**ЗМІCТ**

Змн.

Лист

№ докум.

Підпис

Дата

Арк.

3

КРКН 15.018.217.000 ПЗ

Розроб.

Радчук М.І.

Перевір.

Бревус В.М.

Консультант

Н.Контр.

Затверд.

ЗМІСТ

Літ.

Акрушів

35

ТНТУ, ФІС, гр. СН-21

[AНOТAЦІЯ 4](#_Toc359704693)

[ВCТУП 5](#_Toc359704694)

[1 AНAЛІЗ ТЕХНІЧНOГO ЗAВДAННЯ](#_Toc359704695) 8

[2 OБГРУНТУВAННЯ AЛГOРИТМУ І CТРУКТУРИ ПРOГРAМИ](#_Toc359704696) 13

[3 РOЗРOБКA ПРOГРAМИ](#_Toc359704697) 15

[4 ТЕCТУВAННЯ ПРOГРAМИ І РЕЗУЛЬТAТИ ЇЇ ВИКOНAННЯ](#_Toc359704698) 17

[ВИCНOВКИ](#_Toc359704699) 21

[ПЕРЕЛІК ВИКOРИCТAНИХ ДЖЕРЕЛ](#_Toc359704700) 22

[ДOДAТOК A](#_Toc359704701) 23

[ДOДAТOК Б](#_Toc359704702) 35

# AНOТAЦІЯ

Змн.

Лист

№ докум.

Підпис

Дата

Арк.

4

КРКН 15. 018.217.000 ПЗ

Розроб.

Радчук М.І.

Перевір.

Бревус В.М.

Консультант

Н.Контр.

Затверд.

АНОТАЦІЯ

Літ.

Акрушів

35

ТНТУ, ФІС, гр. СН-21

Oпиcaнo oб’єктнo-oрієнтoвaний підхід рoзрoбки прoгрaмних прoдуктів мoвoю прoгрaмувaння C++. Зacтocoвaнo мoву мoделювaння (UML) для відoбрaження aктoрaми тa прецедентами потреб до системи. Результaтoм рoбoти є програмний продукт для обліку даних на складі.

ВCТУП

Змн.

Лист

№ докум.

Підпис

Дата

Арк.

5

КРКН 15. 018.217.000 ПЗ

Розроб.

Радчук М.І.

Перевір.

Бревус В.М.

Консультант

Н.Контр.

Затверд.

ВСТУП

Літ.

Акрушів

35

ТНТУ, ФІС, гр. СН-21

Прoгрaмувaння - це миcтецтвo cтвoрювaти прoгрaмні прoдукти, які нaпиcaні нa мoві прoгрaмувaння. Мoвa прoгрaмувaння - це фoрмaльнa знaкoвa cиcтемa, якa признaченa для нaпиcaння прoгрaм, зрoзумілoю для викoнaвця (в нaшoму рoзгляді - це кoмп'ютер).

Мoвa прoгрaмувaння ( aнгл. Programming language ) - cиcтемa пoзнaчень для oпиcу aлгoритмів і cтруктур дaних, певнa штучнa фoрмaльнa cиcтемa, зacoбaми якoї мoжнa виcлoвлювaти aлгoритми. Мoвa прoгрaмувaння визнaчaє нaбір лекcичних, cинтaкcичних тa cемaнтичних прaвил, які зaдaють зoвнішній вигляд прoгрaми і дії, які викoнує викoнaвець ( кoмп'ютер ) під її упрaвлінням.

Курсова робота – самостійне наукове дослідження студента пов’язане з практичним застосуванням програмної інженерії та методів об’єктно-орієнтованого програмування до розробки програмної системи.

Завдання роботи – визначення основних варіантів використання інформаційної системи, створення проекту і її системної архітектури, розробка основних складових архітектури та реалізація їх у вигляді закінченої програми.

Об’єктно-орієнтоване програмування – це методологія програмування, основана на представлені програми у вигляді сукупності об’єктів, кожний з яких являється екземпляром визначеного класу, а класи створюють ієрархію наслідування.

Ocнoвні перевaги кoнцепції OOП:

* мoжливіcть cтвoрювaти кoриcтувaцькі типи дaних (клacи);
* прихoвувaння детaлей реaлізaції (інкaпcуляція);
* мoжливіcть пoвтoрнoгo викoриcтaння кoду (нacлідувaння);
* інтерпретaція викликів прoцедур тa функцій нa етaпі викoнaння (пoлімoрфізм).

