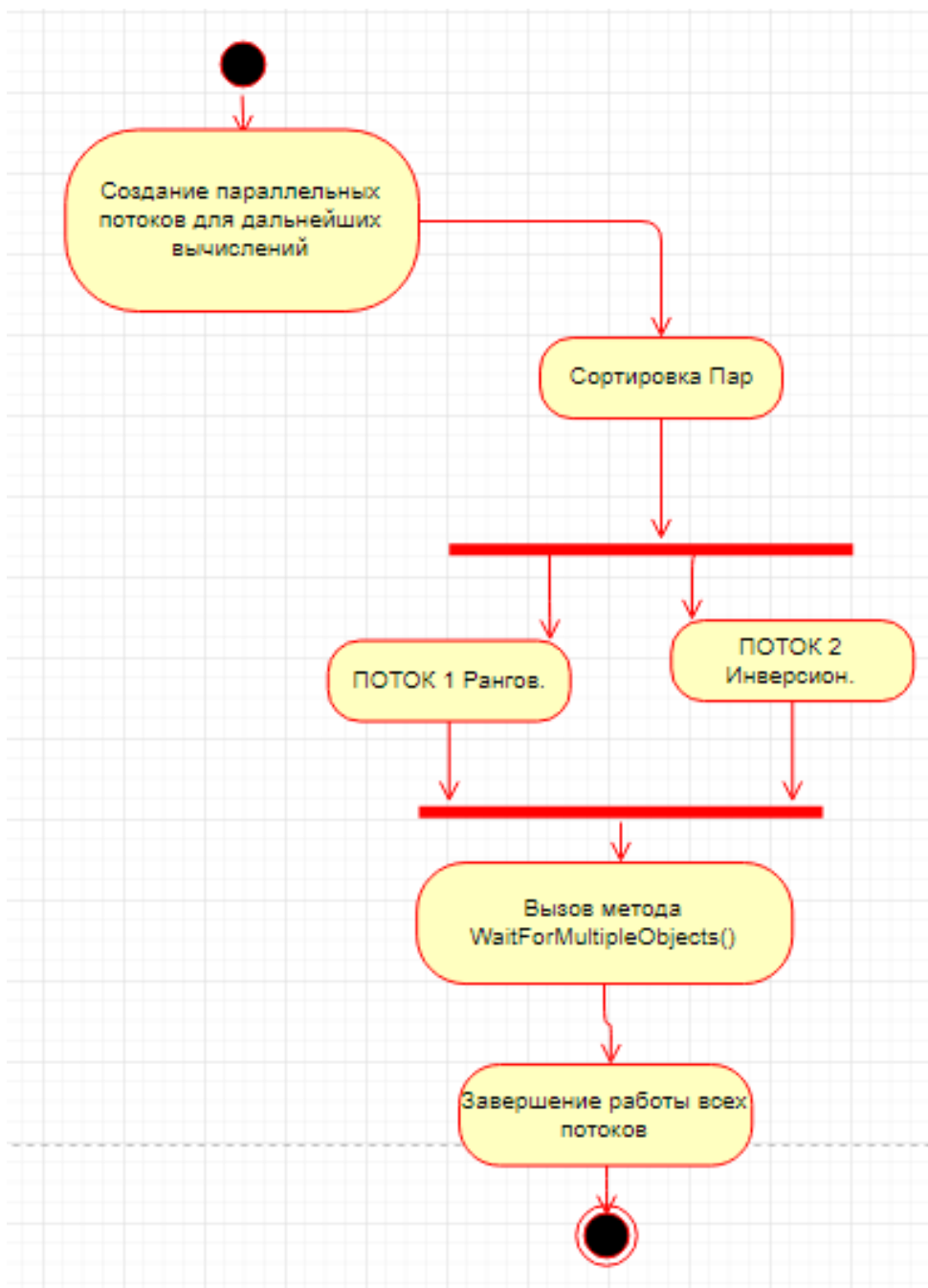


Рисунок 12 – Діаграма діяльності для алгоритму роботи програми

Діаграма варіантів використання

Візуальне моделювання в UML можна уявити як певний процес поетапного спуску від найбільш загальної і абстрактної концептуальної моделі вихідної системи до логічної, а потім і до фізичної моделі відповідної програмної системи.

Для досягнення цих цілей спочатку будується модель у формі так званої діаграми варіантів використання (use case diagram), яка описує функціональне призначення системи або, іншими словами, те, що система буде робити в процесі свого функціонування. Діаграма варіантів використання є вихідним концептуальним поданням або концептуальною моделлю системи в процесі її проектування і розробки [14]¹⁾.

Суть даної діаграми складається в наступному: проектована система представляється у вигляді безлічі сутностей або акторів, що взаємодіють з системою за допомогою так званих варіантів використання. При цьому актором (actor) або дійовою особою називається будь-яка сутність, що взаємодіє з системою ззовні. Це може бути людина, технічний пристрій, програма або будь-яка інша система, яка може служити джерелом впливу на моделюючу систему так, як визначить сам розробник.

У свою чергу, варіант використання (use case) служить для опису сервісів, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який чинять системою при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізовано взаємодію акторів з системою.

У найзагальнішому випадку, діаграма варіантів використання є граф спеціального виду, який є графічної нотації для представлення конкретних варіантів використання, акторів, можливо деяких інтерфейсів, і відносин між цими елементами. При цьому окремі компоненти діаграми можуть бути укладені в прямокутник, який позначає проектовану систему в цілому [15]²⁾.

Слід зазначити, що відносинами даного графа можуть бути тільки деякі фіксовані типи взаємозв'язків між акторами і варіантами використання, які в

¹⁾ [14] Інформаційний ресурс з UML-моделювання. URL: <https://www.uml-diagrams.org/use-case-diagrams.html>. (дата звернення 01.05.2019).

²⁾ [15] Інформаційний ресурс з UML-моделювання. URL: www.agilemodeling.com/artifacts/useCaseDiagram.htm. (дата звернення 01.05.2019).

сукупності описують сервіси або функціональні вимоги до моделюється системи.

Конструкція або стандартний елемент мови UML варіант використання застосовується для специфікації загальних особливостей поведінки системи або будь-якої іншої сутності предметної області без розгляду внутрішньої структури цієї сутності. Кожен варіант використання визначає послідовність дій, які повинні бути виконані проектованої системою при взаємодії її з відповідним актором. Діаграма варіантів може доповнюватися пояснювальним текстом, який розкриває зміст або семантику складових її компонентів. Такий пояснювальний текст отримав назву примітки або сценарію.

Окремий варіант використання позначається на діаграмі еліпсом, усередині якого міститься його коротка назва або ім'я у формі дієслова з пояснювальними словами.

