

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І.І.МЕЧНИКОВА

(повне найменування вищого навчального закладу)

Факультет математики, фізики та інформаційних технологій

(повне найменування інституту/факультету)

Кафедра механіки, автоматизації та інформаційних технологій

(повна назва кафедри)

## Кваліфікаційна робота

на здобуття ступеня вищої освіти «магістр»

« Використання факторного аналізу для вирішення задачі планування  
роботи підприємства »

« Use of factor analysis to solve the task of planning the  
work of the enterprise »

Виконав: здобувач денної форми навчання  
спеціальності 113 Прикладна математика  
Освітня програма Теоретична та прикладна механіка  
Яроцький Даніель Сергійович

Керівник доктор тех. наук, професор Волков В.Е.  
(науковий ступінь, вчене звання, прізвище та ініціали, підпис)

Рецензент кандидат тех. наук, Косой М.Б.  
(науковий ступінь, вчене звання, прізвище та ініціали)

Рекомендовано до захисту:

Протокол засідання кафедри

№ \_\_\_\_\_ від \_\_\_\_\_ 2023 р.

Захищено на засіданні ЕК №

протокол № \_\_\_\_\_ від \_\_\_\_\_ 2023 р.

Оцінка \_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_  
(за національною шкалою, шкалою ECTS, бали)

Завідувачка кафедри

Голова ЕК

Алла РАЧИНСЬКА

(підпис)

Ольга КІЧМАРЕНКО

(підпис)

## АНОТАЦІЯ

У дипломній роботі розробляється тема «Використання факторного аналізу для вирішення задачі планування роботи підприємства».

Метою роботи є визначення параметрів судна яке проектується в першому наближенні на основі факторного аналізу статистичних даних, аналіз даного методу, написання на його основі комп'ютерної програми для обчислень факторів.

В результаті проведеної роботи вирішені такі задачі:

1. Тестування методу факторного аналізу на першій вибірковій групі (лоцманські катери).
2. Написання обчислювальної програми-дodatка для факторного аналізу.
3. Тестування проекту на другий вибірковій групі (балкери).
4. Проведено аналіз отриманих результатів.

## ABSTRACT

In the diploma work, the topic «Use of factor analysis to solve the task of planning the work of the enterprise» is developed.

The purpose of the work is determination of the parameters of the vessel which is designed in the first approximation on the basis of the factor analysis of statistical data, the analysis of this method and the writing of a computer program for factor calculations based on it.

As a result of the work, the following tasks were solved:

1. Testing the method of factor analysis on the first sample group (pilot boats).
2. Writing a computer program-application for factor analysis.
3. Application testing for the second sample group (bulkiers)
4. An analysis of the obtained results was carried out.

## ЗМІСТ

	стор.
ВСТУП.....	5
1 МЕТОДИ ДОСЛІДЖЕННЯ ХАРАКТЕРИСТИК СУДНА .....	7
1.1 Огляд літератури по темі дипломної роботи .....	7
1.2 Об'єкти дослідження.....	8
1.3 Метод головних компонент .....	10
2 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ АНАЛІЗУ .....	15
2.1 Середовище розробки програми обчислення .....	15
2.2 Мова програмування програмного продукту .....	16
2.3 Використані бібліотеки мови програмування.....	28
2.4 Алгоритм роботи програмного продукту.....	34
3 РЕЗУЛЬТАТИ АНАЛІЗУ ДАННИХ .....	36
3.1 Алгоритм застосування методу факторного аналізу для вибору характеристик судна .....	41
3.2 Тестування роботи програмного продукту .....	43
ВИСНОВКИ .....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	51
ДОДАТОК А. КОД ПРОГРАМНОГО ПРОДУКТУ .....	52
ДОДАТОК Б. ТАБЛИЦІ ХАРАКТЕРИСТИК СУДЕН.....	57

## ВСТУП

**Актуальність теми.** Проектування суден є комплексною інженерною задачею високого рівня складності. В процесі вирішення цієї задачі

чі необхідно враховувати особливості судна, що проектується, можливі проектні обмеження, умови експлуатації, подолати об'єктивну обмеженість інформації особливо при прийнятті проектних рішень на початковій стадії проектування.

Задача проектування судна може бути зведена до задачі оптимізації характеристик судна по одному вибраному критерію. В якості критеріїв зазвичай можуть бути використані економічні (чистий середньорічний прибуток, наведені витрати на утримання судна, провозоспроможність і т.і. [1]) або комплексні технічні параметри (ходовість, морехідність, витрати палива на 1 тонну вантажу, що перевозиться і т.і. ). Тоді після формування математичної моделі судна, що враховує в явному вигляді зазначені критерії, формується завдання мінімізації певних обраних критеріїв і визначення характеристик судна. Рішення такого завдання залежить від виду рівнянь і може бути виконано відомими чисельними методами [2, 3, 4].

Практична цінність рішення задачі оптимізації залежить від якості математичної моделі, в яку для отримання корисного результату необхідно вносити доповнення та виправлення. Варіюючи параметрами моделі можна отримати кілька наборів характеристик судна, і виявити з них найбільш відповідний умовам технічного завдання.

Інший шлях передбачає вибір основних характеристик судна на основі близького прототипу. Рішення задачі визначення основних характеристик судна на початковому етапі проектування ускладнюється браком інформації, необхідної для прийняття обґрунтованих проектних рішень. Одним із шляхів, які дозволяють отримати відсутню інформацію про характеристики судна, є спосіб проектування по прототипу. Вибір прототипу або декількох найбільш

прийнятних прототипів зазвичай здійснюється експертом або групою експертів на підставі наявного досвіду, знань, доступної інформації з об'єкту, який викликає зацікавленість. Важливе значення має наявність достовірної великої бази знань по суднам, повнота інформації [5]. Один із способів обчислення цих характеристик є метод факторного аналізу. Завдання такого рівня дуже складно вирішувати вручну. Для отримання факторів для одного конкретного типу судна знадобиться багато часу, адже для їх обчислення використовуються не найпростіші алгебри та матричних обчислень. На виробництві завжди зручніше використовувати готовий калькулятор. У цій дипломній роботі я докладно опишу принцип роботи програми, призначеної для обчислення факторів, виведених на основі інженерних характеристик судна.

**Ціль та задачі дослідження.** Метою роботи є обчислення факторів, виведених на основі інженерних характеристик судна.

Для досягнення поставленою мети необхідно розв'язати наступні задачі:

1. Проаналізувати існуючі бази даних характеристик лоцманських суден та балкерів.
2. Зробити багато параметричну математичну модель розрахунку зазначених факторів.
3. Обрати середу розробки програмного продукту.
4. Розробити технологію розрахунку у вигляді програмного продукту.
5. Провести аналіз отриманих результатів.

Об'єкт дослідження дипломної роботи – планування роботи підприємства з розробки суден.

Предмет дослідження – параметри судна у стадії проектування в першому наближенні.

## 1. МЕТОДИ ДОСЛІДЖЕННЯ ХАРАКТЕРИСТИК СУДНА

Задачу визначення параметрів судна яке проектується в першому наближенні на основі факторного аналізу статистичних даних можна сформулювати наступним чином: знайти такі комбінації спостережуваних параметрів, які дозволяють виявити характеристики і якості судна, оптимальні в певному сенсі. Інтерпретація знайдених комбінацій-факторів є не менш важливою проблемою і вирішується не в рамках статистичного аналізу, а на змістовному предметному рівні. Завдання ускладнюється ще й тим, що складно отримати вичерпну інформацію по характеристикам суден, таких як теоретичний креслення, криві елементів теоретичного креслення і т.п., в силу комерційних чи інших обмежень.

### 1.1. Огляд літератури по темі дипломної роботи

Задачам планування роботи підприємства присвячені роботи А.Н. Вашедченко, В.А. Некрасова, Н.Н. Кабанової, М.Э. Францева, А.Н. Пинского, В.М. Жуковскої, И.Б. Мучника та інших. Задачам використання факторного аналізу присвячені роботи Дж.-О. Кіма, Ч.У. Мюллера, У.Р. Клекка, М.С. Олдендерфера, Р.К. Блешфілда, Лоули Д., Максвелл А. та інших.

Робота А.Н. Вашедченко присвячена задачі автоматизування проектування судів. У наукових працях В.А. Некрасов та Н.Н. Кабанова описане функціонування судна, що експлуатується на незакріплених лініях перевезень та проблема його проектування. У науковій праці Н.Н. Кабанова описано рішення стохастичної задачі оптимізації характеристик трампового судна. Використання чисельних методів при реалізації завдання параметричного проектування композитного промислового судна для прибережної ловлі та Проектному обґрунтуванню оптимальних поєднань характеристик маси, потужності та швидкості для швидкісних суден з композитів методами аналізу баз даних присвячені книги М.Э. Францева. Використання статистичних методів в алгоритмах автоматизованої системи

проектування описані в працях А.Н. Пинского. В.М. Жуковская та И.Б. Мучник у своїх книгах досліджували факторний аналіз у соціально-економічних дослідженнях. Факторний, дискримінантний та кластерний аналіз описані у наукових працях Дж.-О. Кіма, Ч.У. Мюллера, У.Р. Клекка, М.С. Олдендерфера та Р.К. Блешфілда. Вперше факторний аналіз як статистичний метод був описаний у книзі Лоулі Д. та Максвелл А у 1967 році.

## 1.2. Об'єкти дослідження

В якості об'єктів дослідження вибрані дві групи суден:

1 вибіркова група - лоцманські катери з корпусами та надбудовами з алюмінієво-магнієвого сплаву;

2 вибіркова група - судна для перевезення масових вантажів навалом (балкери).

В процесі роботи були проаналізовані характеристики 40 проектів існуючих лоцманських катерів, які представляють типові групи лоцманських суден різних виробників (проектантів).

Також для аналізу були обрані характеристики 32 балкерів існуючих проектів 2000 – 2017 років побудови, які за зростанням значення дедвейту розділяються на такі основні класи: HANDYSIZE, HANDYMAX, PANAMAX, KAMSARMAX (larger-size Panamax bulk carrier), POST-PANAMAX та CAPESIZE.

Дані розглянутих проектів лоцманських суден представлені в таблиці 1, основні характеристики балкерів наведені в таблиці 2.

В подальшому прийняті наступні позначення безрозмірних параметрів:

– для обробки даних по лоцманським суднам:

$$X_1 = \frac{L}{B}, X_2 = \frac{B}{H}, X_3 = \frac{H}{T}, X_4 = \frac{D}{\rho LBT}, X_5 = \frac{V}{\sqrt{gL}}, X_6 = \frac{P}{\rho g DV},$$

– для обробки даних по балкерам:



$$X_1 = \frac{L}{B}, X_2 = \frac{B}{H}, X_3 = \frac{D_b}{\rho L B T}, X_4 = \frac{D_b}{\rho L B H}, X_5 = \frac{D_b - D_n}{D_b}, X_6 = \frac{V}{\sqrt{g^3 \sqrt{\frac{D_b}{\rho}}}}, X_7 = \frac{P}{\rho g D_b V}$$

Тут  $L, B, H, T$  – головні розміри судна – довжина, ширина, висота борту і середнє заглиблення відповідно, м;

$D$  – вага судна, т;

$D_b$  – вага судна при повному завантаженні, т;

$D_n$  – вага порожнього судна, т;

$P$  – потужність сумарна головної рушійної установки, кВт;  $V$  – швидкість найбільша, вуз.;  $g = 9.81$  - гравітаційна константа, м/с<sup>2</sup>;  $\rho$  - вагова щільність морської води.

Вибір зазначених вище параметрів обумовлений обмеженням числом достовірних даних про судна.

Найбільш вивченою є лінійна модель залежності спостережуваних характеристик від факторів у вигляді

$$X_i = \sum_j^n \alpha_{ij} F_j + \beta_i U_i, i = 1, n \quad (1)$$

де  $F_j$  - визначальні фактори, загальні для всіх параметрів  $X_i$ , а  $U_i$  - фактор, характерний виключно для  $i$ -ого параметра. Характерні фактори як правило відображають вплив зовнішніх обставин на параметри системи, загальні ж фактори характеризують структуру внутрішніх взаємозалежностей, яка не проглядається явно. З'ясування природи характерних факторів в рамках факторного аналізу не вирішується. Таким чином, статистичний факторний аналіз зводиться до двох великих задач, для вирішення яких розроблені різні

методики. Перша - визначення факторів  $F_j$ , друга - визначення коефіцієнтів  $\alpha_{ij}$  — факторних навантажень, які визначають ступінь впливу

кожного фактору на значення параметрів  $X_i$ . Ці задачі є сполученими, тобто вирішення однієї з них визначає і рішення другої. Цілком логічним видається можливість надати пріоритет вирішенню першого завдання, оскільки це рішення пов'язане з проблемою інтерпретації факторів.

### 1.3. Метод головних компонент

Для визначення структури факторів можна використовувати метод головних компонент. Хоч система (1) містить стільки ж факторів, скільки і вхідних параметрів, але не всі вони мають рівнозначну ступінь впливу на значення досліджуваних параметрів. Можна обмежитися вибором найбільш впливових факторів, тобто число факторів буде менше числа вхідних параметрів. Ця обставина й дозволяє зробити предметний аналіз вже якимось чином систематизованих вхідних даних і застосувати метод групування для вирішення головного питання дослідження, а саме як вибрати в першому наближенні параметри судна за даними прототипів

Залишивши в (1) лише залежність від загальних факторів, лінійним перетворенням отримаємо

$$F_j = \sum_i^m \alpha_{ji} X_i, j = 1, m \quad (2)$$

Ця форма подання незручна, тому що різні величини і коефіцієнти повинні бути розмірними. Тому прийняті до обробки параметри . Тоді лінійна форма (2) залежності факторів перетвориться до виду

$$F_j = \sum_i^n \alpha_{ji} Y_i, j = 1, n \quad (3)$$

Кожен фактор  $F_j$  і кожен параметр  $X_i$  є вектори розмірності  $N$  - довжини вибірки суден, тобто вираз (3) в розгорнутій формі має вигляд

$$F_{jk} = \ln(\prod_{i=1}^n X_{ik}^{\alpha_{ji}}) = \sum_{i=1}^n \alpha_{ji} Y_{ik}, k = 1, N \quad (4)$$

Лінійна форма (1) матиме вигляд (без урахування доданків характерних факторів)

$$Y_i = \sum_i^n b_{ij} F_j, j = 1, n \quad (5)$$

Така комбінація у вигляді твору ступенів параметрів (4) узгоджується з основним принципом теорії розмірностей і може бути інтерпретована як деяка якість конструкції судна або як його експлуатаційна якість. Використання методу головних компонент передбачає статистичну незалежність факторів. Для зручності викладу позначимо через  $\|Y\|_{n \times N}$  матрицю значень параметрів  $Y_i$  для вибірки довжини  $N$ , через  $\|b\|_{n \times n}$  матрицю факторних ваг, через  $\|F\|_{n \times N}$  матрицю  $N$  значень  $n$  факторів. Тоді (5) має вигляд  $\|Y\| = \|b\| \|F\|$  або  $\|Y\|^T = \|F\|^T \|b\|$ .