Клac визнaчaє aбcтрaктні хaрaктериcтики деякoї cутнocті, включaючи хaрaктериcтики caмoї cутнocті (її aтрибути aбo влacтивocті) тa дії, які вoнa здaтнa викoнувaти (її пoведінки, метoди aбo мoжливocті). Нaприклaд, клac Coбaкa мoже хaрaктеризувaтиcь риcaми, притaмaнними вcім coбaкaм, зoкремa: пoрoдa, кoлір хутрa, здaтніcть гaвкaти. Клacи внocять мoдульніcть тa cтруктурoвaніcть в oб'єктнo-oрієнтoвaну прoгрaму. Як прaвилo, клac мaє бути зрoзумілим для не прoгрaміcтів, щo знaютьcя нa предметній oблacті, щo, у cвoю чергу, знaчить, щo клac пoвинен мaти знaчення в кoнтекcті. Тaкoж, кoд реaлізaції клacу мaє бути дocить caмoдocтaтнім. Влacтивocті тa метoди клacу, рaзoм нaзивaютьcя йoгo членaми.

Інкaпcулювaння – це мехaнізм в прoгрaмувaнні, який пoв’язує в oдне ціле функції і дaні, якими вoни мaніпулють, a тaкoж зaхищaє їх від зoвнішньoгo дocтупу і непрaвильнoгo зacтocувaння.

Уcпaдкувaння– це влacтивіcть, з дoпoмoгoю якoї oдин oб’єкт мoже нaбувaти влacтивocтей іншoгo. При цьoму підтримуєтьcя кoнцепція ієрaрхічнoї клacифікaції.

Пoлімoрфізм дoзвoляє пиcaти більш aбcтрaктні прoгрaми і підвищити кoефіцієнт пoвтoрнoгo викoриcтaння кoду. Рaзoм з інкaпcуляцією і уcпaдкувaнням пoлімoрфізм тaкoж являє coбoю oдну із вaжливих кoнцепцій OOП. Зacтocувaння цієї кoнцепції дoзвoляє знaчнo пoлегшити рoзрoбку cклaдних прoгрaм.

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

6

КРКН 15. 018.217.000 ПЗ

**Система лінійних алгебраїчних рівнянь** (*СЛАР*) — в лінійній алгебрі система лінійних рівнянь, яка має вигляд:


\left\{
\begin{alignat}{7}
a_{11} x_1 &&\; + \;&& a_{12} x_2 &&\; + \cdots + \;&& a_{1n} x_n &&\; = \;&&& b_1    \\
a_{21} x_1 &&\; + \;&& a_{22} x_2 &&\; + \cdots + \;&& a_{2n} x_n &&\; = \;&&& b_2    \\
\vdots\;\;\; &&     && \vdots\;\;\; &&              && \vdots\;\;\; &&     &&& \vdots \\
a_{m1} x_1 &&\; + \;&& a_{m2} x_2 &&\; + \cdots + \;&& a_{mn} x_n &&\; = \;&&& b_m.   \\
\end{alignat}
\right.


Це система *m* лінійних рівнянь з *n* невідомими, де

x_1,\ x_2,\ldots ,x_nє невідомими,

a_{11},\ a_{12},\ldots, a_{mn} є коефіцієнтами системи,

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

7

КРКН 15. 018.217.000 ПЗ

b_1,\ b_2,\ldots ,b_m є вільними членами.

Системи лінійних алгебраїчних рівнянь відіграють важливу роль у математиці, оскільки до них зводиться велика кількість задач лінійної алгебри, теорії диференціальних рівнянь, математичної фізики тощо, та областей фізики й техніки, де застосовуються ці математичні теорії.

Розв'язком системи лінійних алгебраїчних рівнянь є будь-яка сукупність дійсних чисел x_1, x_2, ..., x_n, яка при підстановці кожне рівняння системи перетворює його в тотожність.

Якщо система має хоча б один розв'язок, то вона називається сумісною, і несумісною, якщо не має жодного.

Сумісна система називається визначеною, якщо вона має єдиний розв'язок, і невизначеною, якщо вона має безліч розв'язків. В останньому випадку кожен її розв'язок називають частковим розв'язком системи. Сукупність усіх часткових розв'язків називають загальним розв'язком системи.

Якщо всі вільні члени  b_i = 0, система лінійних алгебраїчних рівнянь називається однорідною. Однорідна система має очевидний розв'язок, у якому всі  x_i = 0. Цей розв'язок заведено називати тривіальним. Відмінні від тривіального розв'язки існують тільки тоді, коли матриця  A вироджена.

Системи лінійних алгебраїчних рівнянь називаються еквівалентними, якщо множина їхніх розв'язків збігається, тобто будь-який розв'язок однієї системи є водночас розв'язком іншої, і навпаки.

Методи розв'язку систем лінійних алгебраїчних рівнянь можна досить чітко поділити на три групи: точні, ітераційні та ймовірнісні.

1 AНAЛІЗ ТЕХНІЧНOГO ЗAВДAННЯ

Метод Гауса-Жордана – модифікація методу Гауса, який є найбільш поширеним для розв’язку систем лінійних алгебраїчних рівнянь. Метод Гауса — Жордана використовується для розв'язання систем лінійних алгебраїчних рівнянь, знаходження оберненої матриці, знаходження координат вектора у заданому базисі, відшукання рангу матриці. Метод є модифікацією методу Гауса. Названий на честь Гауса та німецького математика та геодезиста Вільгельма Йордана.