Кожний виконується варіантом використання метод реалізується як неподільна транзакція, а саме виконання сервісу не може бути перервано ніяким іншим екземпляром варіанту використання [14]¹⁾.

Застосування варіантів використання на всіх рівнях діаграми дозволяє не тільки досягти необхідного рівня уніфікації позначень для подання функціональності підсистем і системи в цілому, а й є потужним засобом послідовного уточнення вимог до проектованої системи на основі поетапного спуску від пакетів системи до операцій класів.

З іншого боку, модифікація окремих операцій класу може надати зворотний вплив на уточнення сервісу відповідного варіанту використання. Реалізувати ефект зворотного зв'язку з метою уточнення специфікацій або вимог на рівні пакетів системи [15]²⁾.

¹⁾ [14] Інформаційний ресурс з UML-моделювання. URL: <https://www.uml-diagrams.org/use-case-diagrams.html>. (дата звернення 01.05.2019).

²⁾ [15] Інформаційний ресурс з UML-моделювання. URL: www.agilemodeling.com/artifacts/useCaseDiagram.htm. (дата звернення 01.05.2019).

Актор є будь-яку зовнішню по відношенню до модельованої системи сутність, яка взаємодіє з системою і використовує її функціональні можливості для досягнення певної мети або вирішення приватних завдань.

При цьому актори служать для позначення узгодженого безлічі ролей, які можуть грати користувачі в процесі взаємодії з проектованою системою. Кожен актор може розглядатися як якась окрема роль щодо конкретного варіанту використання.

Стандартним графічним позначенням актора на діаграмах є фігурка "чоловічка", під якою записується конкретне ім'я актора [15]¹⁾.

Приклад діаграми варіантів використання представлено на рис. 13.

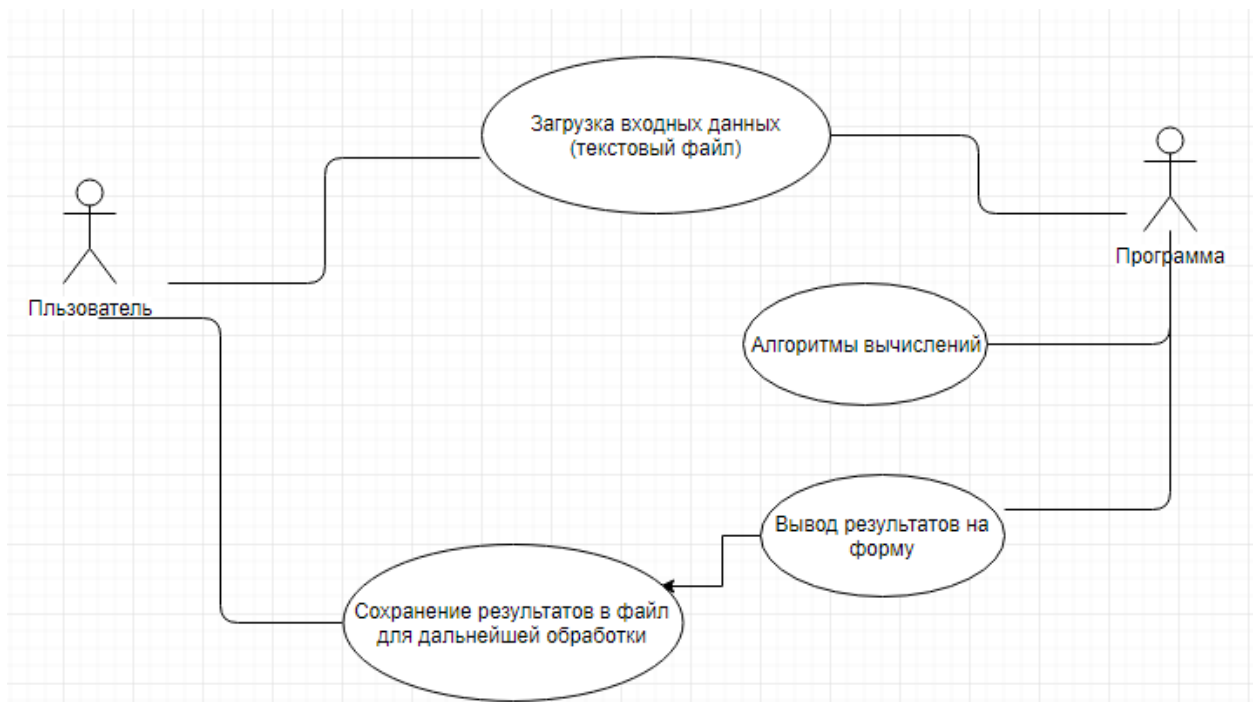


Рисунок 13 – Діаграма варіантів використання для об'єкту розробки

¹⁾ [15] Інформаційний ресурс з UML-модельовання. URL: www.agilemodeling.com/artifacts/useCaseDiagram.htm. (дата звернення 01.05.2019).

Проектування макетів інтерфейсу користувача

Під час проектування інтерфейсу користувача слід дотримуватись декількох правил:

- інтерфейс повинен бути інтуїтивно зрозумілим. Таким, щоб користувачеві не потрібно пояснювати як ним користуватися;
- для спрощення процесу вивчення необхідна довідка. Буквально - графічна підказка. Повне керівництво повинно бути частиною інтерфейсу, доступною в будь-який момент;
- інтерфейс повинен бути інтуїтивно зрозумілим. Таким, щоб користувачеві не потрібно пояснювати як ним користуватися [16]¹⁾.

Крім цього, слід пам'ятати, що програма повинна взаємодіяти з користувачем на основі найменшої значущою одиниці введення. Розробка інтерфейсу зазвичай починається з визначення завдання або набору задач, для яких продукт призначений.

Просте має залишатися простим. Його користувачі не замислюються над тим, як влаштована програма. Все, що вони бачать – це інтерфейс. Тому, з точки зору споживача саме інтерфейс є кінцевим продуктом.

Інтерфейс повинен бути орієнтованим на людину, тобто відповідати потребам людини і враховувати його слабкості. Потрібно постійно думати про те, з якими труднощами може зіткнутися користувач [17]²⁾.

Для створення макету інтерфейса програми використовувався он-лайн конструктор Mockingbird (рис.14).

¹⁾ [16] Інформаційний ресурс Habr. URL: <https://habr.com/ru/post/208966/>. (дата звернення 03.05.2019).

²⁾ [17] Інформаційний ресурс з розробки програмного забезпечення. URL: <https://medium.com/razrabotka-maketa-interfejsa-programm-osnovnie-pravila>. (дата звернення 05.05.2019).