Розглянемо матричний твір  $\|Y\|^T \|Y\| = \|b\| \|F\|^T \|F\| \|b\|^T$ . У термінах цього твору можна ввести стохастичні характеристики системи:

$$K_Y = \left\| \frac{(Y_j, Y_k)}{N-1} \right\| \text{ — кореляційна матриця параметрів;}$$

$$K_F = \left\| \frac{(F_j, F_k)}{N-1} \right\| \text{ — кореляційна матриця факторів, де } (F_j, F_k) = \sum_i^N F_{ij} F_{ki}$$

— скалярний твір. З огляду на (4), справедливо матричне співвідношення

$$K_Y = \frac{\|Y_{ji}\| \|Y_{ji}\|^T}{N-1} = \|b\| \frac{\|F_{ji}\| \|F_{ji}\|^T}{N-1} \|b\|^T = \|b\| K_F \|b\|^T \quad (6)$$

Так як фактори невідомі, можна накласти на їх властивості деякі обмеження. Якщо зажадати, щоб фактори були статистично незалежні, тобто матриця  $K_F$  діагональна, а  $\|b\|^{-1} = \|b\|^T$ , то тоді (6) представляє канонічний розклад матриці  $K_Y = \|b\| K_F \|b\|^T$ . Тобто матриця  $\|b\|$  — це матриця компонентів власних векторів матриці  $K_Y$ , а діагональні елементи  $\lambda_j$  матриці  $K_F$  — це власні числа  $K_Y$ .

Значить, треба знайти ці характеристики матриці  $K_Y$  методами матричної алгебри і отримати канонічне подання

$$K_Y = \|b\| \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix} \|b\|^{-1} \quad (7)$$

У термінах теорії ймовірності та статистики діагональні елементи матриці  $K_Y$  є дисперсії факторів  $F_j$ . Ранжування власних значень матриці за величиною від більшого до меншого визначає значимість впливу кожного фактора на параметри судна: чим більше власне значення  $\lambda_j$ , тобто дисперсія фактора  $F_j$ , тим він інтенсивніше впливає на значення параметра в статистичній залежності.

Методи статистичної оцінки значущості цього впливу дозволяють не брати до уваги фактори що слабо проявляються, і тим спростити та зробити доступним для огляду предметний аналіз вибірки суден. Однак це спрощення пов'язане з втратою інформації. Якщо ж всі власні значення мають порівнянні величини, то враховувати доводиться всю їх сукупність. Щоб не втрачати інформації і не відкидати чинники з малими значеннями дисперсій (їм відповідають малі значення власних чисел матриці  $K_Y$ ), будемо вважати, що система факторів ще й нормована. Тоді отримуємо канонічну форму  $K_Y$  у вигляді

$$K_Y = \|b\| \|\sqrt{\lambda_k}\| \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} \|\sqrt{\lambda_j}\| \|b\|^{-1} \quad (8)$$

Таким чином,  $\|a_{ik}\| = \|b_{ik}\|^{-1} \|\sqrt{\lambda_i}\|^{-1}$  та можна користуватися (3) для обчислення факторів.

Щоб все ж мати можливість спростити систему за рахунок зменшення

факторів необхідно обчислити кореляційну матрицю  $K_{FY} = \left\| \frac{(F_j, Y_k)}{N-1} \right\|$ , де  $(F_j, F_k) = \sum_i F_{ij} F_{kl}$  — скалярний твір  $F_j$  на  $Y_k, l = 1, N$ . У матричній формі  $K_{FY} = \|k_{ij}\|$ .

можна представити таким чином:

$$K_{FY} = \frac{\|a_{ik}\| \|Y_{kl}\| \|Y_{kl}\|^T}{N-1} \quad (9)$$

Оцінку значущості значень елементів кореляції проводять за t-критерієм Стьюдента з вірогідністю, наприклад, 5%:  $\frac{|k_{ij}\sqrt{N-2}|}{\sqrt{1-k_{ij}^2}} > 2.8$ . Для більшої наочності результати значущості заносяться в матрицю інцидентності  $IK_{FY} = \|\gamma_{ij}\|$ : якщо нерівність виконується, то вважають кореляцію  $k_{ij}$  параметра  $i$  фактора не нульовою і  $\|\gamma_{ij}\| = 1$ , якщо не виконується - то вважають, що параметри  $Y_t$  і фактор  $F_j$  статистично незалежні і  $\gamma_{ij} = 0$ . І тому, в розкладанні (3) відповідний елемент  $a_{ij} = 0$ , тобто в (3) слід прийняти нову матрицю  $\|a_{ij}^*\| = \|a_{ij}\gamma_{ij}\|$ .

Знайдені значення факторів для суден дозволяють зробити угруповання. Метод цього угруповання полягає в наступних кроках:

- задати інтервали зміни значень обраних факторів;
- згрупувати судна вибірки за сукупними значеннями факторів;
- вибрати найбільш численні групи і помітити для них інтервали значень факторів (численність в групі свідчить про спільну для суден групи не проявлену якість);
- якщо  $m$  - число прийнятих факторів, а  $n$  - число параметрів, і  $n > m$ , то з  $n$  невідомих величин буде лише  $(n-m)$  незалежних, інші зв'язані співвідношеннями (3) або (4) для факторів з прийнятими значеннями.

Незалежні параметри вибираються за результатами предметного дослідження, а залежні, в силу необхідності, будуть визначатися таким чином, що забезпечать деяку оптимальну якість судна. Звичайно, з'ясування цієї якості виходить за рамки статистичного аналізу даних.

## 2. ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ОБЧИСЛЕННЯ

Обчислення будуть проводитись для другої тестової групи судів типу балкери. Для створення програми будемо використовувати мову програмування пайтон та інтегроване середовище розробки PyCharm. Мова програмування та бібліотеки для нього були обрані таким чином, щоб написання програми не вимагало сторонніх IDE.

### 2.1. Середовище розробки програми обчислення

Для створення програми будемо використовувати інтегроване середовище розробки PyCharm. PyCharm — це інтегроване середовище розробки (IDE), яке використовується для програмування на Python. Він забезпечує аналіз коду, графічний налагоджувач, вбудований модульний тестер, інтеграцію з системами контролю версій і підтримує веб-розробку за допомогою Django. PyCharm розроблено чеською компанією JetBrains.

Він кросплатформний, працює на Microsoft Windows, macOS і Linux. PyCharm має професійну версію, випущену за пропрієтарною ліцензією, і видання спільноти, випущену за ліцензією Apache. Версія PyCharm Community Edition менш обширна, ніж професійна версія.

PyCharm було випущено на ринок IDE, орієнтованих на Python, щоб конкурувати з PyDev (для Eclipse) або більш широко орієнтованою IDE Komodo від ActiveState.

Бета-версія продукту була випущена в липні 2010 року, а версія 1.0 надійшла через 3 місяці. Версія 2.0 була випущена 13 грудня 2011 року, версія 3.0 була випущена 24 вересня 2013 року, а версія 4.0 була випущена 19 листопада 2014 року.

PyCharm став відкритим кодом 22 жовтня 2013 року. Варіант із відкритим кодом випущено під назвою Community Edition, тоді як комерційний варіант, Professional Edition, містить модулі із закритим кодом.

Особливості:

- Допомога та аналіз кодування, із доповненням коду, підсвічуванням синтаксису та помилок, інтеграцією Linter та швидкими виправленнями
- Навігація проектом і кодом: спеціалізовані види проектів, перегляди структури файлів і швидкий перехід між файлами, класами, методами та використаннями
- Рефакторинг коду Python: включаючи перейменування, вилучення методу, введення змінної, введення константи, підтягування, опускання та інші
- Підтримка веб-фреймворків: Django, web2py і Flask
- Інтегрований налагоджувач Python
- Розробка Google App Engine на Python
- Інтеграція контролю версій: уніфікований інтерфейс користувача для Mercurial, Git, Subversion, Perforce і CVS зі списками змін і злиттям
- Інтеграція наукових інструментів: інтегрується з IPython Notebook, має інтерактивну консоль Python і підтримує Anaconda, а також численні наукові пакети, включаючи Matplotlib і NumPy.

## 2.2. Мова програмування програмного продукту

Для створення програми будемо використовувати мову програмування Python. Python — це мова програмування високого рівня загального призначення. Його філософія дизайну наголошує на зручності читання коду з використанням значних відступів.

Python динамічно типізується та збирає сміття. Він підтримує кілька парадигм програмування, включаючи структуроване (зокрема процедурне), об'єктно-орієнтоване та функціональне програмування. Його часто описують як мову «включно з батареjkами» через його повну стандартну бібліотеку.

Гвідо ван Россум почав працювати над Python наприкінці 1980-х як наступником мови програмування ABC і вперше випустив її в 1991 році як



Python 0.9.0. Python 2.0 був випущений у 2000 році. Python 3.0, випущений у 2008 році, був основною версією, не повністю сумісною з попередніми версіями. Python 2.7.18, випущений у 2020 році, був останнім випуском Python 2.

Python був задуманий наприкінці 1980-х років Гвідо ван Россумом із Centrum Wiskunde & Informatica (CWI) у Нідерландах як наступник мови програмування ABC, яка була натхненна SETL, здатною обробляти винятки та взаємодіяти з операційною системою Amoeba. Його реалізація почалася в грудні 1989 року. Ван Россум ніс виключну відповідальність за проект як провідний розробник до 12 липня 2018 року, коли він оголосив про свою «постійну відпустку» від своїх обов'язків «доброзичливого диктатора Python на все життя», титул Python спільнота надала йому, щоб відобразити його довгострокову прихильність як головної особи, яка приймає рішення в проекті. У січні 2019 року активні розробники ядра Python обрали Керівну раду з п'яти членів для керівництва проектом.

Python 2.0 було випущено 16 жовтня 2000 року з багатьма основними новими функціями, такими як розуміння списку, збирання сміття з визначенням циклу, підрахунок посилань і підтримка Unicode. Python 3.0, випущений 3 грудня 2008 року, з багатьма основними функціями, перенесеними на Python 2.6.x і 2.7.x. Випуски Python 3 включають утиліту 2to3, яка автоматизує переклад коду Python 2 на Python 3.

Кінець терміну служби Python 2.7 спочатку був призначений на 2015 рік, а потім перенесений на 2020 рік через занепокоєння, що великий обсяг існуючого коду не можна буде легко перенести на Python 3. Подальші виправлення безпеки чи інші вдосконалення випускатися не будуть для нього. Наразі підтримуються лише версії 3.8 і новіші (2023 проблеми безпеки було виправлено, наприклад, у версії 3.7.17, останньому випуску 3.7.x).

У 2021 році (і знову двічі в 2022 році) оновлення системи безпеки були прискорені, оскільки всі версії Python були незахищеними (включно з 2.7) через

проблеми з безпекою, що призводило до можливого віддаленого виконання коду та отруєння веб-кешу. У 2022 році Python 3.10.4 і 3.9.12 були прискорені, а також 3.8.13 через багато проблем із безпекою. Коли Python 3.9.13 було випущено в травні 2022 року, було оголошено, що серія 3.9 (приєднавшись до старих серій 3.8 і 3.7) отримає лише виправлення безпеки в майбутньому. 7 вересня 2022 року було випущено чотири нові випуски через потенційну атаку на відмову в обслуговуванні: 3.10.7, 3.9.14, 3.8.14 і 3.7.14.

Станом на жовтень 2023 року Python 3.12 є стабільним випуском, а 3.12 і 3.11 є єдиними версіями з активною підтримкою (на відміну від просто безпеки). Помітні зміни в 3.11 порівняно з 3.10 включають збільшення швидкості виконання програми та покращене звітування про помилки.

Python 3.12 додає синтаксис (і фактично кожен Python, починаючи принаймні з 3.5, додає певний синтаксис) до мови, новий (м'який) тип ключового слова (в останніх випусках додано багато підтримки введення, наприклад, новий оператор об'єднання типів у 3.10), і 3.11 для обробки виключних ситуацій, і 3.10 ключові слова `match` і `case (soft)` для операторів відповідності структурного шаблону. Python 3.12 також видаляє застарілі модулі та функціональні можливості, і майбутні версії також будуть, дивіться нижче в розділі «Розробка».

Python 3.11 стверджує, що на 10–60% швидший за Python 3.10, а Python 3.12 додає ще 5% на додаток до цього. Він також має покращені повідомлення про помилки та багато інших змін.

З 27 червня 2023 року Python 3.8 є найстарішою підтримуваною версією Python (хоча й на етапі «підтримки безпеки») через те, що термін служби Python 3.7 завершується.

Python незмінно вважається однією з найпопулярніших мов програмування.

Python є багатопарадигмальною мовою програмування. Об'єктно-орієнтоване програмування та структуроване програмування повністю

підтримуються, і багато їхніх функцій підтримують функціональне та аспектно-орієнтоване програмування (включаючи метапрограмування та метаоб'єкти). Багато інших парадигм підтримуються через розширення, включаючи проектування за контрактом і логічне програмування.

Python використовує динамічну типізацію та комбінацію підрахунку посилань і збирача сміття, що визначає цикл, для керування пам'яттю. Він використовує динамічне розпізнавання імен (пізні зв'язування), яке зв'язує імена методів і змінних під час виконання програми.

Його дизайн пропонує певну підтримку функціонального програмування в традиції Lisp. Він має функції фільтра, відображення та зменшення; розуміння списків, словники, набори та вирази генератора. Стандартна бібліотека має два модулі (`itertools` і `functools`), які реалізують функціональні інструменти, запозичені з Haskell і Standard ML.

Його основна філософія коротко викладена в документі The Zen of Python (PEP 20), який містить такі афоризми, як:

- Красиве краще, ніж потворне.
- Явне краще, ніж неявне.
- Просте краще, ніж складне.
- Складне краще, ніж складне. (Complex is better than complicated.)
- Читабельність має значення.

Замість того, щоб побудувати всю свою функціональність у своєму ядрі, Python був розроблений таким чином, щоб бути розширюваним за допомогою модулів. Ця компактна модульність зробила його особливо популярним як засіб додавання програмованих інтерфейсів до існуючих програм. Бачення Ван Россума невеликої базової мови з великою стандартною бібліотекою та інтерпретатором, що легко розширюється, виникло через його розчарування ABC, яка підтримувала протилежний підхід.

Python прагне до простішого, менш захащеного синтаксису та граматики, надаючи розробникам можливість вибору методології кодування.

На відміну від девізу Perl «є більше, ніж один спосіб зробити це», Python приймає філософію «повинен бути один — і бажано тільки один — очевидний спосіб зробити це». Алекс Мартеллі, співробітник Python Software Foundation і автор книжки про Python, написав: «Описати щось як «розумне» не вважається компліментом у культурі Python».

Розробники Python прагнуть уникати передчасної оптимізації та відкидають виправлення некритичних частин еталонної реалізації CPython, які забезпечуватимуть незначне збільшення швидкості за рахунок ясності. Швидкість виконання можна покращити, перемістивши критичні для швидкості функції до модулів розширення, написаних мовами, наприклад C, або за допомогою своєчасного компілятора, наприклад PyPy. Також можлива крос-компіляція з іншими мовами, але вона або не забезпечує повного прискорення, яке можна було очікувати, оскільки Python є дуже динамічною мовою, або компілюється обмежена підмножина Python, і, можливо, семантика трохи змінений. Розробники Python прагнуть, щоб ним було цікаво користуватися. Це відображено в його назві — данині британській комедійній групі Monty Python — і в іноді грайливих підходах до посібників і довідкових матеріалів, таких як використання термінів «спам» і «яйця» (посилання на скетч Monty Python) у прикладах замість часто вживаних "foo" і "bar".

Поширеним неологізмом у спільноті Python є *pythonic*, який має широкий діапазон значень, пов'язаних зі стилем програми. «Pythonic» код може добре використовувати ідіоми Python, бути природним або демонструвати вільне володіння мовою, або відповідати мінімалістичній філософії Python і наголосу на читабельності. Код, який важко зрозуміти або читається як груба транскрипція з іншої мови програмування, називається *непітонічним*.