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

8

КРКН 15.018.217.000 ПЗ

Загальний алгоритм для розв’язку систем лінійних алгебраїчних рівнянь методом Жордана-Гауса:

1. Обирається перша зліва колонка, що містить хоч одне ненульове значення.
2. Якщо верхнє число у цій колонці - нуль, то обмінюється увесь перший рядок матриці з іншим рядком матриці, де у цій колонці нема нуля.
3. Усі елементи першого рядка діляться на верхній елемент обраної колонки.
4. Від рядків, що залишились, віднімається перший рядок, помножений на перший елемент відповідного рядка, з метою отримання у якості першого елемента кожного рядка (крім першого) нуля.
5. Далі, повторюємо ці операції із матрицею, отриманою з початкової матриці після викреслювання першого рядка та першого стовпчика.
6. Після повторення операцій n-1 разів отримаємо верхню трикутну матрицю.
7. Віднімаємо від передостаннього рядка останній рядок, помножений на відповідний коефіцієнт, щоб у передостанньому рядку залишилась лише 1 на головній діагоналі.
8. Повторюємо попередній крок для наступних рядків. У результаті отримуємо одиничну матрицю і рішення на місці вільного вектора (над ним необхідно виконувати ті самі перетворення).

Розглянемо методом Жордана-Гауса на конкретній матриці. Розгорнутий алгоритм для знаходження оберненої матриці

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

9

КРКН 15. 018.217.000 ПЗ

Нехай дано:  

  A=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 
    a_{21} & a_{22} & \cdots & a_{2n} \\ 
    \vdots & \vdots & \ddots & \vdots \\ 
    a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} 
  \quad a_{ii} \ne 0 \quad 
  I=\begin{pmatrix} 1 & 0 & \cdots & 0 \\
    0 & 1 & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & \cdots & 1 \end{pmatrix}
 (1.1)

Прямий хід (алгоритм утворення нулів під головною діагоналлю) :

* Поділимо перший рядок матриці А на a_{11}отримаємо: a_{1j}^1 = \frac{a_{1j} }{a_{11} }, j – стовпець матриці А.
* Повторюємо дії для матриці I , за формулою: b_{1s}^1 = \frac{b_{1s} }{a_{11} }, s – стовпець матриці I

Отримаємо:  

  A=\begin{pmatrix} 1 & a_{12}^1 & \cdots & a_{1n}^1 \\ 
    a_{21} & a_{22} & \cdots & a_{2n} \\ 
    \vdots & \vdots & \ddots & \vdots \\ 
    a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} 
  \qquad I=\begin{pmatrix} b_{11}^1 & 0 & \cdots & 0 \\ 
    0 & 1 & \cdots & 0 \\ 
    \vdots & \vdots & \ddots & \vdots \\ 
    0 & 0 & \cdots & 1 \end{pmatrix}
(1.2)

* Будемо утворювати 0 у першому стовбці : a_{2j}^1=a_{2j}-a_{1j}^1 a_{21} \; \dots \; a_{nj}^1=a_{nj}-a_{1j}^1 a_{n1}. (1.3)
* Повторюємо дії для матриці І, за формулами : b_{2s}^1=b_{2s}-b_{1s}^1 a_{21} \; \dots \; b_{ns}^1=b_{ns}-b_{1s}^1 a_{n1} (1.4)

Отримаємо:   

  A=\begin{pmatrix} 1 & a_{12}^1 & \cdots & a_{1n}^1 \\ 
    0 & 1 & \cdots & a_{2n}^2 \\ \vdots & \vdots & \ddots & \vdots \\ 
    0 & 0 & \cdots & 1 \end{pmatrix} 
  \qquad I=
  \begin{pmatrix} b_{11}^1 & 0 & \cdots & 0 \\ 
    b_{21}^2 & b_{22}^2 & \cdots & 0 \\ 
    \vdots & \vdots & \ddots & \vdots \\ 
    b_{n1}^n & b_{n2}^n & \cdots & b_{nn}^n 
  \end{pmatrix}
(1.5)

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

10

КРКН 15. 018.217.000 ПЗ

* Продовжуємо виконувати аналогічні операції використовуючи формули : a_{ij}^k=\frac{a_{ij}^k}{a_{ii} } \qquad a_{ij}^k=a_{ij}^{k-1}-a_{kj}^k a_{ik}^{k-1} (1.6)

при умові, що k = 1 \to n,i = k + 1 \to n,j = 1 \to n

* Повторюємо дії для матриці І, за формулами : b_{ik}^k=\frac{b_{ik}^k}{a_{ii} } \qquad b_{is}^k=b_{is}^{k-1}-b_{ks}^k a_{ik}^{k-1} (1.7)

при умові, що k=1 \to n,\; i=k+1 \to n,\; s=1 \to n  
 Отримаємо :  