LOAD		SAVE	
Item 1	Item 2		
Item 3	Item 4		
Item 5	Item 6		
Инверсионный Критерий		Ранговый Критерий	

Рисунок 14 – Макет головної форми програмного додатку

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ

Реалізація алгоритму

Для реалізації алгоритмів Вілкоксона потрібна реалізація ядра на мові Сі. Для даного проекту структура складається з:

- «global» – файл має базові підключення стандартних бібліотек С, а так само структури, з якими виробляються обрахунку, наприклад «sData» і «sItem»;
- «str» – призначений для конкатенації рядків з числами;
- «sort» – сортує масиви даних;
- «freading» – здійснює читання даних з файлу певного стандарту і поміщає його в масив структур «sData»;
- «WinAPI» – зовнішній модуль для роботи з WinAPI функціями;
- «combinatory» – для обчислень комбінаторних функцій;
- «vilkoxone» – визначає однорідність пар рядів даних;
- «entry» – точка входу. Здійснює управління всіма файлами ядра. По закінченню звільняє пам'ять.

Діаграма класів для структури ядра приведена на рис. 15:

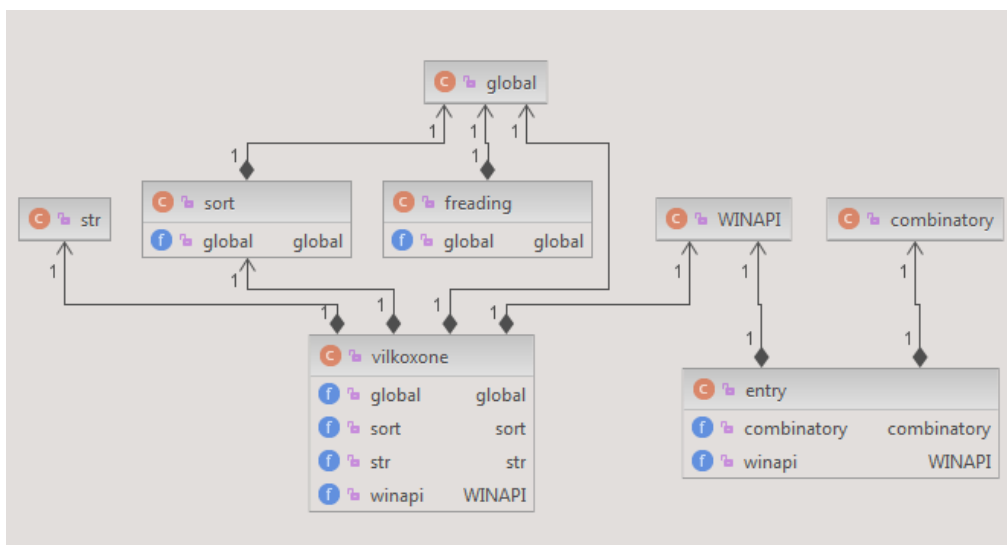


Рисунок 15 – Структура ядра програмного додатку

3.1.1. Формат зберігання даних і їх завантаження

Дані, що зберігаються в файлі, умовно діляться на дві групи:

- хедера;
- дані.

Хедер – це послідовність позитивних цілочисельних чисел, розділених роздільниками, наприклад, кома або пробіл. Закінчується послідовність на крапку з комою.

Кількість чисел у хедері ідентифікують кількість оброблюваних рядів, кожне число хедера визначає розмір відповідного ряду. Кожен ряд представляє такий же ряд, як і хедер, що закінчується на крапку з комою, але числа тут уже речові і можуть мати негативний знак.

Для завантаження даних був розроблений модуль «freading», який має особливу сигнатуру. Так як, що хедер, що ряди мають однакове уявлення, то чи не доцільно було б дублювати методи по зчитування окремо хедера і окремо даних з файлу.

Правда є невелика відмінність – коли читається хедер, невідомо скільки елементів він має, тобто невідомо, наскільки виділяти буфер під считку, в той час як при зчитування рядів даних ми вже заздалегідь знаємо виходячи з хедера.

Метод:

```
readSequence (FILE *, const uint MAX_LEN, uint * realLength)
```

має два параметри довжини – максимальний розмір буфера (MAX_LEN) і реальна довжина лічених даних (realLength), причому реальна довжина це покажчик.

Зроблено так, щоб після виконання методу зовні було відомо скільки ж насправді було лічено.

При читанні хедера MAX_LEN за замовчуванням визначається 100, при читанні же даних цей параметр приймає довжину ряду даних виходячи з хедера.

Комбінаторика та сортування

Після зчитування даних необхідно заздалегідь визначити скільки потоків виділяти під рішення задачі. На допомогу приходять комбінаторика, а саме метод поєднання, який в результаті повертає всі комбінації пар рядів.

$$\binom{n}{k} = C_n^k = \frac{n!}{k!(n-k)!}. \quad (11)$$

Кількість пар нам знадобиться і для ініціалізації GUI-компонентів.

Як сортування була обрана швидке сортування рекурсією «QuikSort». Правда його треба було трохи змінити, так як у нас не зовсім числові послідовності. Елементами масивів даних є структури sItem:

```
typedef struct {
    float value;
    ubyte id;}
sItem;
```

Для економії пам'яті, щоб при склеюванні двох рядів даних в один, і його сортуванні не копіювались дані, створюється покажчик на покажчик на масив, що звичайний алгоритм сортування не зуміє розсортувати.

Ось частина програмного коду, що відображає роботу сортування

```
if (low < high) {
    int pi = partition (items, low, high);
    quickSort (items, low, pi - 1); quickSort
    (items, pi + 1, high);}
```

Конкатенація строки з числом

Стандарт Cі не передбачає такого роду операції над рядками, тут немає таких сутностей як StringBuilder або StringBuffer, тому деякі алгоритми, а саме конкатенацію рядки з рядком або з числами як з цілочисельними, так і з речовими. Далі продемонстрований приклад програмного коду, що відображає принцип роботи конкатенації рядків.

У функцію передаються два рядки. Рядки обов'язково мають формат стандарту Cі, тобто останній символ за кодом має дорівнювати 0.

```
char * strConcatStr (char * str1, char * str2) {
```

Оголошується змінна довжини кінцевої рядки, а також відбувається обхід за вказівниками першого і другого рядка.

```
    uint length = 0;
    for (char * p = str1; * p != 0; ESHIFT (p, char), ++
length);
    for (char * p = str2; * p != 0; ESHIFT (p, char), ++length);
```

Створюємо масив дорівнює довжині конкатенірованної рядки і додаємо ще одиницю для останнього символу 0.

```
    char * res = CREATE (char, length + 1);
```

Знову робимо вказівний обхід обох, що входять в функцію, рядків, але тепер з копіюванням даних в результуючий конкатенірованний рядок.

```
    char * pRes = res;
    for (char * p = str1; * p != 0; ESHIFT (p, char), ESHIFT
(pRes, char))
        * PRes = * p;
    for (char * p = str2; * p != 0; ESHIFT (p, char), ESHIFT
char))
        * PRes = * p;
```

І останнім символом встановлюємо 0.