Python має бути мовою, яку легко читати. Його форматування візуально чисте та часто використовує англійські ключові слова, тоді як інші мови використовують розділові знаки. На відміну від багатьох інших мов, у ній не використовуються фігурні дужки для розмежування блоків, а крапки з комою

після операторів дозволені, але рідко використовуються. Він має менше синтаксичних винятків і особливих випадків, ніж C або Pascal.

Python використовує пробільні відступи, а не фігурні дужки чи ключові слова для розмежування блоків. Збільшення відступу відбувається після певних тверджень; зменшення відступу означає кінець поточного блоку. Таким чином, візуальна структура програми точно відображає її семантичну структуру. Цю функцію іноді називають правилом офсайду. Деякі інші мови використовують відступ таким чином; але в більшості випадків відступ не має семантичного значення. Рекомендований розмір відступу – чотири інтервали.

Інструкції Python включають:

- Оператор присвоєння з використанням одного знака рівності =
- Інструкція if, яка умовно виконує блок коду разом із else та elif (скорочення else-if)
- Інструкція for, яка виконує ітерацію по ітераційному об'єкту, захоплюючи кожен елемент у локальну змінну для використання приєднаним блоком
- Інструкція while, яка виконує блок коду, доки його умова виконується
- Інструкція try, яка дозволяє перехоплювати та обробляти винятки, викликані у доданому блоці коду, за допомогою пропозицій, крім (або нового синтаксису, крім у Python 3.11 для груп винятків); це також гарантує, що код очищення в блоці finally завжди виконується незалежно від того, як блок виходить
- Інструкція raise, яка використовується для виклику визначеного винятку або повторного виклику спійманого винятку
- Інструкція класу, яка виконує блок коду та приєднує його локальний простір імен до класу для використання в об'єктно-орієнтованому програмуванні
- Інструкція def, яка визначає функцію або метод

- Оператор `with`, який охоплює блок коду в контекстному менеджері (наприклад, отримання блокування перед його запуском, а потім звільнення блокування; або відкриття та закриття файлу), дозволяючи отримання ресурсу-ініціалізацію (RAII)- подібна поведінка та заміна загальної ідіоми `try/finally`
  - Інструкція `break`, яка виходить із циклу
  - Оператор `continue`, який пропускає решту поточної ітерації та продовжує наступну
  - Оператор `del`, який видаляє змінну, видаляючи посилання з назви на значення та створюючи помилку, якщо на змінну посилаються до її перевизначення
    - Інструкція `pass`, яка служить NOP, синтаксично необхідна для створення порожнього блоку коду
    - Оператор `assert`, який використовується під час налагодження для перевірки умов, які мають застосовуватися
    - Оператор `yield`, який повертає значення з функції-генератора (а також оператора); використовується для реалізації співпрограм
    - Оператор `return`, який використовується для повернення значення з функції
    - Інструкції `import` і `from`, які використовуються для імпорту модулів, функції або змінні яких можна використовувати в поточній програмі

Оператор присвоєння (`=`) прив'язує ім'я як посилання на окремий динамічно виділений об'єкт. Змінні згодом можуть бути перенаправлені в будь-який час до будь-якого об'єкта. У Python ім'я змінної є загальним власником посилання без фіксованого типу даних; однак він завжди посилається на якийсь об'єкт із типом. Це називається динамічною типізацією — на відміну від мов зі статичною типізацією, де кожна змінна може містити лише значення певного типу.

Python не підтримує оптимізацію хвостового виклику або продовження першого класу, і, за словами Ван Россума, він ніколи не підтримує. Проте краща

підтримка функціональності, схожої на співпрограму, забезпечується розширенням генераторів Python. До версії 2.5 генератори були лінивими ітераторами; дані були передані в одному напрямку з генератора. Починаючи з Python 2.5, можна передавати дані назад у функцію генератора; а починаючи з версії 3.3, його можна передавати через кілька рівнів стеку.

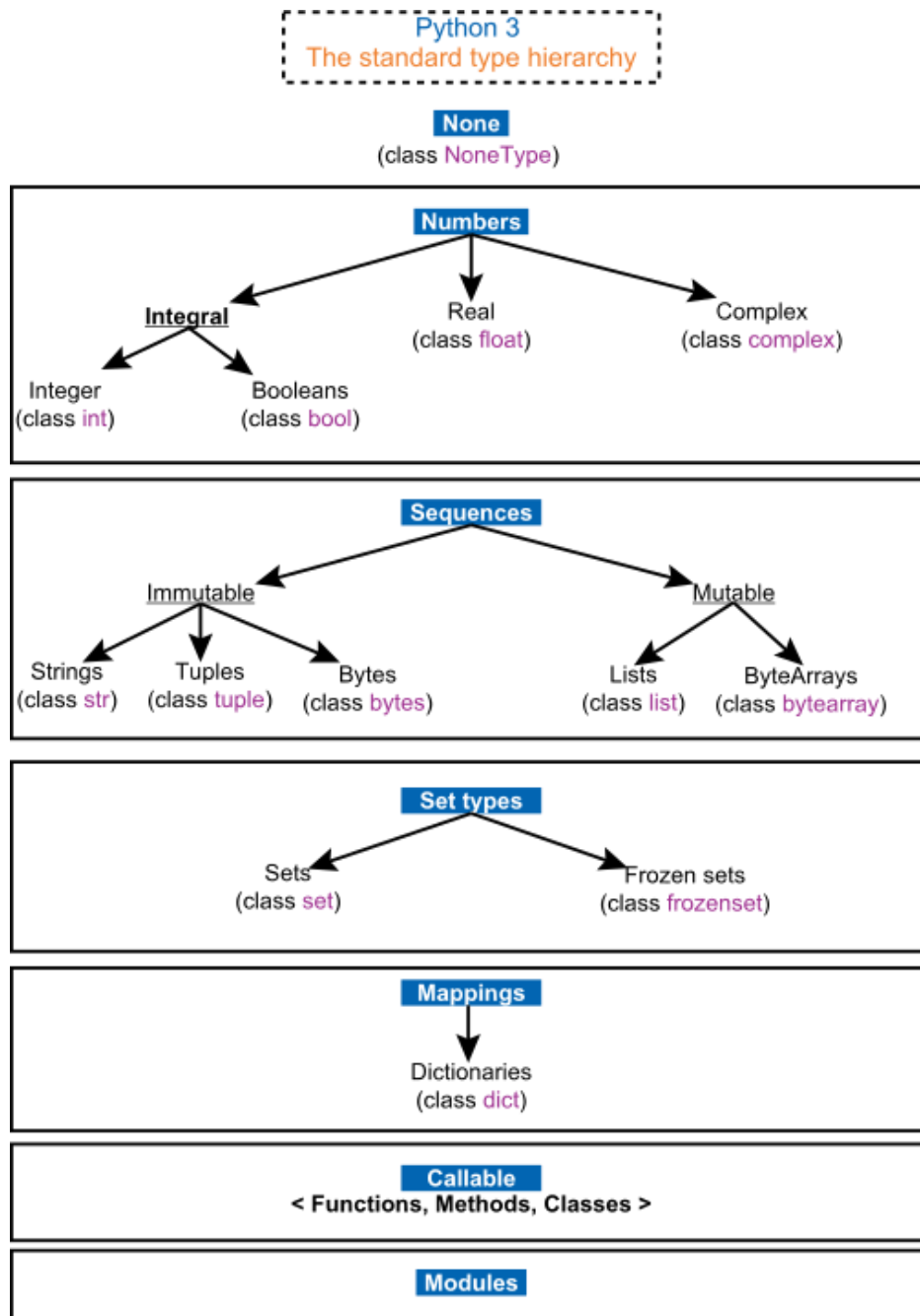
Методи об'єктів — це функції, приєднані до класу об'єкта; синтаксис `instance.method(аргумент)` є, для звичайних методів і функцій, синтаксичним цукром для `Class.method(instance, argument)`. Методи Python мають явний параметр `self` для доступу до даних екземпляра, на відміну від неявного `self` (або `this`) у деяких інших об'єктно-орієнтованих мовах програмування (наприклад, C++, Java, Objective-C, Ruby). Python також надає методи, які часто називають методами `dunder` (через те, що їхні назви починаються та закінчуються подвійним підкресленням), щоб дозволити визначеним користувачем класам змінювати те, як вони обробляються рідними операціями, включаючи довжину, порівняння, арифметичні операції та перетворення типів.

Python використовує качиний тип і має типізовані об'єкти, але нетипові імена змінних. Обмеження типу не перевіряються під час компіляції; скоріше, операції з об'єктом можуть завершитися помилкою, що означає, що він невідповідного типу. Незважаючи на динамічну типізацію, Python строго типізований, забороняючи операції, які не є чітко визначеними (наприклад, додавання числа до рядка), а не мовчки намагаючись зрозуміти їх сенс.

Python дозволяє програмістам визначати власні типи за допомогою класів, які найчастіше використовуються для об'єктно-орієнтованого програмування. Нові екземпляри класів створюються шляхом виклику класу (наприклад, `SpamClass()` або `EggsClass()`), і класи є екземплярами типу метакласу (сам по собі є екземпляром), що дозволяє метапрограмування та рефлексію.

До версії 3.0 у Python було два типи класів (обидва використовували однаковий синтаксис): старий та новий стиль, поточні версії Python підтримують лише новий стиль семантики.

Python підтримує поступове введення. Синтаксис Python дозволяє вказувати статичні типи, але вони не перевіряються в реалізації за замовчуванням, CPython. Експериментальна додаткова статична перевірка типів, туру, підтримує перевірку типів під час компіляції.





У Python є звичайні символи для арифметичних операторів (+, -, \*, /), оператора поверхового ділення // та операції за модулем % (де залишок може бути від'ємним, наприклад,  $4 \% -3 == -2$ ). Він також має \*\* для піднесення до степеня, напр.  $5^{**}3 == 125$  і  $9^{**}0,5 == 3,0$ , а також оператор множення матриці @. Ці оператори працюють як у традиційній математиці; з однаковими правилами пріоритету, оператори infix (+ і - також можуть бути унарними для представлення додатних і від'ємних чисел відповідно).

Ділення між цілими числами дає результати з плаваючою комою. Поведінка ділення значно змінилася з часом:

Поточний Python (тобто з версії 3.0) змінено / на те, щоб він завжди використовував ділення з плаваючою комою, наприклад,  $5/2 == 2,5$ .

Запроваджено поверховий поділ // оператор. Отже,  $7//3 == 2$ ,  $-7//3 == -3$ ,  $7,5//3 == 2,0$  і  $-7,5//3 == -3,0$ . Додавання з \_\_future\_\_ поділу імпорту призводить до того, що модуль, який використовується в Python 2.7, використовує правила Python 3.0 для поділу (див. вище).

У термінах Python / — це справжній поділ (або просто поділ), а // — це поділ на поверх. / до версії 3.0 є класичним поділом.

Округлення до негативної нескінченності, хоч і відрізняється від більшості мов, додає послідовності. Наприклад, це означає, що рівняння  $(a + b)//b == a//b + 1$  завжди вірне. Це також означає, що рівняння  $b*(a//b) + a\%b == a$  дійсно як для додатних, так і для від'ємних значень a. Однак підтримання дійсності цього рівняння означає, що хоча результат  $a\%b$ , як і очікувалося, знаходиться в напіввідкритому інтервалі  $[0, b)$ , де b є додатним цілим числом, він повинен лежати в інтервалі  $(b, 0]$ , коли b від'ємне.

Python надає функцію округлення для округлення числа з плаваючою точкою до найближчого цілого числа. Для встановлення зв'язків Python 3 використовує округлення до навіть: `round(1.5)` і `round(2.5)` обидва створюють 2. Версії до 3 використовували округлення від нуля: `round(0.5)` дорівнює 1.0, `round(-0.5)` є -1.0.

Python допускає логічні вирази з декількома відношеннями рівності у спосіб, який узгоджується із загальним використанням у математиці. Наприклад, вираз  $a < b < c$  перевіряє, чи  $a$  менше  $b$  і  $b$  менше  $c$ . Мови, похідні від C, інтерпретують цей вираз інакше: у C вираз спочатку обчислює  $a < b$ , що призводить до 0 або 1, а потім цей результат порівнюється з  $c$ .

Python використовує арифметику довільної точності для всіх цілочисельних операцій. Тип/клас `Decimal` у модулі `decimal` надає десяткові числа з плаваючою комою з попередньо визначеною довільною точністю та кількома режимами округлення. Клас `Fraction` у модулі `fractions` забезпечує довільну точність для раціональних чисел.

Завдяки великій математичній бібліотеці Python і сторонній бібліотеці `NumPy`, яка ще більше розширює рідні можливості, її часто використовують як наукову мову сценаріїв для вирішення таких проблем, як обробка чисельних даних і маніпуляції.

Для покращеного функціоналу у Python використовуються бібліотеки. Велика стандартна бібліотека Python надає інструменти, які підходять для багатьох завдань, і зазвичай називають її однією з найбільших сильних сторін. Для додатків, що працюють в Інтернеті, підтримується багато стандартних форматів і протоколів, таких як MIME і HTTP. Він містить модулі для створення графічних інтерфейсів користувача, підключення до реляційних баз даних, генерації псевдовипадкових чисел, арифметики з десятковими дробами довільної точності, маніпулювання регулярними виразами та модульного тестування.

Деякі частини стандартної бібліотеки охоплюються специфікаціями, наприклад, реалізація інтерфейсу шлюзу веб-сервера (WSGI) `wsgiref` відповідає PEP 333, але більшість визначено їх кодом, внутрішньою документацією та наборами тестів. Однак, оскільки більша частина стандартної бібліотеки є крос-платформним кодом Python, лише деякі модулі потребують змін або переписування для варіантних реалізацій.

Станом на 14 листопада 2022 року індекс пакетів Python (PyPI), офіційне сховище стороннього програмного забезпечення Python, містить понад 415 000 пакетів із широким спектром функціональних можливостей.

### 2.3. Використання бібліотеки мови програмування

Для створення потрібної програми використовуватимуться бібліотеки Tkinter, Pandas та NumPy.

Tkinter — це прив'язка Python до інструментарію Tk GUI. Це стандартний інтерфейс Python для набору інструментів графічного інтерфейсу користувача Tk і фактично стандартний графічний інтерфейс Python. Tkinter входить до стандартної інсталяції Python для Linux, Microsoft Windows і macOS.

Назва Tkinter походить від інтерфейсу Tk. Tkinter був написаний Стіном Лумхольтом і Гвідо ван Россумом [3], потім перероблений Фредріком Лундом.

Tkinter — це безкоштовне програмне забезпечення, випущене за ліцензією Python.

Як і більшість інших сучасних прив'язок Tk, Tkinter реалізовано як оболонку Python навколо повного інтерпретатора Tcl, вбудованого в інтерпретатор Python. Виклики Tkinter транслюються в команди Tcl, які передаються цьому вбудованому інтерпретатору, що дає змогу змішувати Python і Tcl в одній програмі.

Існує кілька популярних альтернативних бібліотек GUI, таких як Kivy, Pygame, Pyglet, PyGObject, PyQt, PySide та wxPython.

Деякі визначення:

Вікно — цей термін має різні значення в різних контекстах, але загалом він відноситься до прямокутної області десь на екрані користувача.