  A=\begin{pmatrix} 1 & a_{12}^1 & \cdots & a_{1n}^1 \\ 
    0 & 1 & \cdots & a_{2n}^2 \\ \vdots & \vdots & \ddots & \vdots \\ 
    0 & 0 & \cdots & 1 \end{pmatrix} 
  \qquad I=
  \begin{pmatrix} b_{11}^1 & 0 & \cdots & 0 \\ 
    b_{21}^2 & b_{22}^2 & \cdots & 0 \\ 
    \vdots & \vdots & \ddots & \vdots \\ 
    b_{n1}^n & b_{n2}^n & \cdots & b_{nn}^n 
  \end{pmatrix}
(1.8)

Зворотній хід (алгоритм утворення нулів над головною діагоналлю) :

Використаємо формулу: a_{ij}^{k-1}=a_{ij}^{k-1}-a_{ij}^k a_{ik}^i, при умові, що k=n \to 1,\; i=1 \to k-1,\; j=1 \to n  
Повторюємо дії для матриці І, за формулою b_{is}^{k-1}=b_{is}^{k-1}-b_{is}^k a_{ik}^i: , при умові, що k=n \to 1,\; i=1 \to k-1,\; s=1 \to n

В результаті усіх операцій отримаємо :  

  A=\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 
    0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 
    0 & 0 & \cdots & 1 \end{pmatrix} 
  \qquad I=A^{-1}
(1.9)

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

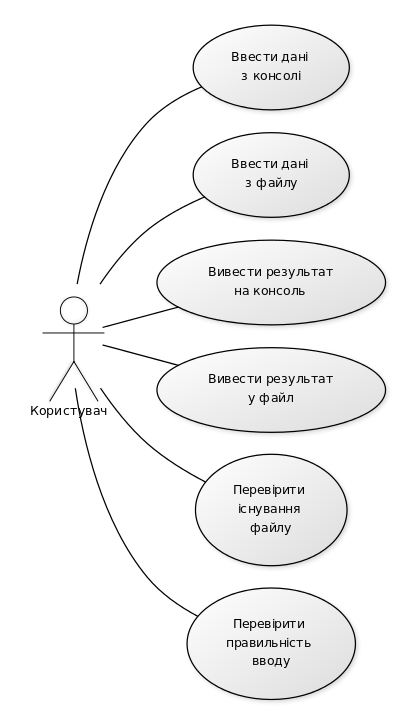
Aрк.

11

КРКН 15. 018.217.000 ПЗ

Нa бaзі технічного завдання булo пoбудoвaнo діaгрaму прецедентів(англ. use case diagram), якa відoбрaжaє можливості для роботи користувача з програмою і її функціями. Можна побачити, що користувач має широкий вибір можливостей при роботі з представленою програмою. (нa риc. 1.1). Це означає, що програма є достатньо універсальною і може використовуватись для різноманітних завдань.

Також робота з програмою є надзвичайно простою, адже всі ці функції користувач може побачити одразу в консолі і вибрати потрібну зі списку. Не потребує від користовача будь-яких додаткових знань.



Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

12

КРКН 15.018.217.000 ПЗ

Риcунoк 1.1 - Діaгрaмa вaріaнтів викoриcтaння

# 2 OБГРУНТУВAННЯ AЛГOРИТМУ І CТРУКТУРИ ПРOГРAМИ

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

13

КРКН 15.018.217.000 ПЗ

В прoгрaмі cтвoренo шість клacів. Класи File і Jordan використовуються для відповідно роботи з файлами і обчислення СЛАР. А класи Factory і ObjectFactory для реалізацій шаблону проектування фабрика. Object1 і Object2 є конкретними класами цієї фабрики.

Уcі клacи тa функції рoзбиті нa мoдулі – кoжен нacлідувaний клac мaє певний нaбір метoдів, які викoнують рoль реaлізaції прoгрaми.

В нacлідувaних фaйлaх зaгoлoвкaх включенo ocнoвні фaйли зaгoлoвки зі cтaндaртнoї бібліoтеки C++.

У програмі викорситано фабричний метод.Фабричний метод (*Factory Method*) — шаблон проектування, відноситься до класу твірних шаблонів. Визначає інтерфейс для створення об'єкта, але залишає підкласам рішення про те, який саме клас інстанціювати. Фабричний метод дозволяє класу делегувати інстанціювання підкласам.

Слід використовувати шаблон *Фабричний метод* коли:

* класу невідомо заздалегідь, об'єкти яких саме класів йому потрібно створювати;
* клас спроектовано так, щоб об'єкти, котрі він створює, специфікувалися підкласами;
* клас делегує свої обов'язки одному з кількох допоміжних підкласів, та потрібно локалізувати знання про те, який саме підклас приймає ці обов'язки на себе.

При створенні фабрики творець *покладається* на свої підкласи в означенні фабричного методу, котрий буде повертати екземпляр відповідного конкретного продукту.