* PRes = 0;

WinAPI потоки

Для паралельної обробки алгоритмів і пар необхідно використовувати потоки. Для даного завдання взяті потоки з глобального модуля з цілим масивом бібліотек «WinAPI». Щоб не витратити час на очікування сортування пар (адже для обробки інверсійного або рангового алгоритму необхідний цей сортований масив) він запускається в паралельному потоці.

Потім відсортувавши, створюються ще два потоки для вирішення двох алгоритмів Вілкоксона. Запустивши всі потоки на обробку, викликається метод «WaitForMultipleObjects ()» для очікування закінчення всіх потоків. По завершенню всіх потоків всі ресурси програми вивільнюються.

Принцип роботи критерія на інверсіях та рангового критерію

Здійснюється обхід об'єднаного сортованого масиву. При обході обчислюються дві суми – U_x і U_y . Кожен раз при обході потрапляючи на елемент масиву, який числиться за рядом X в суму U_x додається кількість елементів масиву, що знаходяться зліва від поточного і належать ряду Y . Так само і по відношенню до елементів, що належить до ряду Y .

В кінці необхідно визначити однорідність порівнюваних рядів. Для цього потрібно визначити, чи знаходяться ці дві суми U_x і U_y в межах: $U_{down} \leq U_x, U_y \leq U_{up}$.

Принцип роботи рангового критерію: тут же при обході сортованого ряду визначаються групи елементів, які мають рівні значення даних. Кожна група має один загальний індекс (ранг), який визначається як середнє арифметичне від кількості елементів в групі.

Для визначення однорідності рядів обчислюється сума W . Для початку визначаються менший з вихідних рядів.

Далі при обході сортованого масиву, наткнувшись на елемент, що належить меншому ряду, виходить ранг групи, який додається в суму W . В кінці, якщо ця сума лежить в межах:

$$Wkr_down < W < Wkr_up,$$

то ряди однорідні, інакше немає.

Зберігання даних

Корисно було б зберегти результати в файл. Так як у роботі розглядаються пари рядів по двом алгоритмам, було б зручно зберігати результати у вигляді таблиці (рис. 16).

	1	2	3	4	5	6	7	8	9	10
1										
2	1									
3	-	+	+	+	+	+	+	+	+	+
4	-	-	+	+	+	+	+	+	+	+
5	-	-	-	+	+	+	+	+	+	+
6	-	-	-	-	+	+	+	+	+	+
7	-	-	-	-	-	+	+	+	+	+
8	-	-	-	-	-	-	+	+	+	+
9	-	-	-	-	-	-	-	+	+	+
10	-	-	-	-	-	-	-	-	-	

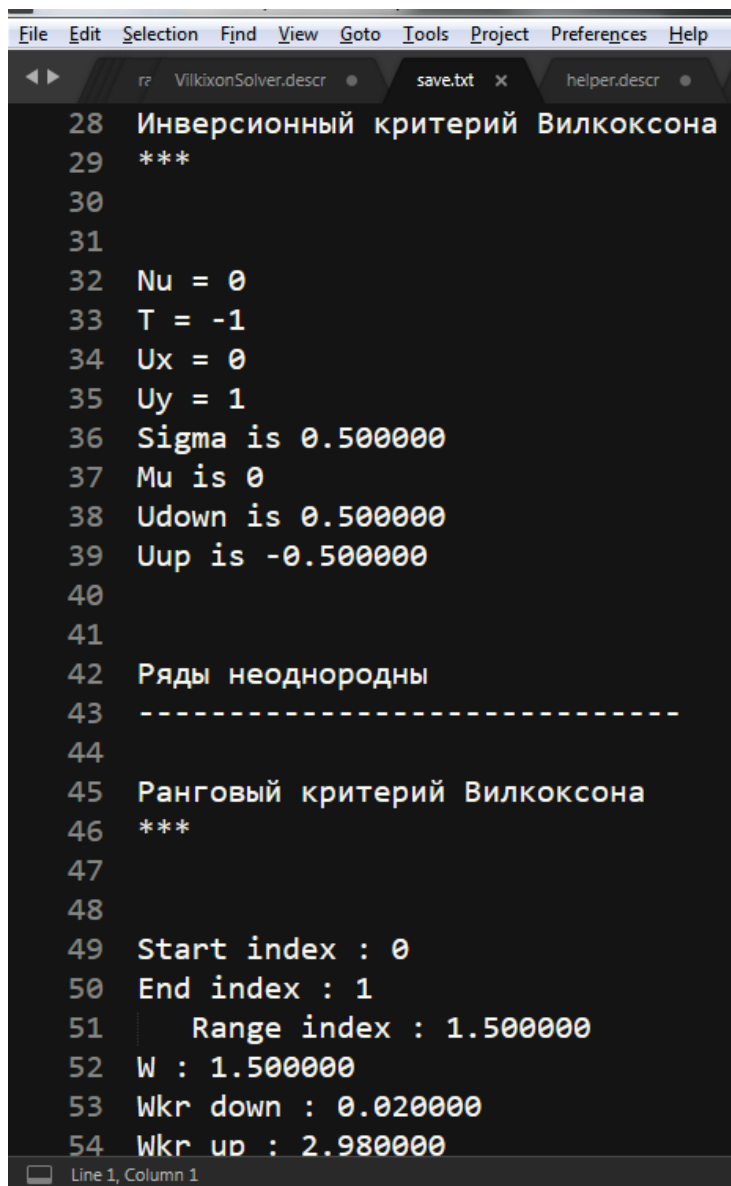
=====
Пара: 1 - 2

Рисунок 16 – Таблица пар

Для цього всі результати перед збереженням поміщаються в двовимірний масив. Так само створюється і масив ширини колонок. Обходячи кожну колонку таблиці визначається максимальне її значення по ширині.

При записи результатів в файл масив ширини використовується щоб колонки були рівними і значення в них центрована по ширині.