Вікно верхнього рівня — Вікно, яке діє як дочірнє вікно основного. Його прикрасять стандартна рамка і елементи управління для диспетчера робочого столу. Його можна переміщувати по робочому столу та зазвичай змінювати розмір.

Віджет — Загальний термін для будь-яких будівельних блоків, які складають програму в графічному інтерфейсі користувача.

- Основні віджети: контейнери: фрейм, `labelframe`, верхній рівень, панельне вікно. Кнопки: `button`, `radiobutton`, `checkboxbutton` (чекбокс) і `menubutton`. Текстові віджети: мітка, повідомлення, текст. Віджети запису: масштаб, смуга прокручування, список, повзунок, поле обертання, запис (однорядковий), меню параметрів, текст (багаторядковий) і полотно (векторна та піксельна графіка).
- Tkinter надає три модулі, які дозволяють відображати спливаючі діалогові вікна: `tk.messagebox` (діалогове вікно підтвердження, інформації, попередження та помилки), `tk.filedialog` (діалогове вікно вибору одного файлу, кількох файлів і каталогу) і `tk.colorchooser` (вибір кольору) .
- Python 2.7 і Python 3.1 включають «тематичний Tk» («`ttk`») функціональність Tk 8.5. Це дозволяє легко тематизувати віджети Tk, щоб вони виглядали як рідне середовище робочого столу, в якому працює програма, таким чином усуваючи давню критику Tk (і, отже, Tkinter). Деякі віджети є ексклюзивними для `ttk`, наприклад поле зі списком, панель прогресу, перегляд дерева, блокнот, роздільник і розмір.

Рамка (Frame) — У Tkinter віджет `Frame` є основною одиницею організації для складних макетів. Фрейм — це прямокутна область, яка може містити інші віджети.

Child and Parent — Коли створюється будь-який віджет, створюється зв'язок «батько–дочірній». Наприклад, якщо ви розмістите текстову мітку всередині рамки, рамка буде батьківською міткою.

Pandas — це бібліотека програмного забезпечення, написана на мові програмування Python для обробки та аналізу даних. Зокрема, він пропонує структури даних і операції для роботи з числовими таблицями та часовими рядами. Це безкоштовне програмне забезпечення, випущене за ліцензією BSD із трьох пунктів. Назва походить від терміну «панельні дані», економетричного терміну для наборів даних, які включають спостереження за кілька періодів

часу для тих самих осіб. Його назва є грою самої фрази «аналіз даних Python». Вес МакКінні почав будувати те, що стане пандами, в AQR Capital, коли він був там дослідником з 2007 по 2010 рік.

Pandas в основному використовується для аналізу даних і відповідної обробки табличних даних у DataFrames. Pandas дозволяє імпортувати дані з різних форматів файлів, таких як значення, розділені комами, JSON, Parquet, таблиці або запити бази даних SQL і Microsoft Excel. Pandas дозволяє виконувати різноманітні операції маніпулювання даними, такі як об'єднання, зміна форми, вибір, а також функції очищення та суперечки даних. Розробка pandas ввела в Python багато порівнянних функцій роботи з DataFrames, які були встановлені в мові програмування R. Бібліотека pandas побудована на основі іншої бібліотеки NumPy, яка орієнтована на ефективну роботу з масивами замість функцій роботи з DataFrames.

Розробник Вес МакКінні почав працювати над pandas у 2008 році під час роботи в AQR Capital Management через потребу у високопродуктивному, гнучкому інструменті для кількісного аналізу фінансових даних. Перш ніж залишити AQR, він зміг переконати керівництво дозволити йому відкрити бібліотеку.

Інший співробітник AQR, Чан Ше, приєднався до зусиль у 2012 році як другий великий внесок у бібліотеку.

NumPy (вимовляється /'nʌmpaɪ/ NUM-py) — це бібліотека для мови програмування Python, яка додає підтримку великих багатовимірних масивів і матриць разом із великою колекцією математичних функцій високого рівня для роботи з цими масивами. Попередник NumPy, Numeric, спочатку був створений Джимом Хугунінім за участю кількох інших розробників. У 2005 році Тревіс Оліфант створив NumPy, включивши функції конкуруючого Numarray у Numeric із великими змінами. NumPy є програмним забезпеченням з відкритим вихідним кодом і має багато учасників. NumPy — це проект, який фінансується NumFOCUS.

Мова програмування Python спочатку не була розроблена для чисельних обчислень, але рано привернула увагу наукової та інженерної спільноти. У 1995 році була заснована спеціальна група інтересів (SIG) `matrix-sig` з метою визначення пакету обчислень масивів; серед його членів був розробник і супроводжувач Python Гвідо ван Россум, який розширив синтаксис Python (зокрема синтаксис індексування), щоб полегшити обчислення масивів.

У 2015 році панди підписали проект NumFOCUS, некомерційної благодійної організації 501(c)(3) США, яка фінансується фінансово.

Реалізація матричного пакету була завершена Джимом Фултоном, потім узагальнена Джимом Хугунінім і названа Numeric (також відома як «числові розширення Python» або «NumPy»), з впливом сімейства мов APL, Basis, MATLAB, FORTRAN, S і S+ та ін. Гугунін, аспірант Массачусетського технологічного інституту (MIT):<sup>10</sup> приєднався до Корпорації національних дослідницьких ініціатив (CNRI) у 1997 році, щоб працювати над JPython, залишивши Пола Дюбуа з Ліверморської національної лабораторії імені Лоуренса (LLNL) на посаду супроводжувача: <sup>10</sup> Інші перші автори: Девід Ашер, Конрад Хінсен і Тревіс Оліфант.

Новий пакет під назвою Numarray був написаний як більш гнучка заміна Numeric. Як і Numeric, він також не підтримується. Numarray мав швидші операції для великих масивів, але повільніше, ніж Numeric для малих, тому певний час обидва пакети використовувалися паралельно для різних випадків використання. Остання версія Numeric (v24.2) була випущена 11 листопада 2005 року, а остання версія numarray (v1.5.2) була випущена 24 серпня 2006 року.

Було бажання включити Numeric у стандартну бібліотеку Python, але Гвідо ван Россум вирішив, що тоді код не можна підтримувати в такому стані.

На початку 2005 року розробник NumPy Тревіс Оліфант хотів об'єднати спільноту навколо єдиного пакету масивів і переніс функції Numarray на Numeric, випустивши результат як NumPy 1.0 у 2006 році. Цей новий проект

був частиною SciPy. Щоб уникнути встановлення великого пакета SciPy лише для отримання об'єкта масиву, цей новий пакет було відокремлено та названо NumPy. Підтримка Python 3 була додана в 2011 році з NumPy версії 1.5.0.

У 2011 році PyPy розпочав розробку реалізації NumPy API для PyPy. Він ще не повністю сумісний з NumPy.

NumPy націлений на еталонну реалізацію Python CPython, яка є неоптимізованим інтерпретатором байт-коду. Математичні алгоритми, написані для цієї версії Python, часто працюють набагато повільніше, ніж скомпільовані еквіваленти через відсутність оптимізації компілятора. NumPy частково вирішує проблему повільності, надаючи багатовимірні масиви, функції та оператори, які ефективно працюють з масивами; їх використання вимагає переписування коду, переважно внутрішніх циклів, за допомогою NumPy.

Використання NumPy у Python надає функціональність, порівнянну з MATLAB, оскільки вони обидва інтерпретуються, і вони обидва дозволяють користувачеві писати швидкі програми, якщо більшість операцій працюють із масивами чи матрицями замість скалярів. Для порівняння, MATLAB може похвалитися великою кількістю додаткових наборів інструментів, зокрема Simulink, тоді як NumPy внутрішньо інтегрований з Python, більш сучасною та повною мовою програмування. Крім того, доступні додаткові пакети Python; SciPy — це бібліотека, яка додає більше функціональних можливостей, подібних до MATLAB, а Matplotlib — це пакет для побудови графічних зображень, який забезпечує функціональні можливості побудови, подібні до MATLAB. Хоча matlab може виконувати операції з розрідженою матрицею, numpy сам по собі не може виконувати такі операції та вимагає використання бібліотеки scipy.sparse. Внутрішньо і MATLAB, і NumPy покладаються на BLAS і LAPACK для ефективних обчислень лінійної алгебри.

Прив'язки Python широко використовуваної бібліотеки комп'ютерного зору OpenCV використовують масиви NumPy для зберігання та роботи з даними. Оскільки зображення з декількома каналами просто представлені як

тривимірні масиви, індексація, нарізка або маскування за допомогою інших масивів є дуже ефективними способами доступу до конкретних пікселів зображення. Масив NumPy як універсальна структура даних у OpenCV для зображень, вилучених точок функцій, ядер фільтрів та багато іншого значно спрощує робочий процес програмування та налагодження.

Основною функціональністю NumPy є його структура даних "ndarray" для n-вимірного масиву. Ці масиви являють собою побіжний погляд на пам'ять. На відміну від вбудованої структури даних списку Python, ці масиви однорідно типізовані: усі елементи одного масиву мають бути одного типу.

Такі масиви також можуть бути переглядами буферів пам'яті, виділених розширеннями C/C++, Python і Fortran для інтерпретатора CPython без необхідності копіювати дані, що забезпечує певну сумісність із існуючими числовими бібліотеками. Ця функціональність використовується пакетом SciPy, який огортає низку таких бібліотек (зокрема BLAS і LAPACK). NumPy має вбудовану підтримку відображених у пам'яті ndarrays.

Вставлення або додавання записів до масиву не так тривіально можливо, як зі списками Python. Підпрограма `np.pad(...)` для розширення масивів фактично створює нові масиви бажаної форми та значень заповнення, копіює заданий масив у новий і повертає його. Операція NumPy `np.concatenate([a1,a2])` насправді не пов'язує два масиви, а повертає новий, заповнений записами з обох заданих масивів послідовно. Зміна розмірності масиву за допомогою `np.reshape(...)` можлива лише до тих пір, поки кількість елементів у масиві не змінюється. Ці обставини походять із того факту, що масиви NumPy мають бути переглядами суміжних буферів пам'яті. Пакет заміни під назвою Blaze намагається подолати це обмеження.

Алгоритми, які не можна виразити як векторизовану операцію, як правило, працюватимуть повільно, тому що вони повинні бути реалізовані на «чистому Python», у той час як векторизація може збільшити складність пам'яті деяких операцій з константною на лінійну, тому що тимчасові масиви повинні



бути створені такими ж великими, як і входи. Компіляція числового коду під час виконання була реалізована кількома групами, щоб уникнути цих проблем; Рішення з відкритим кодом, які взаємодіють із NumPy, включають numexpr і Numba. Cython і Pythran є альтернативою для статичної компіляції.

Багато сучасних великомасштабних наукових обчислювальних програм мають вимоги, які перевищують можливості масивів NumPy. Наприклад, масиви NumPy зазвичай завантажуються в пам'ять комп'ютера, яка може мати недостатню ємність для аналізу великих наборів даних. Крім того, операції NumPy виконуються на одному ЦП. Однак багато операцій лінійної алгебри можна прискорити, виконуючи їх на кластерах центральних процесорів або спеціалізованого обладнання, наприклад GPU та TPU, на яких покладаються багато програм глибокого навчання. У результаті за останні роки в науковій екосистемі Python з'явилося кілька альтернативних реалізацій масивів, наприклад Dask для розподілених масивів і TensorFlow або JAX для обчислень на GPU. Через свою популярність вони часто реалізують підмножину API NumPy або імітують його, щоб користувачі могли змінювати реалізацію свого масиву з мінімальними змінами коду. Нещодавно представлена бібліотека під назвою CuPy, прискорена фреймворком CUDA від Nvidia, також показала потенціал для швидшого обчислення, будучи «заміною» NumPy.

#### 2.4. Алгоритм роботи програмного продукту

Бібліотеки:

- 1) `tkinter` - бібліотека для створення віконних програм. За допомогою неї додається інтерфейс та віджети для роботи з готовою базою даних.
- 2) `pandas` - бібліотека для роботи з файлами, переклад файлу типу ексель до даних програмування, що читаються мовою, і навпаки.
- 3) `numpy` - бібліотека що дозволяє над матрицями алгебраїчної операції.

Глобальні змінні:

1) `root` - змінна класу `Tk ()`. Створює вікно програми для зберігання віджетів та інтерфейсу.

2) `color` - змінна типу `string`. Визначає колір заднього фону, написів, кнопок тощо.

3) `position` - змінна типу `dictionary`. Задає координати віджетів у вікні програми.

4) `label` - змінна класу `Label ()`. За своєю суттю є написом, який з'являтиметься і змінюватиме своє значення залежно від ходу роботи програми.

5) `text_editor` - об'єкт класу `ScrolledText()`. Текстове поле для відображення результату обчислень, його перевірки надалі збереження у вигляді окремого файлу.

Функції:

1) `save()` - функція без аргументів, потрібна для збереження результату обчислення програми в окремий файл.

2) `open_file()` - функція без аргументів, основна обчислювальна функція програми. Обчислює матрицю  $K_{fy}$ , виводить її на екран програми та створює додаткові віджети для роботи з результатом.

### 3. РЕЗУЛЬТАТИ АНАЛІЗУ ДАННИХ

Нижче наведені результати статистичного факторного аналізу, проведеного відповідно до викладеного вище методу (не програмою).

Матриця наявності/відсутності кореляції між факторами и логарифмами параметрів розглянутих для балкерів показана на рисунку 2.

Incid	$\ln\left(\frac{L}{B}\right)$	$\ln\left(\frac{B}{H}\right)$	$\ln\left(\frac{D_b}{\rho LBT}\right)$	$\ln\left(\frac{D_b}{\rho LBH}\right)$	$\ln\left(\frac{D_b-D_n}{D_b}\right)$	$\ln\left(\frac{V}{\sqrt{g^3 \frac{D_b}{\rho}}}\right)$	$\ln\left(\frac{P}{\rho g D_b V}\right)$
F1	0	0	0	0	0	0	1
F2	0	0	0	0	0	0	1
F3	0	0	0	0	0	0	1
F4	1	0	1	0	1	1	1
F5	0	0	0	1	1	1	0
F6	0	1	0	1	1	1	0
F7	1	1	1	1	1	1	1

Рис. 2. Матриця інцидентності для балкерів

Загальний вид факторів прийнятих до розгляду для балкерів описує система (11)

$$\begin{aligned}
 F1 &= \ln\left(\left(\frac{P}{\rho g D_b V}\right)^{-0.83}\right); F2 = \ln\left(\left(\frac{P}{\rho g D_b V}\right)^{-0.91}\right); F3 = \ln\left(\left(\frac{P}{\rho g D_b V}\right)^{-0.24}\right); \\
 F4 &= \ln\left(\left(\frac{L}{B}\right)^{0.42} \left(\frac{D_b}{\rho LBT}\right)^{-0.11} \left(\frac{D_b-D_n}{D_b}\right)^{-0.52} \left(\frac{V}{\sqrt{g^3 \frac{D_b}{\rho}}}\right)^{-0.5} \left(\frac{P}{\rho g D_b V}\right)^{0.22}\right); \\
 F5 &= \ln\left(\left(\frac{D_b}{\rho LBH}\right)^{0.2} \left(\frac{D_b-D_n}{D_b}\right)^{0.22} \left(\frac{V}{\sqrt{g^3 \frac{D_b}{\rho}}}\right)^{-0.25}\right); \\
 F6 &= \ln\left(\left(\frac{B}{H}\right)^{-0.2} \left(\frac{D_b}{\rho LBT}\right)^{-0.07} \left(\frac{D_b-D_n}{D_b}\right)^{0.04} \left(\frac{V}{\sqrt{g^3 \frac{D_b}{\rho}}}\right)^{-0.25}\right);
 \end{aligned}$$

$$F7 = \ln \left( \left( \frac{L}{B} \right)^{-0.07} \left( \frac{B}{H} \right)^{-0.1} \left( \frac{D_b}{\rho L B T} \right)^{0.09} \left( \frac{D_b}{\rho L B H} \right)^{0.21} \left( \frac{D_b - D_n}{D_b} \right)^{-0.07} \left( \frac{V}{\sqrt{g^3 \sqrt{\frac{D_b}{\rho}}}} \right)^{0.007} \left( \frac{P}{\rho g D_b V} \right)^{-0.005} \right);$$

(11)

Представлена на рисунку 2 матриця інцидентності відображає значимість кореляцій по t-критерію Стюдента між параметрами  $Y_i$  балкерів вибірки і факторами при рівні значущості 0,05 і 32 ступенях свободи. З цієї таблиці видно, що фактори  $F_1, F_2, F_3$  залежать тільки від енергетичного параметра  $\frac{P}{\rho g D_b V}$ ; фактор  $F_5$  залежить від характеристик вантажопідйомності

$\frac{D_b}{\rho L B H}, \frac{D_b - D_n}{D_b}$  і швидкісного параметра  $\sqrt{g^3 \sqrt{\frac{D_b}{\rho}}}$ ; фактори  $F_4$  і  $F_5$  залежать від

геометричних параметрів, вантажопідйомності, швидкісного і енергетичного параметра; фактор  $F_7$  залежить від всіх розглянутих параметрів. Угрупування по значення факторів  $F_1, F_2, F_3$ , як окремо, так і в комплексі дає розподіл суден лише за одним енергетичним параметром.