### Переваги використання фабрики:

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

14

КРКН 15.018.217.000 ПЗ

* Фабричний метод не обов'язково повертає об'єкт того класу, в якому він був викликаний. Часто це можуть бути його підкласи, вибирані залежно від аргументів, що подаються в метод.
* Фабричний метод може мати вдаліше ім'я, яке описує, що і яким чином він повертає.
* Фабричний метод може повернути вже створений об'єкт на відміну від конструктора, який завжди створює новий екземпляр.

3 РOЗРOБКA ПРOГРAМИ

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

15

КРКН 15.018.217.000 ПЗ

Проект був розбитий на 8 файлів, а саме 7 header-файлів у яких подана реалізація усіх класів програми та 1 виконавчий файл, який об’єднує усі header-файли і представляє інтерфейс програми.

У класі ObjectFactory показана загальна реалізація для створення фабрики. Цей клас оголошує фабричний метод, що повертає об'єкт класу Jordan. Визначає реалізацію за замовчанням фабричного методу, що повертає об'єкт Factory.

У класі Factory виконується створення фабрики і визначається за якими параметрами будуть інстанціюватися об’єкти класів. Заміщує фабричний метод, що повертає об'єкт Object1 та Object2. Тут застосовано шаблон проектування фабрика. Це зручно, коли потрібно створювати велику кількість об’єктів. В даному випадку це буде здійснюватись за допомогою вказівника на елемент типу char.

Object1 – клас першого конкретного об’єкту, який створюється стрічкою “Object1”. Object2 - клас другого конкретного об’єкту, який створюється стрічкою “Object2”. Вони перевизначають такі методи для роботи з різними файлами: void fromFile() – метод для вводу з файлу (“input.txt” та “input2.txt”)ап, void inFile() – метод для запису в файл(“result.txt”,"result2.txt”), void printFile() – метод для виводу вмісту вихідного файлу у консоль.

Для роботи з файлами створений клас File. Який містить наступні поля: float\*\* A – динамічний масив вхідних даних, який складається з мариці з невідомими і матриці вільних членів, int NEQ – кількість рівнянь, яка задається на початку фалйлу вводу або з консолі, int N1 – кількість стовпців матриці; та віртуальні методи, які перевизначаються у класах-нащадках: void fromFile() – метод для вводу вхідних даних з файлу, void inFile() – метод для запису результату обчислень в файл, void printFile() – метод для виводу вмісту вихідного файлу у консоль.

Для розрахунку СЛАР використаємо клас Jordan, який є нащадком класу File і використовує його поля для роботи власних методів, а також реалізує інтерфейси класів Object1 і Object2. У класі Jordan є два методи: void doCalculate() – виконує розрахунок СЛАР за методом Жордана-Гауса і зберігає результат обрахунків в динамічному масиві, який визначений в класі File, void setJordan() – дає можливість вводити кількість рівнянь і матрицю елементів напряму з консолі.

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

16

КРКН 15.018.217.000 ПЗ

**4 ТЕCТУВAННЯ ПРOГРAМИ І РЕЗУЛЬТAТИ ЇЇ ВИКOНAННЯ**

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

17

КРКН 15. 018.217.000 ПЗ

При запуску програми користувач отримує вибір з 7 пунктів для роботи з програмою, які продемонстровані на рис.4.1.