Крім таблиці у файлі зберігаються логи виконання програми (рис. 17).



```
File Edit Selection Find View Goto Tools Project Preferences Help
r7 VilkixonSolver.descr save.txt x helper.descr
28 Инверсионный критерий Вилкоксона
29 ***
30
31
32 Nu = 0
33 T = -1
34 Ux = 0
35 Uy = 1
36 Sigma is 0.500000
37 Mu is 0
38 Udown is 0.500000
39 Uup is -0.500000
40
41
42 Ряды неоднородны
43 -----
44
45 Ранговый критерий Вилкоксона
46 ***
47
48
49 Start index : 0
50 End index : 1
51 Range index : 1.500000
52 W : 1.500000
53 Wkr down : 0.020000
54 Wkr up : 2.980000
Line 1, Column 1
```

Рисунок 17 – Представлення логів у файлі

Опис графічного інтерфейсу користувача

Для зручності користування програмою передбачено GUI інтерфейс.

Сюди входять:

- кнопка «Завантажити»;
- кнопка «Зберегти»;
- таблиця результатів критеріїв;
- «ListView» для управління логами окремих пар рядів;
- два «EditText» компонента для виведення логів по інверсійної і рангових алгоритму.

За допомогою конструктора Delphi можна досить швидко і зручно створити GUI інтерфейс, чому і був обраний. Стартова форма програмного додатку представлена на рис. 18:

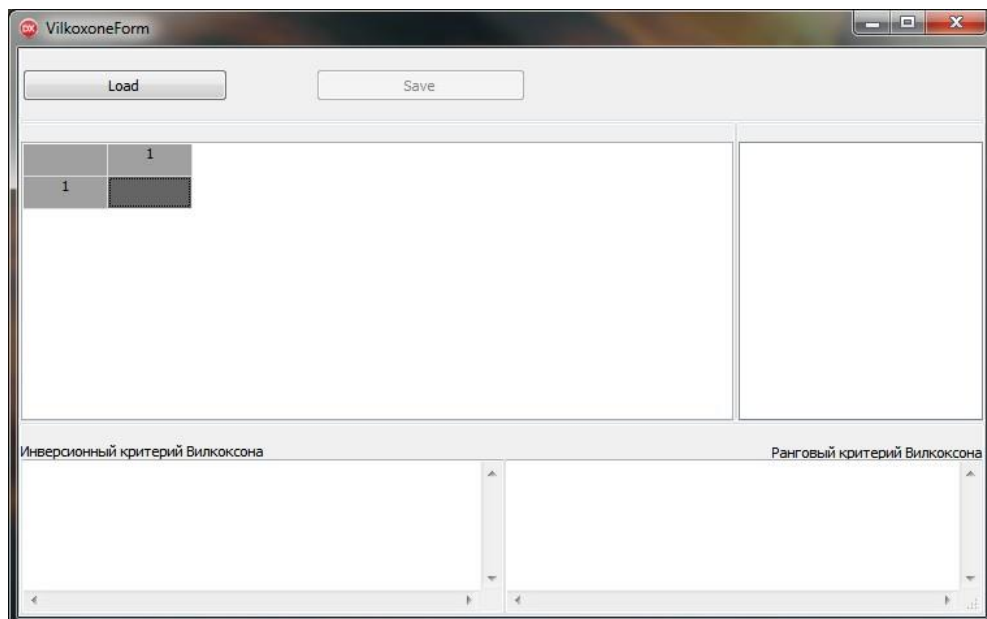


Рисунок 18 – Стартова форма програмного додатку «Vilkozone Solver»

Для користувача доступні дві кнопки:

- «Load» – для завантаження вхідних даних з текстового файлу (рис. 19);

– «Save» – для збереження даних у окремий текстовий файл (рис. 20).