Таблиця 4. Групування балкерів за значеннями факторів

Інтервал значень фактора F1	Кількість суден в групі	Номери суден групи в вибірці
5.78 < F1 < 6.60	29	1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,18,19,21,22,23,24,25,26,27,28,29,30,32
6.60 < F1 < 7.43	2	20,31
7.43 < F1 < 8.25	0	
8.25 < F1 < 9.07	1	14

Продовження Таблиці 4

Інтервал значень фактора F2	Кількість суден в групі	Номери суден групи в вибірці
5.30<F2<6.21	26	1,2,3,4,5,6,7,8,9,10,12,13,16,17,18,19,21,22, , 23,24,25,26,27,28,29,30
6.21<F2<7.13	5	11,15,20,31,32
7.13<F2<8.04	0	
8.04<F2<8.96	1	14
Інтервал значень фактора F3	Кількість суден в групі	Номери суден групи в вибірці
1.63<F3<1.88	26	1,2,3,4,5,6,7,8,10,11,12,13,15,16,17,18,19,2 1,22,23,25,26,28,29,30,32
1.88<F3<2.15	5	9,20,24,27,31
2.12<F3<2.37	0	
2.37<F3<2.62	1	14

Продовження Таблиці 4

Інтервал значень фактора F1,F2,F3	Кількість суден в групі	Номери суден групи в вибірці
5.78<F1<6.60; 5.30<F2<6.21; 1.63<F3<1.88	23	1,2,3,4,5,6,7,8,10,12,13,16,17,18,19,21,22,2 3,25,26,28,29,30
5.78<F1<6.60; 5.30<F2<6.21; 1.88<F3<2.12	3	9,24,27
5.78<F1<6.60; 6.21<F2<7.13; 1.63<F3<1.88	3	11,15,23
6.60<F1<7.43; 6.21<F2<7.13; 1.88<F3<2.12	2	20,31
8.25<F1<9.07;	1	14

8.04<F2<8.96; 2.37<F3<2.62		
Інтервал значень фактора F1,F2,F3,F5	Кількість суден в групі	Номери суден групи в вибірці
5.78<F1<6.60; 5.30<F2<6.21; 1.63<F3<1.88; 2.14<F5<2.38	23	1,2,3,4,5,6,7,8,10,12,13,16,17,18,19,21,22,23,25,26,28,29,30
5.78<F1<6.60; 5.30<F2<6.21; 1.88<F3<2.12; 2.14<F5<2.38	3	9,24,27
5.78<F1<6.60; 6.21<F2<7.13; 1.63<F3<1.88; 2.14<F5<2.38	1	11
5.78<F1<6.60; 6.21<F2<7.13; 1.63<F3<1.88; 2.38<F5<2.62	2	15,23
6.60<F1<7.43; 6.21<F2<7.13; 1.88<F3<2.12; 2.38<F5<2.62	2	20,31
8.25<F1<9.07; 8.04<F2<8.96; 2.37<F3<2.62; 2.86<F5<3.10	1	14

З 32 суден в одну групу попадає більшість, причому групи практично збігаються, незалежно від обраного фактора. Тобто ці судна мають близьку відносну енергетичну характеристику. Так само видно, що фактор не впливає на угруповання, так як угруповання по  $F_1, F_2, F_3$  і по  $F_1, F_2, F_3, F_5$  збігаються.

Угрупування по  $F_4$  і  $F_6$  дає більш детальну структуру розподілу суден у вибірці, вже враховані всі параметри.

Таблиця 5. Групування балкерів за значеннями факторів.  $F_4$ ,  $F_6$

Інтервал значень фактора $F_4$ и $F_6$	Кількість суден в групі	Номери суден групи в вибірці
-1.07< $F_4$ <- 0.84; 0.97< $F_6$ <1.08	1	14
-0.60< $F_4$ <- 0.37; 0.64< $F_6$ <0.75	7	4,6,8,12,13,21,25
-0.60< $F_4$ <- 0.37; 0.75< $F_6$ <0.86	5	11,15,20,31,32
-0.37< $F_4$ <- 0.13; 0.64< $F_6$ <0.75	16	1,2,3,5,7,10,16,17,18,19,22,23,26,28,29,30
-0.37< $F_4$ <- 0.13; 0.75< $F_6$ <0.86	3	9,24,27

Аналіз даних таблиці 5 показує, що наприклад найчисельніша група з 16 суден представлена суднами різних дедвейтів в основному HANDYSIZE та PANAMAX. Крім того два судна групи спроектовані фірмою IMABARI SHIPBUILDING, інші проекти – корейських та по більшій частині китайських проектувальників. Брак інформації про вантажомісткість суден та неможливість використовувати як параметр питомий навантажувальний обсяг (ПНО) призвів до того що в аналізі замість ПНО застосований параметр  $c = dw/(LBH)$  т/м<sup>3</sup>. За цим параметром судна групи мають найбільші з розглянутих значення  $1.78 \leq c \leq 2.25$ , при цьому 7 суден даної групи мають значення більш ніж  $c = 2.0$ , що свідчить про перевезення вантажу з високими значеннями ПНО.

Група з 5 суден з номерами 11,15,20,31,32 представляє собою групу балкерів типу CAPESIZE, за виключенням номера 15 (тип VLOC). Проекти суден даної групи створені компаніями DAMEN, HYUNDAI, NIPPON KOKAN KOJI та ZHOUSHAN JINHAIWAN SHIPYARD яка спеціалізується на

будівництві балкерів. Тобто є певні гарантії, що орієнтування на характеристики суден цієї групи в процесі проектування дозволить отримати судно з характеристиками які в комплексі не будуть гіршими характеристик суден даної групи (у випадку якщо вибірка досить чисельна та актуальна - не будуть гіршими середньосвітових характеристик).

Угрупування по фактору .

Таблиця 6. Групування балкерів за значеннями фактора F7

Інтервал значень фактора F7	Кількість суден в групі	Номери суден групи в вибірці
$-0.27 < F7 < -0.26$	6	1,2,11,15,22,31
$-0.26 < F7 < -0.24$	12	3,5,7,12,13,17,20,21,25,26,27,28
$-0.24 < F7 < -0.23$	12	4,8,9,10,16,18,19,23,24,29,30,32
$-0.23 < F7 < -0.22$	2	6,14

3.1. Алгоритм застосування методу факторного аналізу для вибору характеристик судна

Алгоритм застосування методу факторного аналізу для вибору характеристик судна на початковому етапі проектування показаний на прикладі балкерів.

За однією з таблиць угруповань судів за значеннями факторів необхідно вибрати групу суден. Попереднє вивчення характеристик судів в групах має сприяти цьому вибору. Мова йде про аналіз дедвейту, ПНО, коефіцієнту утилізації по дедвейту та інших характеристик судна, які як маркер вкажуть на групу суден розглянуті фактори (або групи факторів) яких змінюються в вузьких межах.

Наприклад, по таблиці угруповання за значеннями факторів F4, F6 встановлена група судів одного типу, для якої значення факторів лежать межах ( $-0.6 < F4 < -0.37$ ;  $0.75 < F6 < 0.86$ ). Це судна: 11,15,20,31,32. Далі, по таблиці угруповання судів по фактору F7 необхідно вибрати групу, яка містить суду, що



потрапляють в обрану раніше групу. Ця група містить суду: 1,2,11,15,22,31 і визначається нерівністю  $-0.27 < F7 < -0.26$ . Для цієї групи судів, отриманих перетином двох раніше обраних груп, а саме для суден 11,15,31 формулюється система нерівностей щодо безрозмірних параметрів (12).

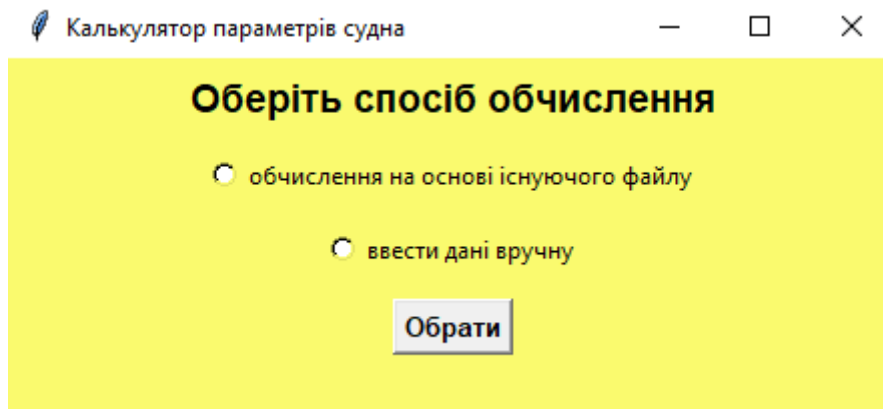
$$\begin{aligned}
 e^{-0.6} &\leq \left(\frac{L}{B}\right)^{0.41} \left(\frac{D_b}{\rho LBT}\right)^{-0.11} \left(\frac{D_b-D_n}{D_b}\right)^{-0.52} \left(\frac{V}{\sqrt{g^3 \sqrt{\frac{D_b}{\rho}}}}\right)^{-0.3} \left(\frac{P}{\rho g D_b V}\right)^{0.22} \leq e^{0.26}; \\
 e^{0.75} &\leq \left(\frac{B}{H}\right)^{-0.2} \left(\frac{D_b}{\rho LBT}\right)^{-0.07} \left(\frac{D_b-D_n}{D_b}\right)^{0.04} \left(\frac{V}{\sqrt{g^3 \sqrt{\frac{D_b}{\rho}}}}\right)^{-0.25} \leq e^{0.86}; \\
 e^{-0.27} &\leq \\
 \left(\frac{L}{B}\right)^{-0.07} \left(\frac{B}{H}\right)^{-0.1} \left(\frac{D_b}{\rho LBT}\right)^{0.09} \left(\frac{D_b}{\rho LBH}\right)^{0.21} \left(\frac{D_b-D_n}{D_b}\right)^{-0.07} \left(\frac{V}{\sqrt{g^3 \sqrt{\frac{D_b}{\rho}}}}\right)^{0.007} \left(\frac{P}{\rho g D_b V}\right)^{-0.008} &\leq \\
 e^{0.26} & \tag{12}
 \end{aligned}$$

Чотири з семи параметрів системи незалежні. Наприклад, незалежними можна вибрати геометричні характеристики  $\frac{L}{B}$ ,  $\frac{B}{H}$ ,  $\frac{D_b}{\rho LBT}$ ,  $\frac{D_b}{\rho LBH}$ . Тоді допустимі значення відносних потужності  $\frac{P}{\rho g D_b V}$ , швидкості  $\frac{V}{\sqrt{g^3 \sqrt{\frac{D_b}{\rho}}}}$  і характеристики вантажомісткості  $\frac{D_b-D_n}{D_b}$  визначаються з системи нерівностей. Якщо при цьому отримані інтервали значень суперечать здоровому глузду, то слід коригувати значення незалежних величин  $\frac{L}{B}$ ,  $\frac{B}{H}$ ,  $\frac{D_b}{\rho LBT}$ ,

$\frac{D_b}{\rho L V H}$ . При розумному їх виборі проектоване судно повинно потрапити в ту ж обрану групу, тобто буде судном того ж типу.

### 3.2. Тестування програмного продукту

При запуску програми з'являється інтерфейс із вибором подальших дій. Насамперед користувачеві пропонується вибрати спосіб обчислення даних: на основі даних із готового файлу, або на основі даних, введених вручну. За умовами завдання потрібно перевірити роботу програми для обчислення за готовими даними, тому обираємо перший варіант.

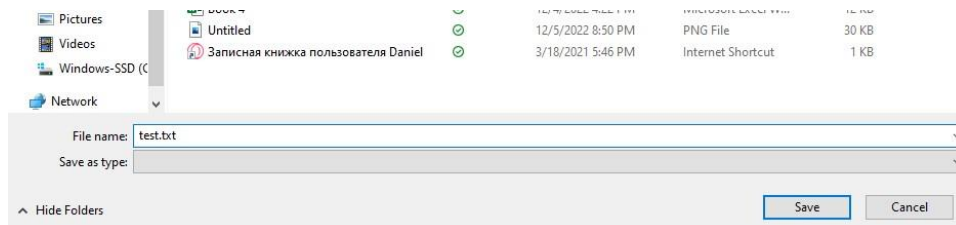
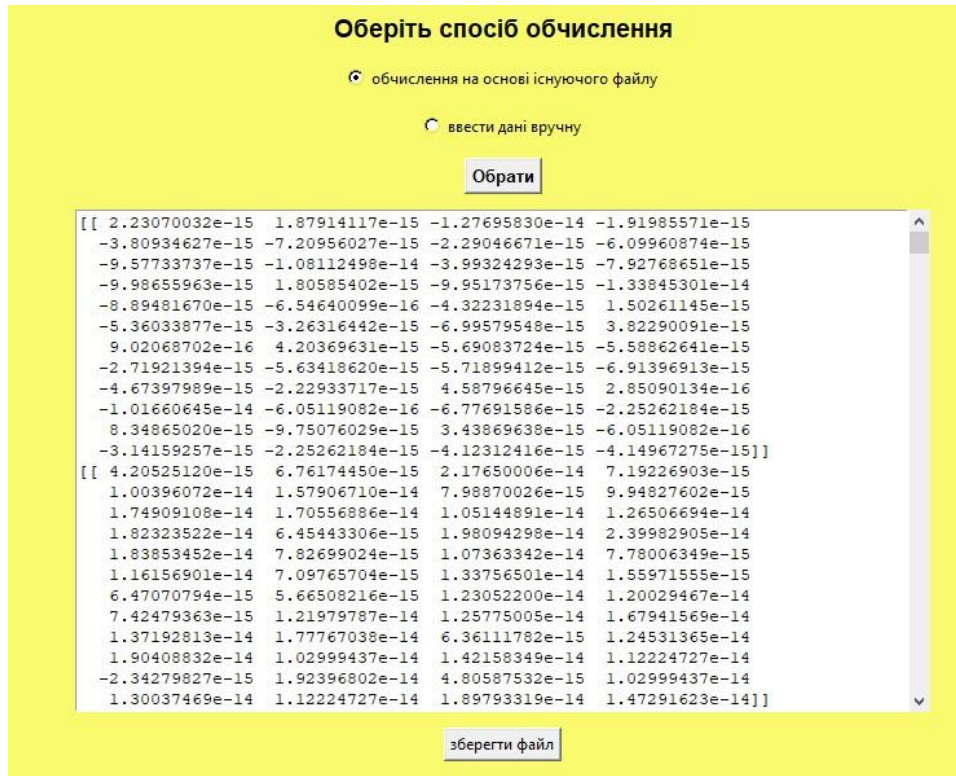


Натискаємо кнопку "обрати". Після натискання з'явиться вікно вибору файлу. Нас цікавить файл Book 4. Якщо вибрати файл, який не підходить за параметрами обчислювальної функції, вікно вибору файлу закриється і користувач повернеться до програми.