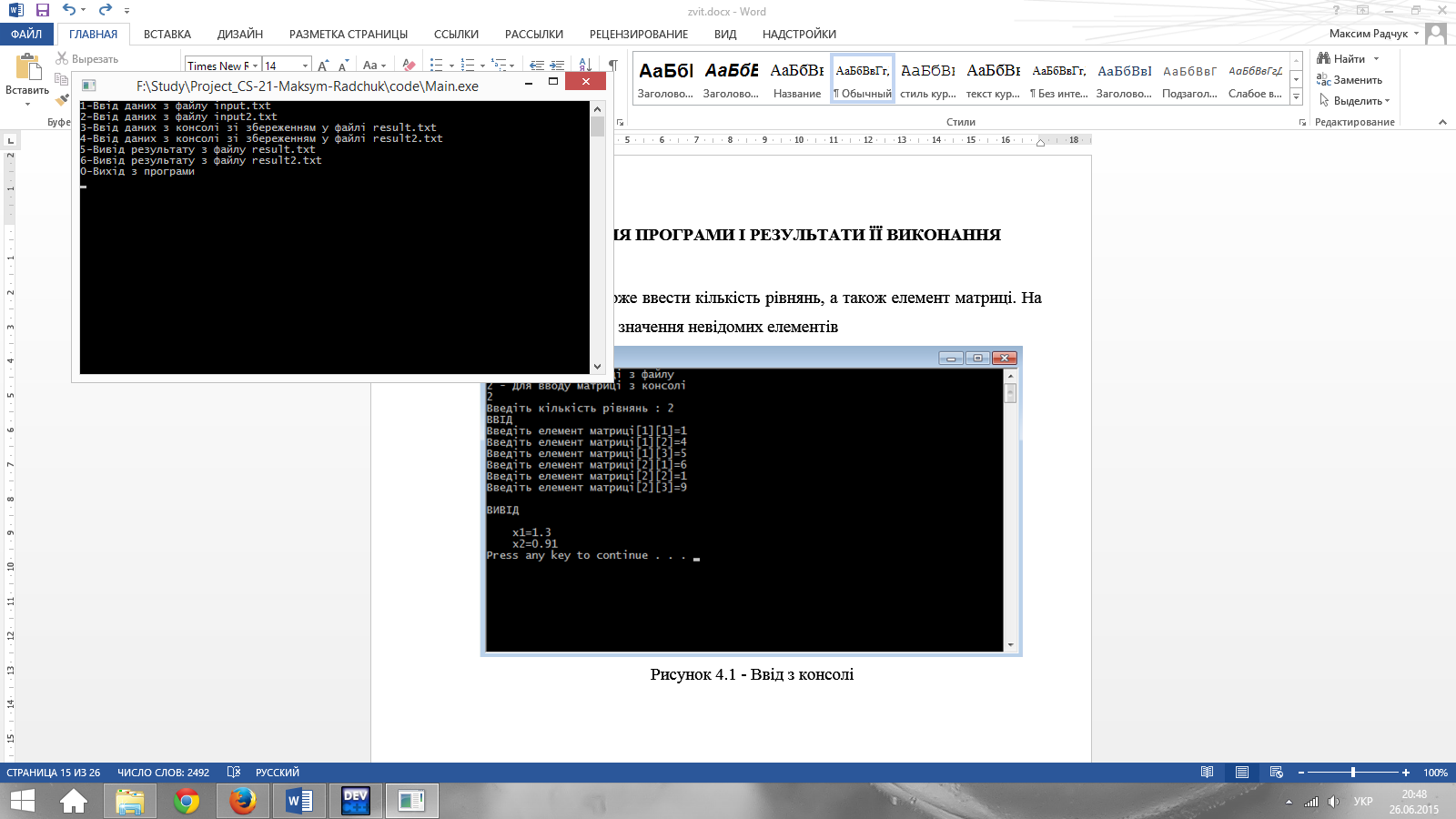
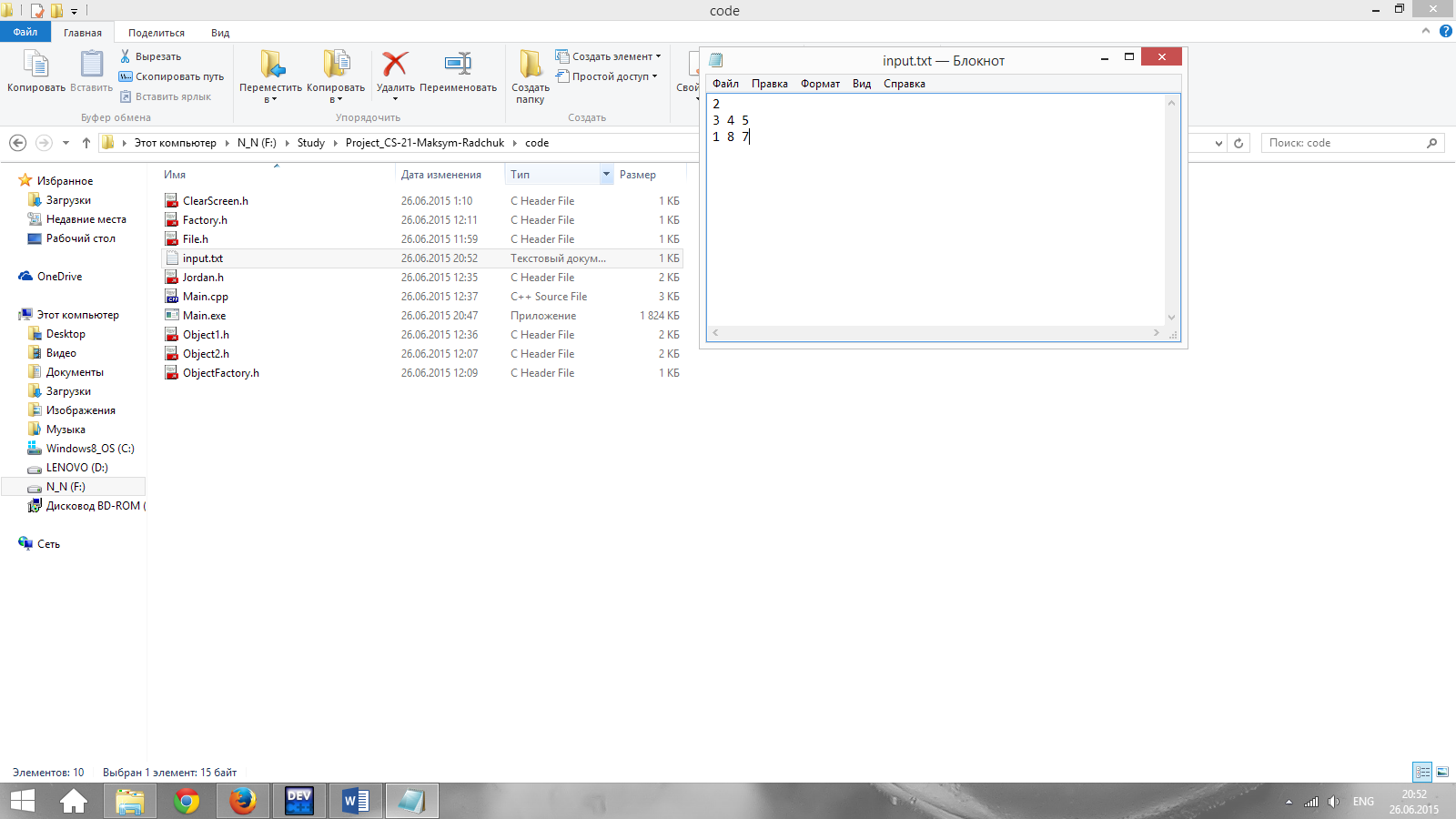