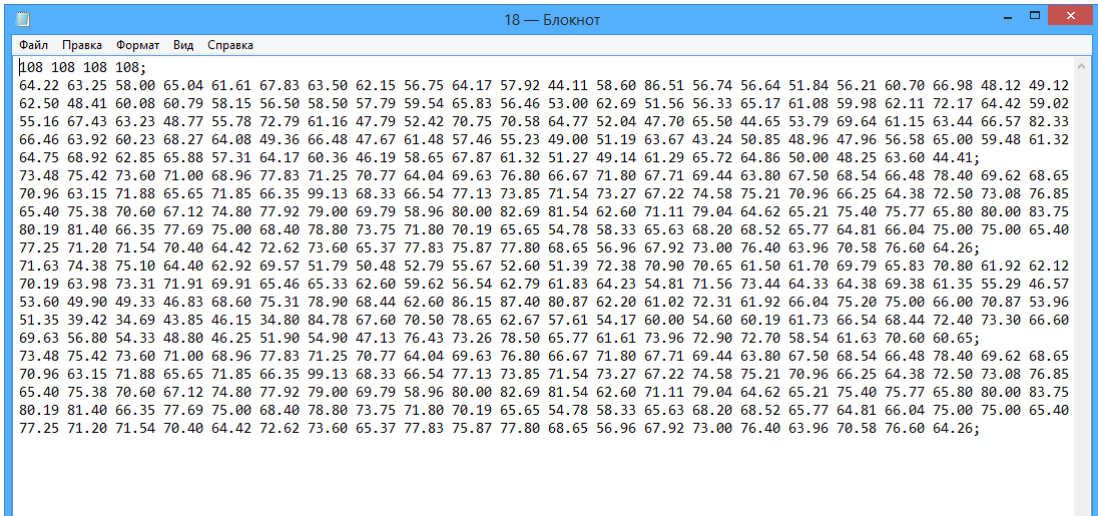


Рисунок 19 – Приклад вхідних даних у текстовому файлі

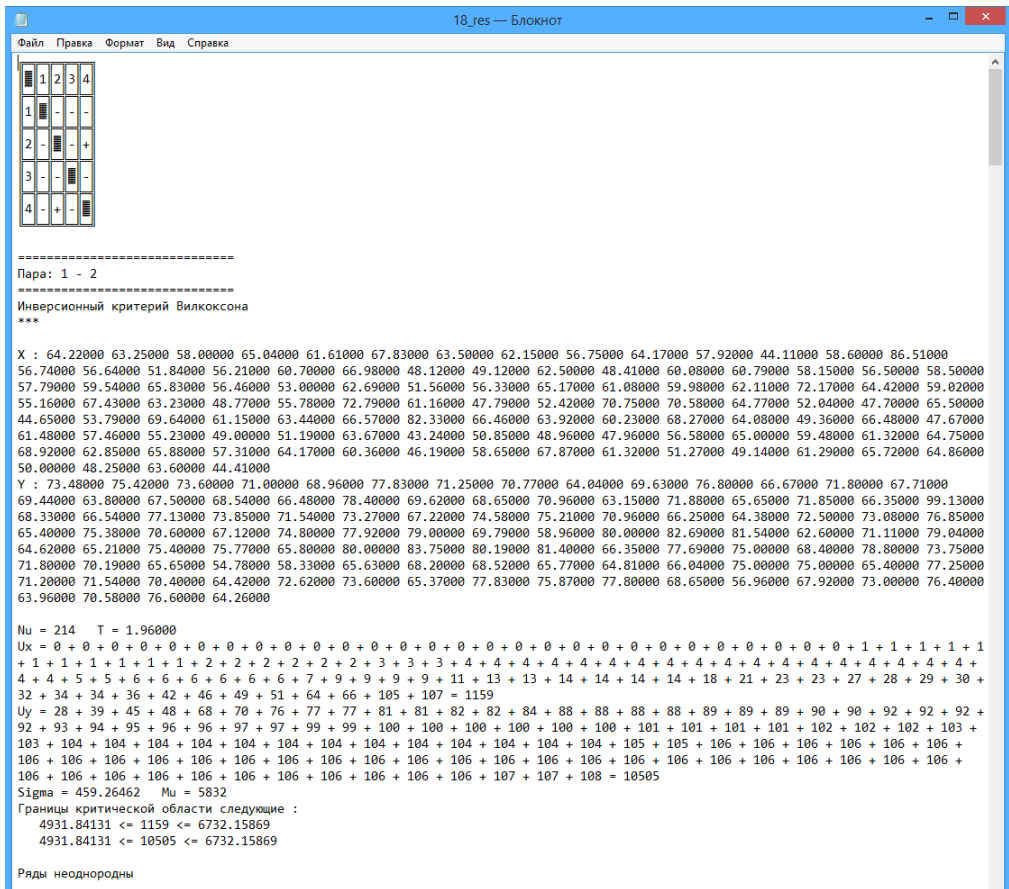


Рисунок 20 – Приклад збереження обчислювань у вихідному файлі

В лівому вікні відображається таблиця з отриманими результатами, в правій – пари чисел (рис. 21).

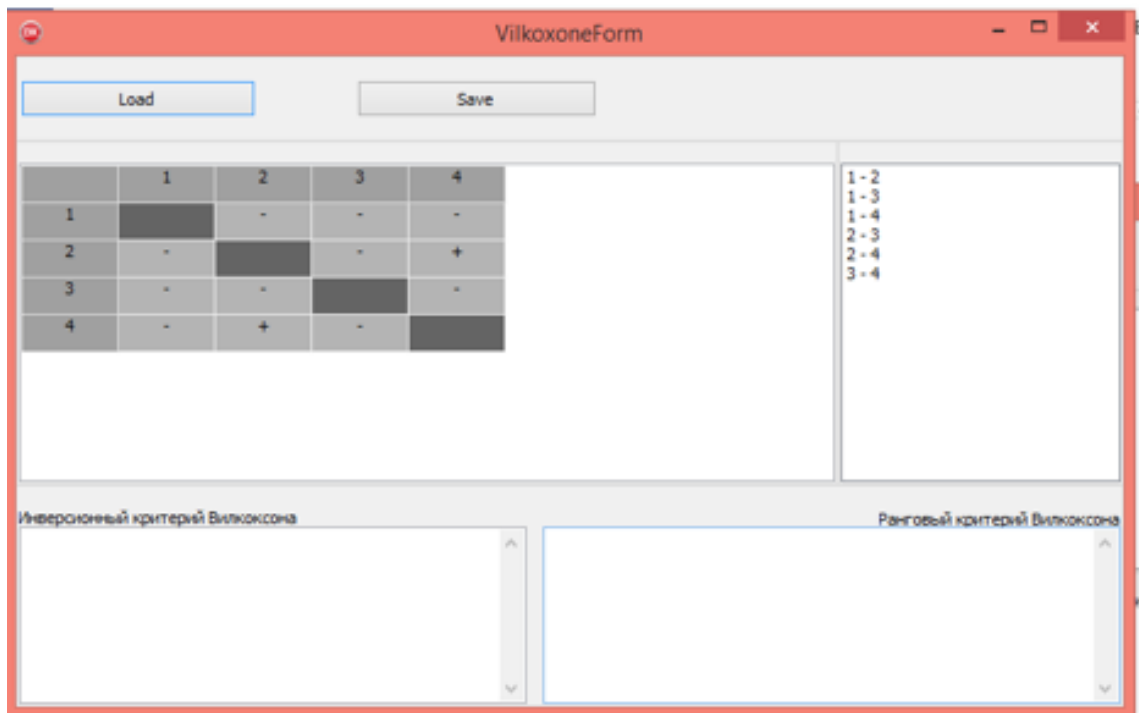


Рисунок 21 – Представлення вхідних даних на формі

Приклад виводу розрахунків по двом критеріям (інверсійному і за ранговим критерієм Вілкоксона) проводиться одночасно і для зручності користувача на форму виводяться одразу результати обох обчислювань для їх подальшого аналізу (рис. 22).

Після отримання розрахунків користувач програмного додатку має можливість зберігати отримані дані у окремому файлі так як розрахунки являють собою невід’ємну частину курсового проекту для студентів (рис. 23).

Під час зберігання є можливість обрати формат вихідного файлу.