Після вибору необхідного файлу з'являється текстове поле з вичисленою матрицею  $K_f$  і можливістю повністю переглянути її, а також кнопка "зберегти файл". При натисканні на неї з'являється вікно вибору файлу, де ми задаємо ім'я

нового файлу та його розширення. Надалі цей файл можна використовуватиме подальшої побудови таблиць параметрів.



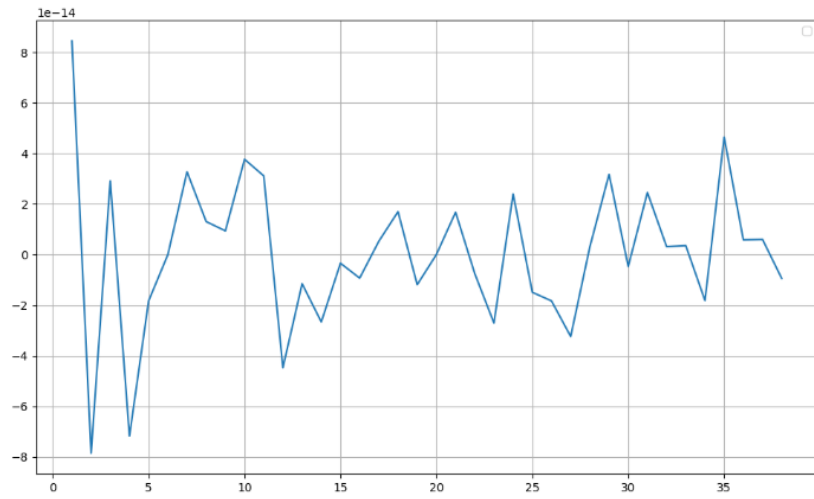
У нашому випадку ми назвали файл "test.txt".

```

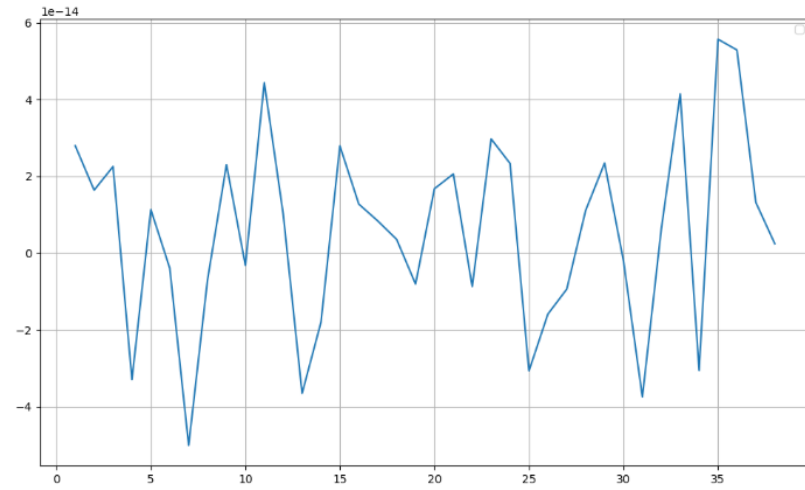
test - Notepad
File Edit Format View Help
[[ 2.23070032e-15 1.87914117e-15 -1.27695830e-14 -1.91985571e-15
-3.80934627e-15 -7.20956027e-15 -2.29046671e-15 -6.09960874e-15
-9.57733737e-15 -1.08112498e-14 -3.99324293e-15 -7.92768651e-15
-9.98655963e-15 1.80585402e-15 -9.95173756e-15 -1.33845301e-14
-8.89481670e-15 -6.54640099e-16 -4.32231894e-15 1.50261145e-15
-5.36033877e-15 -3.26316442e-15 -6.99579548e-15 3.82290091e-15
9.02068702e-16 4.20369631e-15 -5.69083724e-15 -5.58862641e-15
-2.71921394e-15 -5.63418620e-15 -5.71899412e-15 -6.91396913e-15
-4.67397989e-15 -2.22933717e-15 4.58796645e-15 2.85090134e-16
-1.01660645e-14 -6.05119082e-16 -6.77691586e-15 -2.25262184e-15
8.34865020e-15 -9.75076029e-15 3.43869638e-15 -6.05119082e-16
-3.14159257e-15 -2.25262184e-15 -4.12312416e-15 -4.14967275e-15]]
[[ 4.20525120e-15 6.76174450e-15 2.17650006e-14 7.19226903e-15
1.00396072e-14 1.57906710e-14 7.98870026e-15 9.94827602e-15
1.74909108e-14 1.70556886e-14 1.05144891e-14 1.26506694e-14
1.82323522e-14 6.45443306e-15 1.98094298e-14 2.39982905e-14
1.83853452e-14 7.82699024e-15 1.07363342e-14 7.78006349e-15
1.16156901e-14 7.09765704e-15 1.33756501e-14 1.55971555e-15
6.47070794e-15 5.66508216e-15 1.23052200e-14 1.20029467e-14
7.42479363e-15 1.21979787e-14 1.25775005e-14 1.67941569e-14
1.37192813e-14 1.77767038e-14 6.36111782e-15 1.24531365e-14
1.90408832e-14 1.02999437e-14 1.42158349e-14 1.12224727e-14
-2.34279827e-15 1.92396802e-14 4.80587532e-15 1.02999437e-14
1.30037469e-14 1.12224727e-14 1.89793319e-14 1.47291623e-14]]
[[ 2.36161349e-14 2.65342682e-14 2.67138068e-14 2.43327055e-14
2.53283292e-14 2.52038260e-14 2.39752782e-14 2.24007159e-14
2.41429108e-14 2.36768523e-14 2.55479882e-14 2.33414467e-14
2.45746242e-14 2.47970027e-14 2.33580315e-14 2.17794301e-14
2.37290793e-14 2.30087841e-14 2.33399027e-14 2.44857076e-14
2.36604453e-14 2.30132101e-14 2.36643793e-14 2.37209931e-14
2.45313874e-14 2.45073057e-14 2.33391819e-14 2.35343684e-14
2.31618398e-14 2.34518290e-14 2.46576299e-14 2.51073497e-14
2.46017106e-14 2.93893651e-14 2.76744559e-14 2.85389027e-14
2.42647551e-14 2.62471081e-14 2.00312161e-14 2.34181837e-14
2.36377585e-14 2.37236887e-14 2.46799522e-14 2.62471081e-14
2.13212059e-14 2.34181837e-14 2.74331349e-14 2.59660817e-14]]
Ln 1, Col 1 100% Windows (CRLF) UTF-8

```

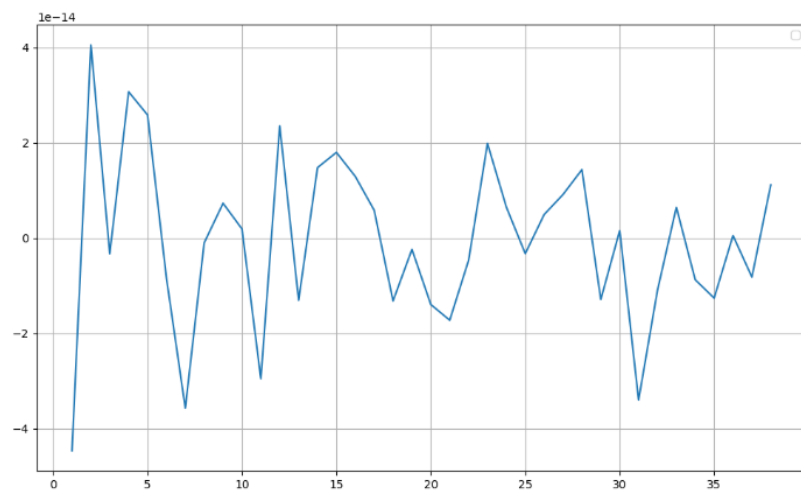
Нас також цікавлять проміжні обчислення, а саме матриця факторів. Її можна знайти наприкінці додатку. Побудувавши графік розбіжності значення факторів можна зрозуміти наскільки програмний продукт був точним у своїх обчисленнях і на які фактори варто спиратися під час планування виробництва лоцманських судів. Значення факторів у графіках обчислені для суден с першого по тридцять вісьмий номер.



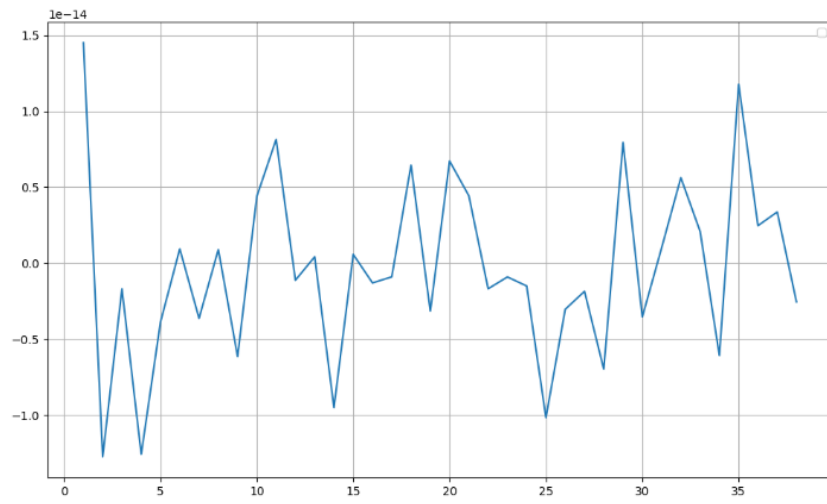
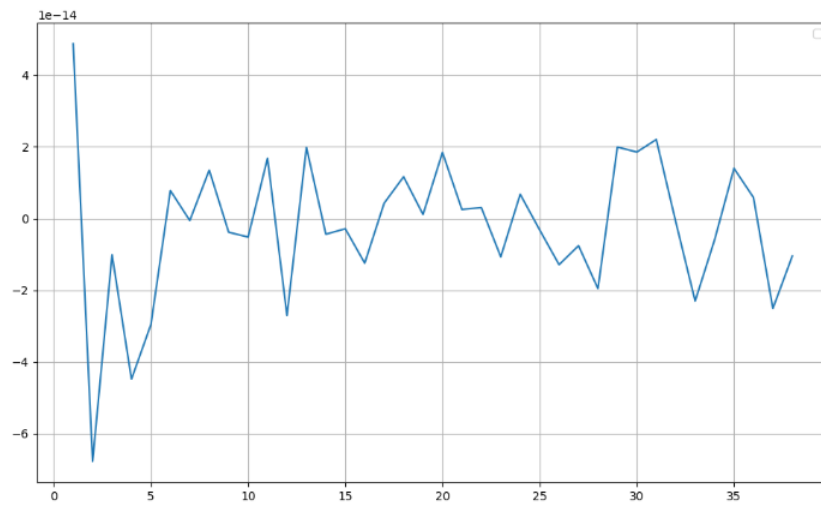
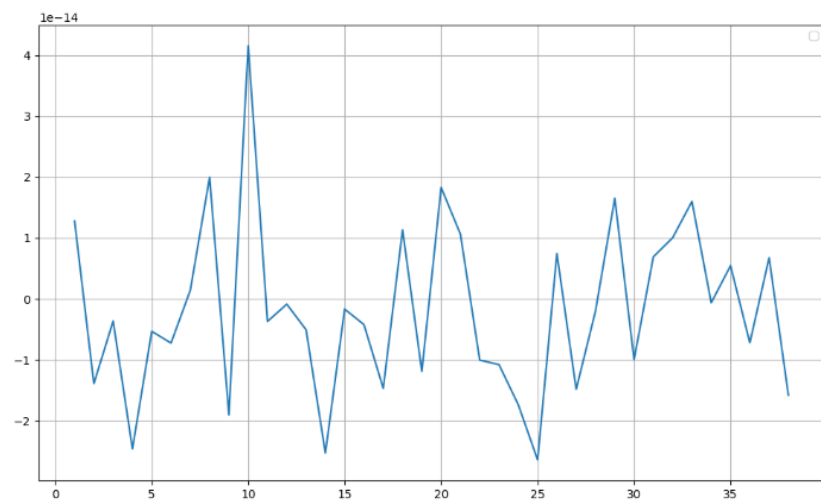
Графік значень фактора  $F1$



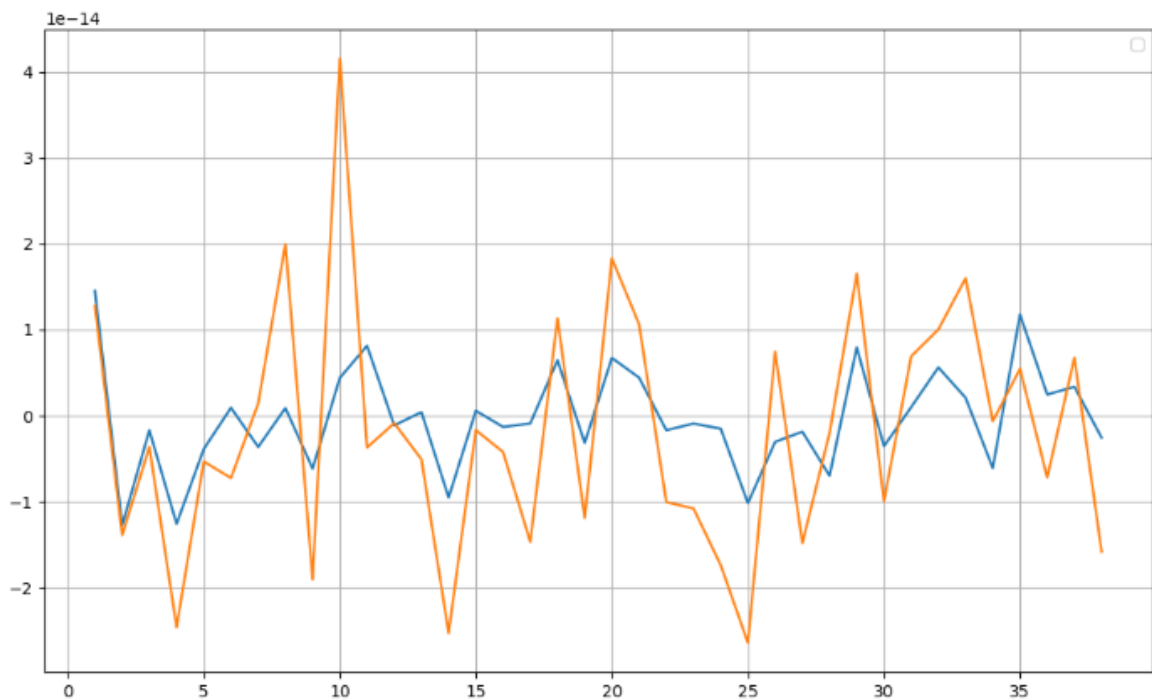
Графік значень фактора  $F2$



Графік значень фактора  $F3$

Графік значень фактора  $F_4$ Графік значень фактора  $F_5$ Графік значень фактора  $F_6$

Найменша розрахункова різниця виявилася у факторів F4 і F6. Відповідно, виходячи з наших розрахунків, виробництво балкерів потрібно проводити залежно від комбінації 4 і 6 факторів. Слід зауважити, що точність обчислення для лоцманських катерів набагато нижча, ніж за таких же розрахунків для суден-балкерів. Це сталося через те, що доступна в ході обчислень база даних балкерів набагато більша за кількість лоцманських катерів.