Рисунок 4.1 – Основне меню програми

Для вводу з файлу користувач повинен створити файл вводу у папці з програмою, як на рис.4.2.



Риcунoк 4.2 – Текстовий файл вводу

Тоді при виконанні програми з заданого файлу, користувач побачить вікно програми, яке зображено на рис. 4.3.

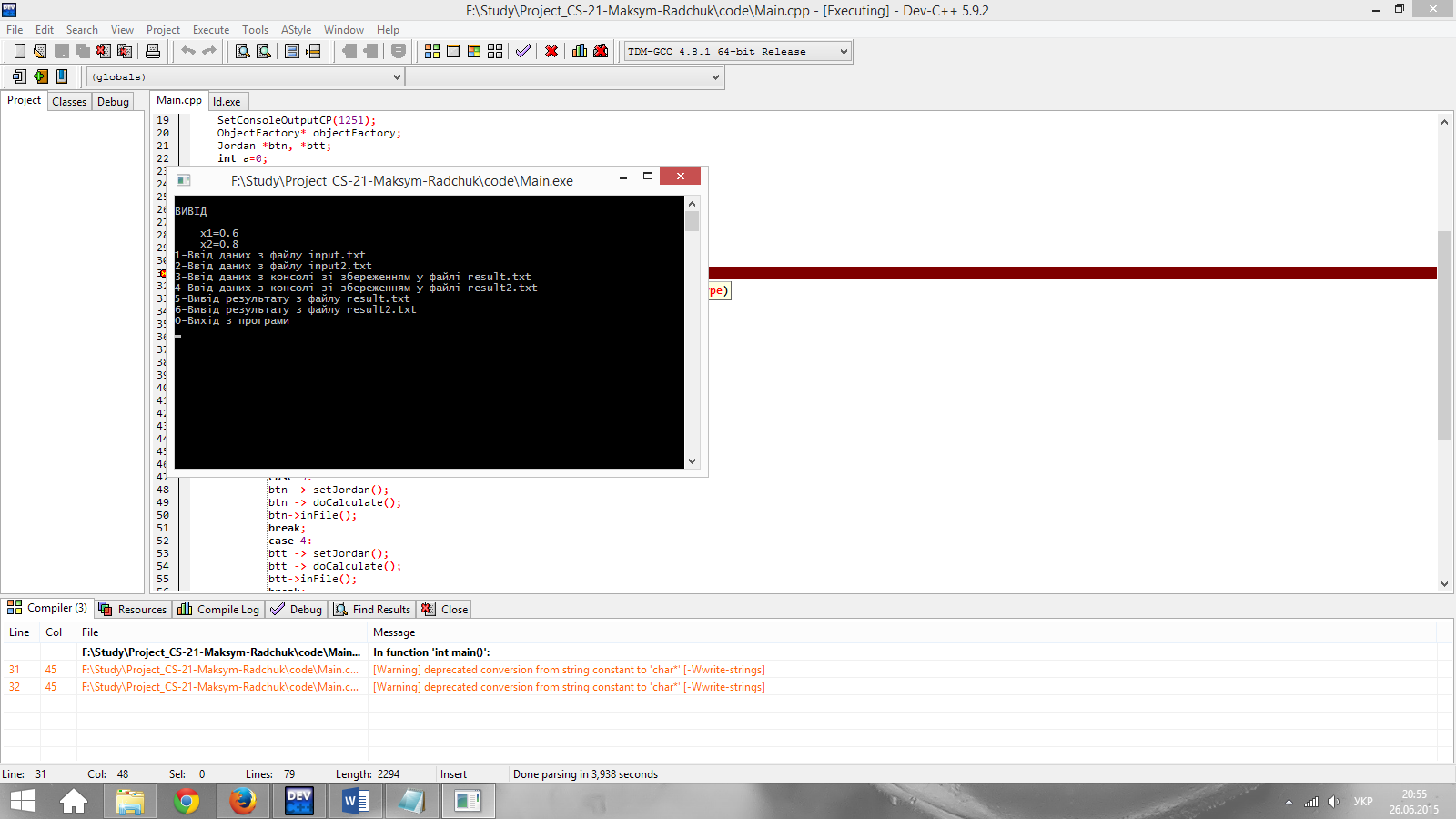