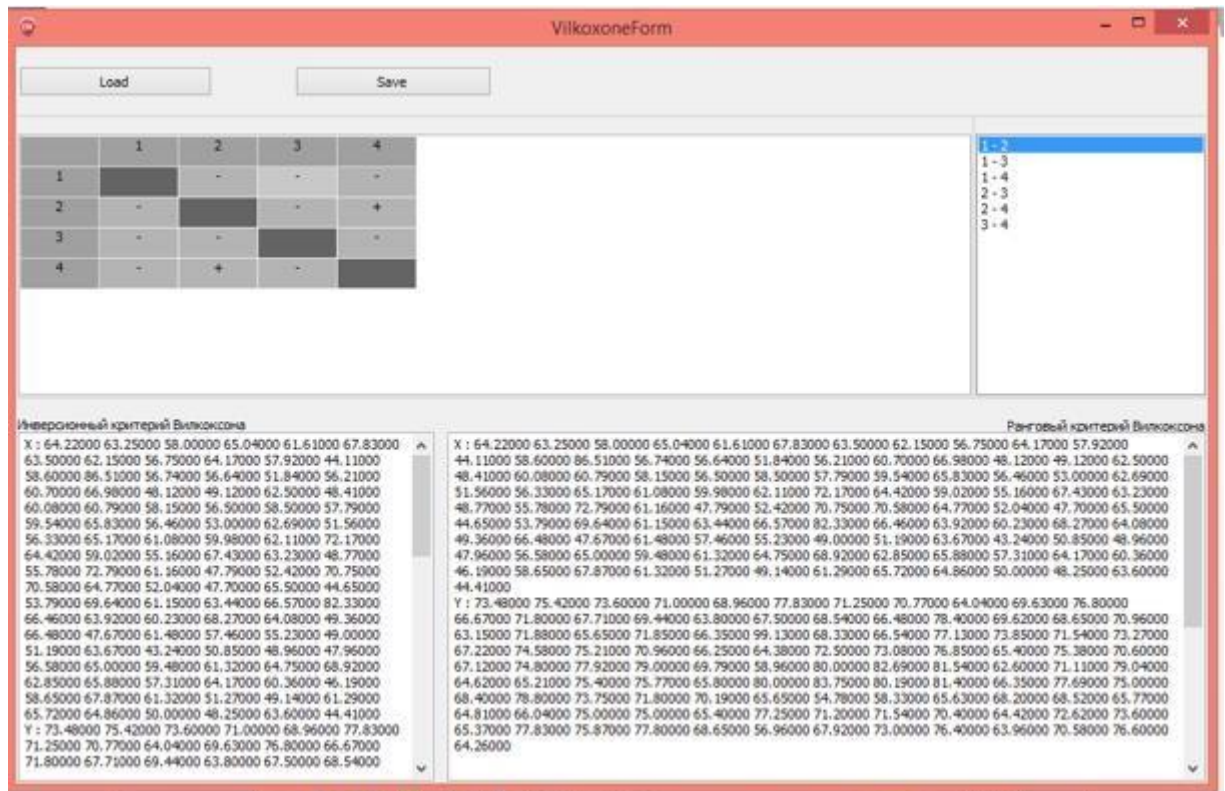


Рисунок 22 – Результати розрахунків на формі користувача

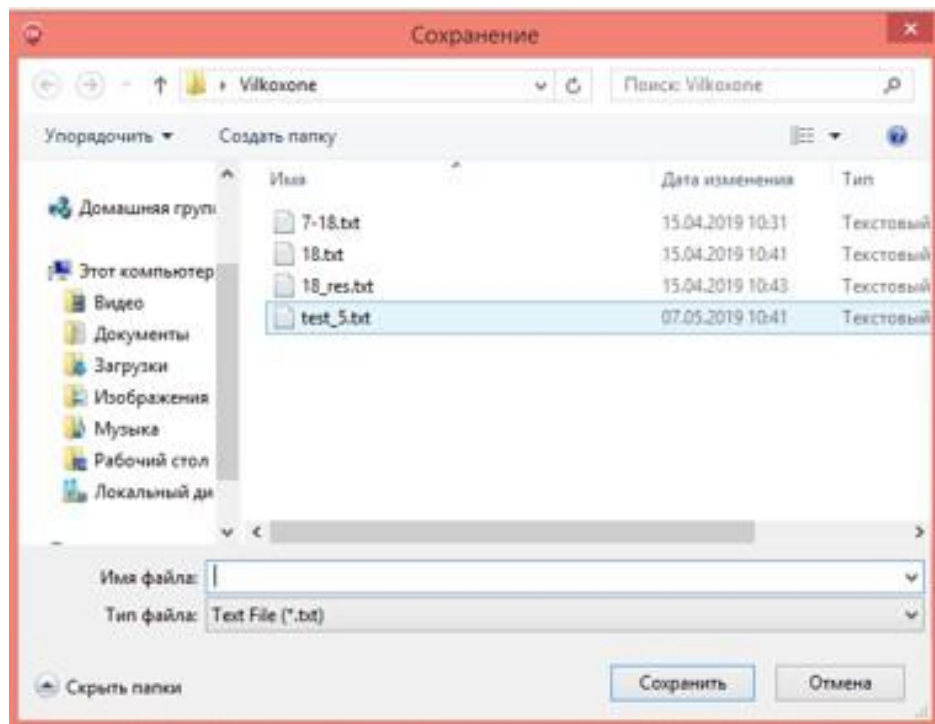


Рисунок 23 – Приклад зберігання вихідних даних у файл

Ранговый критерий Вилкоксона

X : 64.22000 63.25000 58.00000 65.04000 61.61000 67.83000 63.50000 62.15000 56.75000 64.17000 57.92000 44.11000 58.60000 86.51000
 56.74000 56.64000 51.84000 56.21000 60.70000 66.98000 48.12000 49.12000 62.50000 48.41000 60.08000 60.79000 58.15000 56.50000 58.50000
 57.79000 59.54000 65.83000 56.46000 53.00000 62.69000 51.56000 56.33000 65.17000 61.08000 59.98000 62.11000 72.17000 64.42000 59.02000
 55.16000 67.43000 63.23000 48.77000 55.78000 72.79000 61.16000 47.79000 52.42000 70.75000 70.58000 64.77000 52.04000 47.70000 65.50000
 44.65000 53.79000 69.64000 61.15000 63.44000 66.57000 82.33000 66.46000 63.92000 60.23000 68.27000 64.08000 49.36000 66.48000 47.67000
 61.48000 57.46000 55.23000 49.00000 51.19000 63.67000 43.24000 50.85000 48.96000 47.96000 56.58000 65.00000 59.48000 61.32000 64.75000
 68.92000 62.85000 65.88000 57.31000 64.17000 60.36000 46.19000 58.65000 67.87000 61.32000 51.27000 49.14000 61.29000 65.72000 64.86000
 50.00000 48.25000 63.60000 44.41000

Y : 73.48000 75.42000 73.60000 71.00000 68.96000 77.83000 71.25000 70.77000 64.04000 69.63000 76.80000 66.67000 71.80000 67.71000
 69.44000 63.80000 67.50000 68.54000 66.48000 78.40000 69.62000 68.65000 70.96000 63.15000 71.88000 65.65000 71.85000 66.35000 99.13000
 68.33000 66.54000 77.13000 73.85000 71.54000 73.27000 67.22000 74.58000 75.21000 70.96000 66.25000 64.38000 72.50000 73.08000 76.85000
 65.40000 75.38000 70.60000 67.12000 74.80000 77.92000 79.00000 69.79000 58.96000 80.00000 82.69000 81.54000 62.60000 71.11000 79.04000
 64.62000 65.21000 75.40000 75.77000 65.80000 80.00000 83.75000 80.19000 81.40000 66.35000 77.69000 75.00000 68.40000 78.80000 73.75000
 71.80000 70.19000 65.65000 54.78000 58.33000 65.63000 68.20000 68.52000 65.77000 64.81000 66.04000 75.00000 75.00000 65.40000 77.25000
 71.20000 71.54000 70.40000 64.42000 72.62000 73.60000 65.37000 77.83000 75.87000 77.80000 68.65000 56.96000 67.92000 73.00000 76.40000
 63.96000 70.58000 76.60000 64.26000