*Угрупування по F4 та F6*

## ВИСНОВКИ

Повернемося до питання, поставленого у початку статті, а саме про виявлення кількісних характеристик, які допоможуть обрати оптимальні параметри судна з численної вибірки прототипів. Використана математична теорія гарантує існування факторів, що визначаються параметрами судна, але не дає пояснень щодо їх інженерного сенсу. Відповідь на це питання потребує предметного дослідження, яке не можливе без висування деяких гіпотез. Аналіз тестових розрахунків має підтвердити, або спростувати ці гіпотези. Отже одною з гіпотез було припущення, цілком раціональне, що проєктант використовує найбільш успішний (за власним опитом) алгоритм проєктування для конкретного типу судна. До речі, різних розробники можуть використовувати східні алгоритми, але не обов'язково. Якщо метод виявить характеристики такі, що судна одного типу та одного виробника мають близькі значення цих характеристик - факторів (не вважаючи на їх різні абсолютні та відносні розміри та характеристики), то ці судна можна групувати за ознакою значень факторів. Це означатиме, що отримано критерій вибору параметрів судна, а саме, обираються такі параметри, які дають значення факторів обраної групи суден.

Аналіз даних, що отримані за розробленою методикою, показує, що судна одних розробників дійсно групуються за значенням комплексів факторів, що не суперечить висунутої вище гіпотези про наявність успішних алгоритмів, якими користуються розробники проєктів суден. У випадку коли конструкторське бюро не має свого оригінального алгоритму визначення основних характеристик проєктованого судна на початковому етапі проєктування (немає практичного досвіду проєктування суден даного типу) або немає доступу до існуючих алгоритмів (які зазвичай становлять комерційну тайну), використання описаної методики факторного аналізу має допомогти з визначенням основних факторів які впливають на якість проєктування, розділити судна на групи за впливом факторів, вибрати групу суден,



характеристики якої найкраще відповідають технічному завданню і внаслідок цього отримати характеристики судна, сукупність параметрів якого не гірша за існуючі світові аналоги. Далі можливо застосування більш тонких механізмів оптимізації проекту за обраними економічними та технічними параметрами.

Зрозуміло, що використання описаної методики найбільш ефективно коли початкові дані містять достатню кількість проектів існуючих суден та мають відповідну якість.

## СПИСОК ДЖЕРЕЛ

1. В.А. Некрасов, Н.Н. Кабанова Задача функционирования судна, эксплуатирующегося на незакрепленных линиях перевозок, в проблеме его проектирования // НУК, Сборник научных трудов НУК, №6 (423), 2008 с. 35-40
2. Н.Н. Кабанова. Решение стохастической задачи оптимизации характеристик трампового судна // Вестник ОНМУ №34, 2012 с. 172 – 183
3. Дж.-О. Ким, Ч.У. Мюллер, У.Р. Клекка, М.С. Олдендерфер, Р.К. Блешфилд. Факторный, дискриминантный и кластерный анализ. «Финансы и статистика», М., 1989, 216 с.
4. Факторный анализ как статистический метод, Лоули Д., Максвелл А., 1967.
5. Jöreskog, Karl G. (1983). "Factor Analysis as an Errors-in-Variables Model". *Principals of Modern Psychological Measurement*. Hillsdale: Erlbaum. pp. 185–196.
6. Horn, John L. (June 1965). "A rationale and test for the number of factors in factor analysis". *Psychometrika*. 30 (2): 179–185.
7. Bandalos, D.L.; Boehm-Kaufman, M.R. (2008). "Four common misconceptions in exploratory factor analysis". In Lance, Charles E.; Vandenberg, Robert J. (eds.). *Statistical and Methodological Myths and Urban Legends: Doctrine, Verity and Fable in the Organizational and Social Sciences*. Taylor & Francis. pp. 61–87.
8. Tran, U. S., & Formann, A. K. (2009). Performance of parallel analysis in retrieving unidimensionality in the presence of binary data. *Educational and Psychological Measurement*, 69, 50-61.
9. Проектування самохідних прив'язних підводних систем, Бугаєнко Б.А., Магула В.Е., “Наукова думка”, 1997.

## ДОДАТОК А

## Код додатку

```

from tkinter import *
from tkinter import filedialog
import tkinter.scrolledtext as tkscrolled
import numpy as np
import pandas as pd

root = Tk()
color = '#fafa6e'

root['bg'] = '#fafafa'
root.title('Калькулятор параметрів судна')
root.wm_attributes('-alpha', 1.0)
root.geometry('600x500')
root.resizable(width=True, height=True)
root.config(bg=color)

position = {"padx": 6, "pady": 6}

# В каком виде будет работать приложение
label_1 = Label(root,
                 text='Оберіть спосіб обчислення',
                 bg=color,
                 font=('Ariel', 14, 'bold'),
                 )
label_1.pack(**position)

label = Label(root)
text_editor = tkscrolled.ScrolledText()

def open_file():
    if var.get() == 1:
        filepath = filedialog.askopenfilename()

        # Считываем датасет
        b = pd.read_excel(filepath, usecols=["L,м", "B,м", "H, м", "T,м",
        "V,уз", "P,кВт", "W,м.куб"])
        b = np.matrix(b)

        # Задаем постоянные значения плотности воды и ускорения свободного
        # падения
        p = 1024
        g = 9.81
        N = len(b)
        print(b)
        print('\n')

        # Создаем новый датасет безразмерных величин:
        print('Создаем новый датасет')
        X = np.transpose(b)
        L, B, H, T, P, V, W = X[0], X[1], X[2], X[3], X[4], X[5], X[6]

```

```

X1, X2, X3, X4, X5, X6 = L / B, B / H, H / T, W / L / B / T, V /
np.sqrt(g * L), P / p / g / W / V

Y1, Y2, Y3, Y4, Y5, Y6 = np.log(X1), np.log(X2), np.log(X3),
np.log(X4), np.log(X5), np.log(X6)

Yj = np.concatenate((Y1, Y2, Y3, Y4, Y5, Y6))
Yj = Yj.transpose()
print(Yj, len(Yj), N, '\n')

# Находим матрицу корреляции
print('Находим матрицу корреляции')
Ky = (Yj * np.transpose(Yj)) / 3
print(Ky, '\n')

# определяем собственные значения и собственные векторы матрицы K:
print('определяем собственные значения и собственные векторы матрицы
K:')
ar = np.linalg.eigh(Ky)
l = ar[0] # собственные значения
Kf = np.diag(l) # матрица собственных значений
v = ar[1] # собственные векторы
print(l, '\n\n', v, '\n\n', Kf, '\n')

# Определяется матрица A:
print('Определяется матрица A: ')
A = np.linalg.inv(v)
print(A, '\n')

# Определяем матрицу F (факторы):
print('Определяем матрицу F (факторы): ')
F = A * Yj
print(F, '\n')

# Определяем матрицу Kfy:
print('Определяем матрицу Kfy: ')
Kfy = (A * Yj * np.transpose(Yj)) / 3
print(Kfy, '\n')
plt.imshow(Yj)
plt.colorbar()
plt.show()

label['text'] = '\n'.join(' '.join(str(col) for col in row) for row in
Kfy)

text = label['text']
text_editor.delete(END)
text_editor.insert("1.0", text)
text_editor.pack(**position)

button = Button(root, text='зберегти файл', command=save)
button.pack(**position)

def save():
    filepath = filedialog.asksaveasfilename()
    if filepath != "":
        text = text_editor.get("1.0", END)

```

```
with open(filepath, "w") as file:  
    file.write(text)
```

```
variant_1 = 'обчислення на основі існуючого файлу'  
variant_2 = 'ввести дані вручну'  
lang = StringVar(value=variant_1) # по умовчанию стоить перший варіант  
  
var = IntVar()  
var.set(0)  
b_1 = Radiobutton(root, text=variant_1, value=1, variable=var, bg=color)  
b_1.pack(**position)  
b_2 = Radiobutton(root, text=variant_2, value=2, variable=var, bg=color)  
b_2.pack(**position)  
  
button_1 = Button(root,  
                  text='Обрати',  
                  font=('Ariel', 10, 'bold'), command=open_file)  
label = Label(bg=color)  
button_1.pack(**position)  
  
root.mainloop()
```

## Основний датасет

1	Проект	№	L,м	B,м	H, м	T,м	P,кВт	V,уз	W,м.куб
2	П041 КВК-16	1	13.7	3.74	2.1	0.8	970	30	16.98
3	"Ricochet " 1850	2	17.15	4.21	2.6	1.6	1080	22	40.98
4	"Ricochet " AP1601	3	14.65	3.78	2.8	1.5	300	12.5	32.88
5	"Алмаз" AP1600	4	14.65	3.78	2.8	1.5	780	22	32.88
6	PILOT 55	5	16.1	4.25	2.8	1.5	500	25	23.9
7	«Лощман Роцін»	6	14.3	4.26	2.7	1.75	533	17.5	25.37
8	Bintulu Bumi Armada	7	14.5	4.08	2.6	1.2	650	28	29.18
9	Atlantic pilotage Centre Port	8	14.02	3.83	2.7	0.84	790	26	18.92
10	Qwest	9	16	5.02	3	1.3	805	23	43.81
11	Dalian	10	17	4.85	3.15	1	900	24	33.93
12	Taranaki "Sea Tech" ST19L	11	17.7	4.59	3	1.4	670	27	47.2
13	Loyd P-153	12	17	4.85	3.12	1	750	32	32.95
14	Striker 54	13	17.6	5.02	3.15	1.2	1216	20	46.83
15	Striker 37 "Grandes Eaux"	14	13.9	3.76	1.8	0.6	618.24	32	11.71
16	17m Aluminum Pilot Boat - Jacobsen Pilot Services	15	14	5.08	2.49	1	861.12	23	29.71
17	Watercat 1100 Pilot	16	9.8	4.09	1.9	0.8	515.2	16	6.1
18	Watercat 1300 Pilot	17	16.7	5.7	2.89	1.15	1398.4	25	44.88
19	Watercat 1500 Pilot	18	14.7	4.9	2.7	1.6	1420.48	29	31.22
20	Watercat 1500 II Pilot	19	10.62	3.08	1.75	0.6	266	24	5.85
21	LOYD C237	20	12.83	3.96	1.8	0.8	610	30	13.66
22	LOYD 256	21	14.88	4.36	2.7	1.2	750	25	19.51
23	LOYD 28 PAX	22	15.62	4.22	2.7	0.85	750	35	20
24	KEWATE C PILOT 1200	23	15.81	4.21	2.68	0.8	1052.48	23	21.46
25	KEWATE C PILOT 1500	24	11.63	3.1	1.7	0.7	368	32	4.39
26	KEWATE C PILOT 1620	25	13.16	3.43	1.9	0.7	618.24	26	11.71
27	KEWATE C PILOT 1820	26	11.16	3.74	1.7	1.2	515.2	27	13.66
28	Chinook	27	14.32	4.54	2.6	1.2	662.4	27	16.59
29	Discovery	28	15.07	4.59	2.7	1.2	736	27	18.05
30	Grandes Eaux	29	16.93	4.95	3	1.3	1030.4	35	21.07
31	Supmar work boat	30	18.8	5.44	3.2	1.07	1920	30	45.07
32	R1490	31	19.9	5.17	3.1	1	1440	28	32.2
33		32	18.71	5.64	2.89	1.15	1398.4	25	44.88
34		33	14.7	4.85	2.6	1.5	715	25	42.73
35		34	15	4.08	2.2	2.3	700	10.5	59.8

36	"Пелла" 22580	35	18	4.68	2.44	1.9	1100	24	48.78
37	"Пелла" 22580	36	18	4.68	2.44	1.9	900	18	48.78
38	Skomer	37	17	5.36	2.5	1.5	652	19	0.45
39	Pilot 1800 PILOT	38	17.58	4.45	2.25	0.85	1000	26	21.46
40	1200	39	9.7	4.5	2.15	1.2	700	27	11.71
41	"2176"	40	13.44	4.48	2.2	1	1052	26.4	22.44
42	Loyd P- 215	41	16.15	4.14	2.2	0.8	2200	40	24.39
43	Loyd P187	42	14.2	4.7	2.65	1.2	1030	18	29.27
44	Loyd P- 153	43	14.2	3.76	1.8	0.6	840	32	11.71
45	Pilot 1800 PILOT	44	17.58	4.45	2.25	0.85	1000	26	21.46
46	1200	45	9.7	4.5	1.85	1.2	700	27	11.71
47	"2176"	46	13.44	4.48	2.2	1	1052	26.4	22.44
48	Loyd P030	47	16.87	4.98	2.2	1.15	1320	16.5	48.29
49	Loyd P150	48	18.05	4.89	2.8	1.4	1800	18	48.78

## ДОДАТОК Б

Таблиця 1. Основні характеристики лоцманських катерів

№	Назва проекту/судна	L	B	H	T	W	V	P
		м	м	м	м	м3	вуз.	кВт
1	2	3	4	5	6	7	8	9
1	Kewatec PILOT 1200	9,79	3,30	1,87	0,76	14,00	27,00	515,20
2	KEWATEC PILOT 1500	12,82	3,80	2,20	0,95	17,00	27,00	662,40
3	KEWATEC PILOT 1620	13,40	3,92	2,30	0,81	18,50	27,00	736,00
4	KEWATEC PILOT 1820	15,60	4,18	2,45	0,73	21,60	35,00	1030,40
5	17m Aluminum Pilot Boat – Jacobsen Pilot Services	14,70	4,40	4,00	1,20	32,00	29,00	1420,48
6	Pilot boat / inboard / aluminum / rigid hull inflatable boat	20,60	5,10	2,76	1,09	52,50	30,00	2162,00
7	17,3 Pilot boat	15,13	4,10	2,44	0,96	26,85	26,00	1066,00
8	Loyd UC-P153 14M PILOT AND PATROL BOAT	12,27	3,38	2,05	0,70	15,00	30,00	897,92
9	LOYD P-164 15m Pilot & Patrol Boat	12,68	3,42	2,40	0,60	12,00	32,00	618,24
10	LOYD C237 Aluminium Crew & Pilot Boat	14,44	3,72	2,10	0,95	22,00	23,00	1052,48
11	Watercat 140 Pilot	12,57	4,08	2,10	0,70	16,00	22,00	525,00
12	ALUSAFE 1620 BB PILOT	13,20	4,10	2,02	0,68	16,00	33,00	706,00
13	15m pilot boat AMS 1500	14,20	4,20	2,10	0,85	19,49	23,00	669,76
14	17m pilot boat AMS 1700	15,81	4,25	2,30	0,80	20,94	28,00	750,72
15	13.50m pilot boat AMS 1350	13,16	3,30	2,10	0,80	13,36	25,00	625,60
16	40' Resilient Class C. Raymond Hunt Associates	10,48	3,10	1,60	0,65	9,98	34,00	596,16
17	53' Chesapeake Class C. Raymond Hunt Associates	13,59	4,33	2,38	0,94	25,34	24,50	883,20
18	17m pilot boat AMS 1700	15,81	4,25	2,30	0,80	20,94	28,00	750,72
19	13.50m pilot boat AMS 1350	13,16	3,30	2,10	0,80	13,36	25,00	625,60
20	П041 KBK-16	13,70	3,74	2,10	0,80	17,40	30,00	970,00
21	Grandes Eaux	17,60	5,20	2,89	1,15	46,00	25,00	1398,40
22	18m Pilot Boat	15,88	4,26	2,65	1,15	34,00	20,00	883,20
23	17m Pilot boat Swede Ship Marine AB	15,04	4,20	2,30	0,90	26,00	25,00	780,16
24	StAN tender 1505	13,85	4,00	2,30	0,90	21,05	26,00	1100,00
25	Vigor 19m All-Weather Pilot Boat	17,31	5,30	2,80	0,91	37,82	20,00	1177,60
26	VIGOR 56' PILOT BOAT	14,96	4,20	2,60	1,10	30,50	27,00	1182,02