Ряд X меньше. Ориентируемся по нему

Wнабл. = 1.00000 + 2.00000 + 3.00000 + 4.00000 + 5.00000 + 6.00000 + 7.00000 + 8.00000 + 9.00000 + 10.00000 + 11.00000 + 12.00000 +
 13.00000 + 14.00000 + 15.00000 + 16.00000 + 17.00000 + 18.00000 + 19.00000 + 20.00000 + 21.00000 + 22.00000 + 23.00000 + 24.00000 +
 25.00000 + 26.00000 + 27.00000 + 28.00000 + 30.00000 + 31.00000 + 32.00000 + 33.00000 + 34.00000 + 35.00000 + 36.00000 + 37.00000 +
 38.00000 + 39.00000 + 40.00000 + 42.00000 + 43.00000 + 44.00000 + 45.00000 + 46.00000 + 47.00000 + 49.00000 + 50.00000 + 51.00000 +
 53.00000 + 54.00000 + 55.00000 + 56.00000 + 57.00000 + 58.00000 + 59.00000 + 60.00000 + 61.00000 + 62.00000 + 63.00000 + 64.00000 +
 65.00000 + 66.50000 + 66.50000 + 68.00000 + 69.00000 + 70.00000 + 71.00000 + 72.00000 + 74.00000 + 75.00000 + 77.00000 + 78.00000 +
 79.00000 + 80.00000 + 81.00000 + 82.00000 + 84.00000 + 87.00000 + 88.50000 + 88.50000 + 90.00000 + 93.50000 + 96.00000 + 97.00000 +
 99.00000 + 100.00000 + 101.00000 + 102.00000 + 107.00000 + 111.00000 + 114.00000 + 115.00000 + 120.00000 + 121.50000 + 124.00000 +
 126.00000 + 129.00000 + 132.00000 + 133.00000 + 136.00000 + 143.00000 + 148.00000 + 152.50000 + 155.00000 + 169.00000 + 172.00000 +
 212.00000 + 215.00000 = 7045.50000
 10816.84180 < 7045.50000 < 12619.15820

Ряды X и Y не принадлежат к одной генеральной совокупности

Рисунок 25 – Приклад інформації у вихідному файлі для рангового
 розрахунку

ВИСНОВКИ

Об'єкт розробки – це програмний додаток для студентів ОДЕКУ, який стане допомогою у практичних заняттях з дисципліни «Обробка і аналіз інформації» для студентів напряму підготовки 6.040106 «Екологія, охорона навколишнього середовища та збалансоване природокористування».

Дипломний проект допоможе у розрахунку критерію суми рангів Уилкоксона для інверсійної і рангової обробки даних.

На першій стадії розробки диплома було виконано:

- аналітичний огляд предметної області, а саме аналіз онлайн калькуляторів розрахунку показника Вілкоксона для непараметричних даних;
- за результатами аналізу сформовані вимоги і задачі до об'єкту розробки;
- обрані програмні засоби для реалізації програмного додатку.

На етапі проектування виконано:

- побудовані UML-діаграми варіантів використання і діяльності для об'єкту розробки;
- розроблено алгоритм для паралельної обробки вхідних даних.

Під час розробки були запрограмовані такі вимоги до проекту, як:

- реалізація алгоритму розрахунку даних;
- розробка оптимального інтерфейсу користувача для полегшення вводу інформації;
- реалізація функції «створити звіт розрахунків» для подальшого аналізу результатів;
- розробка додаткових функцій для паралельної обробки великих рядків даних з метою прискорення часу обчислення.

Подальшим розвитком проекту буде доповнення алгоритму функціями для обчислювання іншими методами, які використовують студенти ОДЕКУ під час свого навчання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Інформаційний ресурс Medstatistic. URL: <http://medstatistic.ru/theory/wilcoxon.html>. (да-та звернення 02.03.2019).
2. Інформаційний ресурс Math.semestr. URL: <https://math.semestr.ru/group/wilcoxon.php>. (дата звернення 01.03.2019).
3. Інформаційний блог з програмування Elizarov. URL: <https://elizarov.livejournal.com/27125.html>. (дата звернення 10.03.2019).
4. Інформаційний ресурс Shiftoffproblem з роботи у Visual Studio. URL: <http://shiftoffproblem.com/visual-studio-community-2015-review/>. (дата звернення 11.03.2019).
5. Інформаційний ресурс з програмування. URL: <http://cppstudio.com/post/9384/>. (11.03.2019).
6. Інформаційний портал з програмування. URL: <https://habr.com/ru/post/352096/>. (дата звернення 13.03.2019).
7. Інформаційний блог з програмування. URL: <https://eax.me/tag/winapi/>. (дата звернення 13.03.2019).
8. Інформаційний ресурс електронної бібліотеки Pidruchniki. URL: https://pidruchniki.com/12590605/statistika/statistichni_kriteriyi. (дата звернення 15.03.2019).
9. Лосєва І. Д., Бургаз О.А. Збірник методичних вказівок до практичних занять з дисципліни «Обробка і аналіз інформації» для студентів денної форми навчання, напрям підготовки 6.040106 «Екологія, охорона навколишнього середовища та збалансоване природокористування». Одеса:ОДЕКУ, 2010. 65 с.
10. Електронна бібліотека Studopedia. URL: https://studopedia.com.ua/1_47878-statistichniy-kriteriy-perevirki-osnovnoi-gipotezi.html. (дата звернення 20.03.2019).

11. Електронна бібліотека Westudents. URL: <https://westudents.com.ua/glavy/88649-parametrichn-neparametrichn-kriter.html>. (дата звернення 23.03.2019).
12. Офіційний сайт UML/ URL: <https://www.uml.org/>. (дата звернення 28.03.2019).
13. Інформаційний ресурс Fgithub. URL: https://flexberry.github.io/ru/fd_activity-diagram.html. (дата звернення 29.03.2019).
14. Інформаційний ресурс з UML-моделювання. URL: <https://www.uml-diagrams.org/use-case-diagrams.html>. (дата звернення 01.05.2019).
15. Інформаційний ресурс з UML-моделювання. URL: www.agilemodeling.com/artifacts/useCaseDiagram.htm. (дата звернення 01.05.2019).
16. Інформаційний ресурс Habr. URL: <https://habr.com/ru/post/208966/>. (дата звернення 03.05.2019).
17. Інформаційний ресурс з розробки програмного забезпечення. URL: <https://medium.com/razrabotka-maketa-interfejsa-programm-osnovnie-pravila>. (дата звернення 05.05.2019).