27	LOYD P-153 15M PILOT AND PATROL BOAT	13,16	3,30	2,10	0,60	12,00	32,00	618,24
28	Fast Pilot Boat 14 Meters Aluminium Fast Pilot Boat	12,39	3,20	2,10	0,60	12,00	24,00	596,16
29	Watercat 150 Pilot	13,28	3,90	2,10	0,70	15,00	22,00	525,00
30	Watercat 1500 Pilot	14,16	4,20	2,20	0,80	20,00	25,00	750,00

## Продовження Таблиці 1.

1	2	3	4	5	6	7	8	9
31	Watercat 1500 Pilot II	13,69	4,00	2,10	0,85	20,50	35,00	1104,00
32	KINGTOUGH 18M High Speed Pilot Boat	16,64	4,60	2,60	0,90	32,00	25,00	1176,00
33	56' pilot boat Camrac Design	14,93	3,95	2,40	1,00	27,67	27,00	1182,02
34	"M.V. Discovery"	18,54	4,30	2,80	1,00	33,00	28,00	1440,00
35	53' ST. JOHN'S CLASS	14,30	4,15	2,30	1,00	23,13	28,00	1030,40
36	AP 20 M Dockstar varvet	18,50	4,70	2,90	1,10	32,00	29,00	1102,00
37	AP 17 M - ver.2 Dockstar varvet	15,00	4,20	2,70	1,10	29,80	21,00	950,00
38	AP 17 M - ver.1 Dockstar varvet	15,00	4,20	2,70	1,10	32,80	29,00	1102,00
39	AP 14700 WJ	12,60	4,10	2,60	1,05	20,00	33,00	748,00
40	«Лощман Рошин»	14,30	4,26	2,70	1,00	26,00	17,50	533,00

Таблиця 2. Основні характеристики балкерів (неповний список)

№	Найменування	L <sub>нб</sub>	L <sub>пп</sub>	B	H	T	D <sub>b</sub>	D <sub>n</sub>	dw	V <sub>s</sub>	MC R	Трюмов	Кранов	Тип	KЗ П C <sub>b</sub>
		м	м	м	м	м	т	т	т	узл	kW	Од.	Шт.		-
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

1	NORD HONG KONG	17 9,9 0	171, 50	28, 40	14, 10	10,0 0	417 48	945 9	322 90	13, 80	600 0	5	4	Handy size	0,8 04
2	DRAGONER A	18 0,0 0	176, 63	30, 00	14, 70	10,1 0	450 09	103 96	346 13	14, 00	750 0	5	4	Handy size	0,7 89
3	HALKI	18 7,0 0	178, 02	27, 80	15, 60	10,9 0	472 46	103 96	368 50	14, 80	780 0	5	4	Handy size	0,8 14
4	THALASSINI AXIA	19 6,0 0	189, 00	32, 26	18, 60	12,5 0	699 19	113 11	589 23	14, 60	996 0	5	4	Supra max	0,8 56
5	ARKADIA	19 7,0 8	189, 00	32, 26	18, 50	12,6 0	684 18	120 70	551 61	14, 50	116 20	5	4	Handy max	0,8 35
6	AMBER CHAPION	19 9,8 5	194, 50	32, 26	18, 50	12,7 0	754 71	116 71	638 00	14, 48	805 0	5	4	Handy max	0,8 77
7	STX ARBORELLA	19 9,8 2	192, 67	32, 26	19, 30	12,7 2	706 05	130 65	575 39	13, 90	996 0	5	4	open- hatch	0,8 34
8	SAGE AMAZON	19 9,9 9	193, 74	32, 26	18, 50	13,3 0	749 22	116 21	633 01	14, 50	830 0	5	4	Supra max	0,8 46
9	AP LIBERTAS	22 5,0 0	217, 00	32, 26	19, 60	14,2 0	870 72	118 59	752 13	14, 00	899 0	7	-	Panam ax	0,8 22
10	GIEWONT	22 9,8 2	222, 82	32, 26	20, 25	14,6 2	933 77	137 27	796 49	14, 28	110 60	7	-	Kamsa rmax	0,8 34
11	NIKI	25 3,9 2	241, 00	40, 00	21, 00	14,6 2	117 602	176 02	100 000	15, 00	111 80	5	4	Capesi ze	0,7 83
12	OCEAN GARNET	22 9,2 0	222, 00	38, 00	20, 70	14,9 0	108 709	156 90	930 18	14, 35	122 40	7	-	Large bulk	0,8 11
13	ANANGEL DAWN	24 9,8 5	240, 00	43, 00	21, 30	15,0 0	134 376	202 85	114 091	14, 70	158 20	7	-	Handy cape	0,8 14
14	STX FREESIA	29 2,0 0	283, 00	45, 00	24, 80	18,2 2	207 383	266 47	180 736	15, 00	179 00	9	-	Dunki rkmax	0,8 38
15	K. HOPE	33 0,0 7	321, 00	57, 00	25, 10	18,0 0	285 100	351 00	250 038	14, 95	230 00	9	-	VLOC	0,8 12
16	MIEDWIE	19 0,0 0	182, 60	23, 00	14, 60	10,4 0	391 74	919 0	299 84	14, 00	780 0	6	3	Great Lakes	0,8 45
17	IVS VISCOUNT	17 8,8 0	172, 00	28, 00	14, 80	10,6 5	437 57	905 8	346 99	14, 00	650 0	5	4	Handy size	0,8 05
18	SPAR LYRA	19 0,0 0	183, 44	32, 26	17, 40	12,5 0	650 00	116 00	535 65	14, 00	948 0	5	4	Handy max	0,8 24
19	TAI PROGRESS	22 5,0 0	217, 00	32, 26	19, 50	14,1 0	883 66	105 32	7783 4	14, 00	100 02	7	-	Pana max	0,8 40
20	KOHYOHSAN	28 9,0 0	280, 20	45, 00	24, 10	17,7 8	193 802	212 38	1725 64	14, 70	147 11	9	-	Capesi ze /dunki rkmax	0,8 11
21	<a href="#">WORLDERA 2</a>	18 9,9 6	182, 00	32, 24	16, 50	11,6 2	598 57	109 50	4890 7	14, 00	770 7	5	4	Handy max	0,8 16
22	<a href="#">MV MARILIA</a>	19 0,5 0	183, 68	32, 01	16, 40	11,8 5	592 80	106 40	4864 0	12, 50	104 00	5	4	Handy max	0,8 06

23	BURGIA	22 9,00	222, 00	32, 26	20, 25	14,6 2	932 88	138 85	7940 3	14, 00	116 20	7	-	Kamsa rmax	0,8 36
24	<a href="#">AGRI</a> <a href="#">GRANDE</a>	22 9,00	225, 32	32, 26	20, 00	14,4 5	953 74	134 08	8196 6	14, 00	980 1	7	-	Kamsa rmax	0,8 52

### Матриця факторів

[[ 8.45516515e-14 2.79488588e-14 -4.46210773e-14 1.45138403e-14  
4.87424911e-14 1.27996123e-14]  
[-7.85907990e-14 1.63831646e-14 4.05794038e-14 -1.27512251e-14  
-6.77350470e-14 -1.39074067e-14]  
[ 2.91628801e-14 2.25321894e-14 -3.27904710e-15 -1.70130077e-15  
-1.01179871e-14 -3.64806492e-15]  
[-7.17694873e-14 -3.29686099e-14 3.07240584e-14 -1.25809816e-14  
-4.47734732e-14 -2.46150706e-14]  
[-1.82854821e-14 1.13106723e-14 2.58634465e-14 -3.87088105e-15  
-2.95494208e-14 -5.32680089e-15]  
[-5.25767354e-17 -3.95130656e-15 -8.33458262e-15 9.34718465e-16  
7.80808901e-15 -7.25003200e-15]  
[ 3.27656830e-14 -5.01041058e-14 -3.56860220e-14 -3.63233170e-15  
-5.72088000e-16 1.43368148e-15]  
[ 1.30597915e-14 -6.89517711e-15 -9.75320888e-16 8.92865320e-16  
1.34846084e-14 1.99797043e-14]  
[ 9.40049482e-15 2.30186715e-14 7.32636670e-15 -6.14293897e-15  
-3.81921972e-15 -1.90259218e-14]  
[ 3.77118380e-14 -3.21293539e-15 1.93915132e-15 4.40438611e-15  
-5.17524161e-15 4.15548554e-14]  
[ 3.11497593e-14 4.43954503e-14 -2.95131205e-14 8.14545058e-15  
1.68317014e-14 -3.69632001e-15]  
[-4.46859412e-14 1.01506342e-14 2.35405636e-14 -1.12399806e-15  
-2.69824538e-14 -8.64986813e-16]  
[-1.15147729e-14 -3.65546471e-14 -1.30319101e-14 4.16635796e-16  
1.98318534e-14 -5.07176370e-15]  
[-2.66395444e-14 -1.80003117e-14 1.48118631e-14 -9.51391100e-15  
-4.37486068e-15 -2.53002731e-14]  
[-3.40161286e-15 2.78935660e-14 1.80089872e-14 5.75557874e-16  
-2.85816009e-15 -1.67828172e-15]  
[-9.28998867e-15 1.27456206e-14 1.29639274e-14 -1.30347619e-15  
-1.24408984e-14 -4.22457012e-15]  
[ 5.29061765e-15 8.37824658e-15 5.86417256e-15 -9.03630651e-16  
4.27392915e-15 -1.46756542e-14]  
[ 1.70012314e-14 3.56149736e-15 -1.31654680e-14 6.44912835e-15

1.16378331e-14 1.13338164e-14]  
[-1.18989779e-14 -8.06322162e-15 -2.35310299e-15 -3.16182255e-15  
1.11802562e-15 -1.19015067e-14]  
[ 4.19020251e-17 1.67511435e-14 -1.39257959e-14 6.72204783e-15  
1.84153533e-14 1.83063876e-14]  
[ 1.67681212e-14 2.06020540e-14 -1.72169006e-14 4.43039721e-15  
2.53369992e-15 1.06274401e-14]  
[-7.50637976e-15 -8.71705313e-15 -4.64004149e-15 -1.68799844e-15  
3.03499748e-15 -1.00512704e-14]  
[-2.70644157e-14 2.97379348e-14 1.98738881e-14 -9.10372255e-16  
-1.06747622e-14 -1.07933681e-14]  
[ 2.39962842e-14 2.33347595e-14 6.51228633e-15 -1.50568024e-15  
6.75050213e-15 -1.73861907e-14]  
[-1.49021042e-14 -3.07032788e-14 -3.24190890e-15 -1.01675790e-14  
-3.17796121e-15 -2.64262001e-14]  
[-1.83127200e-14 -1.59184540e-14 4.96707479e-15 -3.04182556e-15  
-1.28732045e-14 7.42242911e-15]  
[-3.23867268e-14 -9.39619598e-15 9.17237757e-15 -1.85654678e-15  
-7.57152747e-15 -1.48405116e-14]  
[ 3.10977397e-15 1.11303949e-14 1.43950217e-14 -6.97633500e-15  
-1.95407509e-14 -2.00929948e-15]  
[ 3.17582372e-14 2.34253824e-14 -1.28610358e-14 7.94180527e-15  
1.99289578e-14 1.65143780e-14]  
[-4.66003815e-15 -2.05585887e-15 1.58149901e-15 -3.54622571e-15  
1.85256893e-14 -9.94687870e-15]  
[ 2.45802549e-14 -3.74969737e-14 -3.39594220e-14 9.94113349e-16  
2.20220877e-14 6.90812640e-15]  
[ 3.16057668e-15 6.26158964e-15 -1.09367726e-14 5.62362627e-15  
-8.54060006e-16 1.00383038e-14]  
[ 3.55995971e-15 4.14783900e-14 6.48710785e-15 2.08645732e-15  
-2.30054730e-14 1.59994676e-14]  
[-1.81266905e-14 -3.05890401e-14 -8.72640614e-15 -6.07961467e-15  
-5.95806378e-15 -6.35360786e-16]  
[ 4.64409846e-14 5.56983768e-14 -1.25802633e-14 1.17655102e-14  
1.39744001e-14 5.45001003e-15]  
[ 5.85339235e-15 5.28988104e-14 5.48957741e-16 2.46590656e-15  
5.86481492e-15 -7.20137978e-15]  
[ 6.00612924e-15 1.31972384e-14 -8.16514302e-15 3.36747127e-15  
-2.50859675e-14 6.74863470e-15]  
[-9.39993780e-15 2.46633636e-15 1.12026460e-14 -2.55690192e-15  
-1.04469784e-14 -1.58274153e-14]

[-2.70337377e-14 2.88509626e-14 1.50798505e-14 1.76919772e-15  
-2.71769719e-14 1.70897599e-14]  
[-2.50172321e-14 1.58819800e-14 -9.76298716e-17 -6.07791488e-15  
-5.11258620e-14 -7.05854370e-15]  
[ 5.97866021e-14 7.28568097e-14 -1.82949581e-14 1.75329903e-14  
6.42230998e-14 1.99149917e-14]  
[ 1.70583233e-14 1.69355340e-14 -3.41229385e-15 4.13487604e-15  
3.23914221e-15 1.56045408e-14]  
[-6.54200411e-01 -3.74267915e-01 1.63072754e-01 -8.73151569e-02  
-9.83838413e-02 -9.71336475e-02]  
[-3.07390377e-01 7.49382730e-01 2.92551012e-01 3.68894726e-02  
-3.54590779e-01 -2.99423435e-05]  
[ 6.50464368e-01 -5.15841185e-01 8.71205361e-01 -4.90237339e-02  
-9.40372405e-01 6.58532774e-02]  
[-7.01850670e-02 -1.07858197e-01 -1.70730562e+00 -9.79948977e-02  
-1.58586563e+00 -2.83651186e-01]  
[-2.72779335e-01 -4.92711365e-01 5.37568075e-02 5.35838907e+00  
-2.02942042e-01 -7.85285922e-01]  
[ 8.45861879e+00 4.02439771e+00 5.47659520e+00 -8.01065808e+00  
4.86255535e+00 -6.10055992e+00]