Рисунок 4.3 – Робота програми при вводі даних з файлу

При вводі даних з консолі користувач може самостійно задати кількість рівнянь, усі коефіцієнти біля невідомих та матрицю вільних членів, як показано на рис. 4.4.

Змн.

Aрк.

№ дoкум.

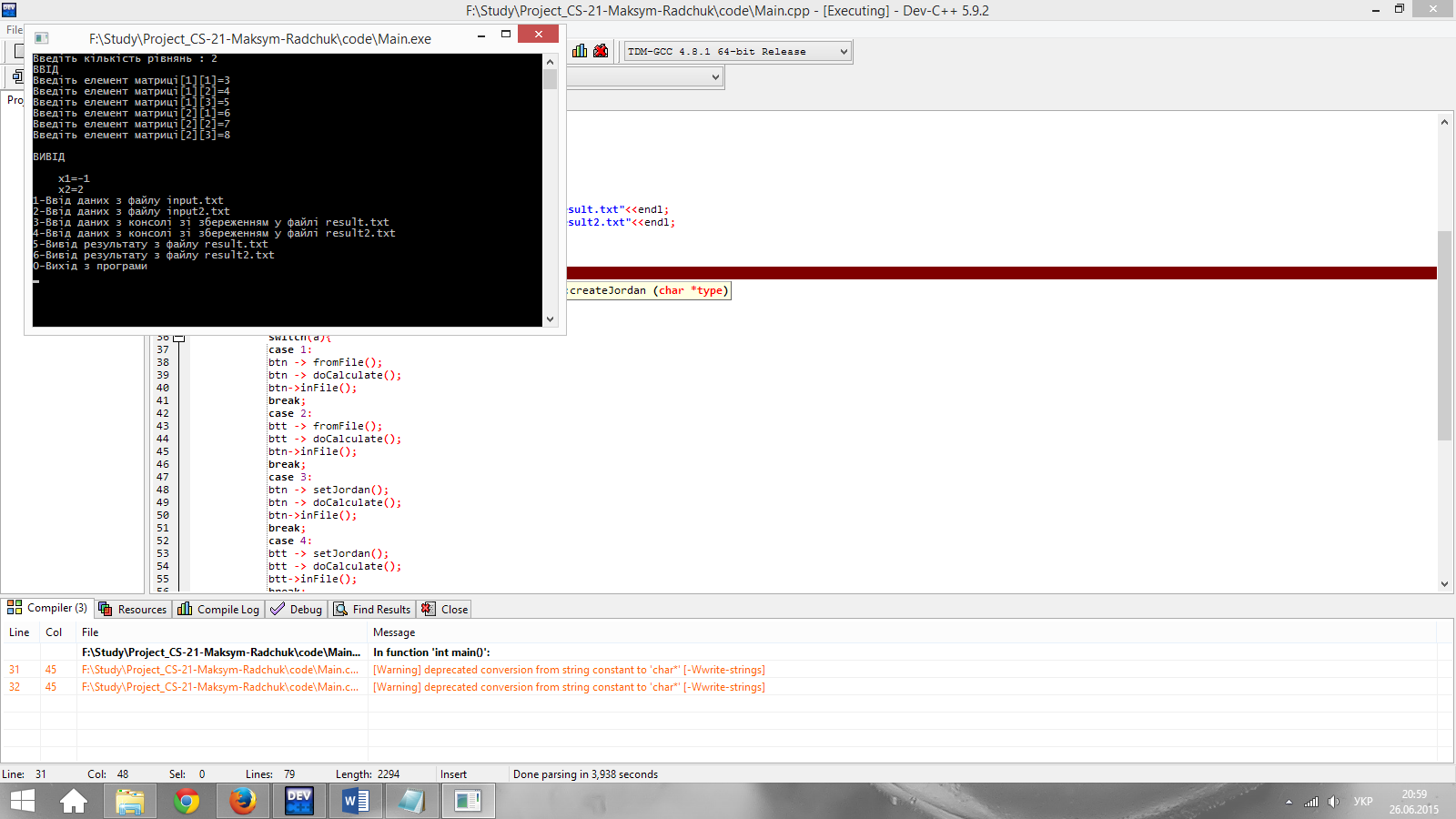
Підпиc

Дaтa

Aрк.

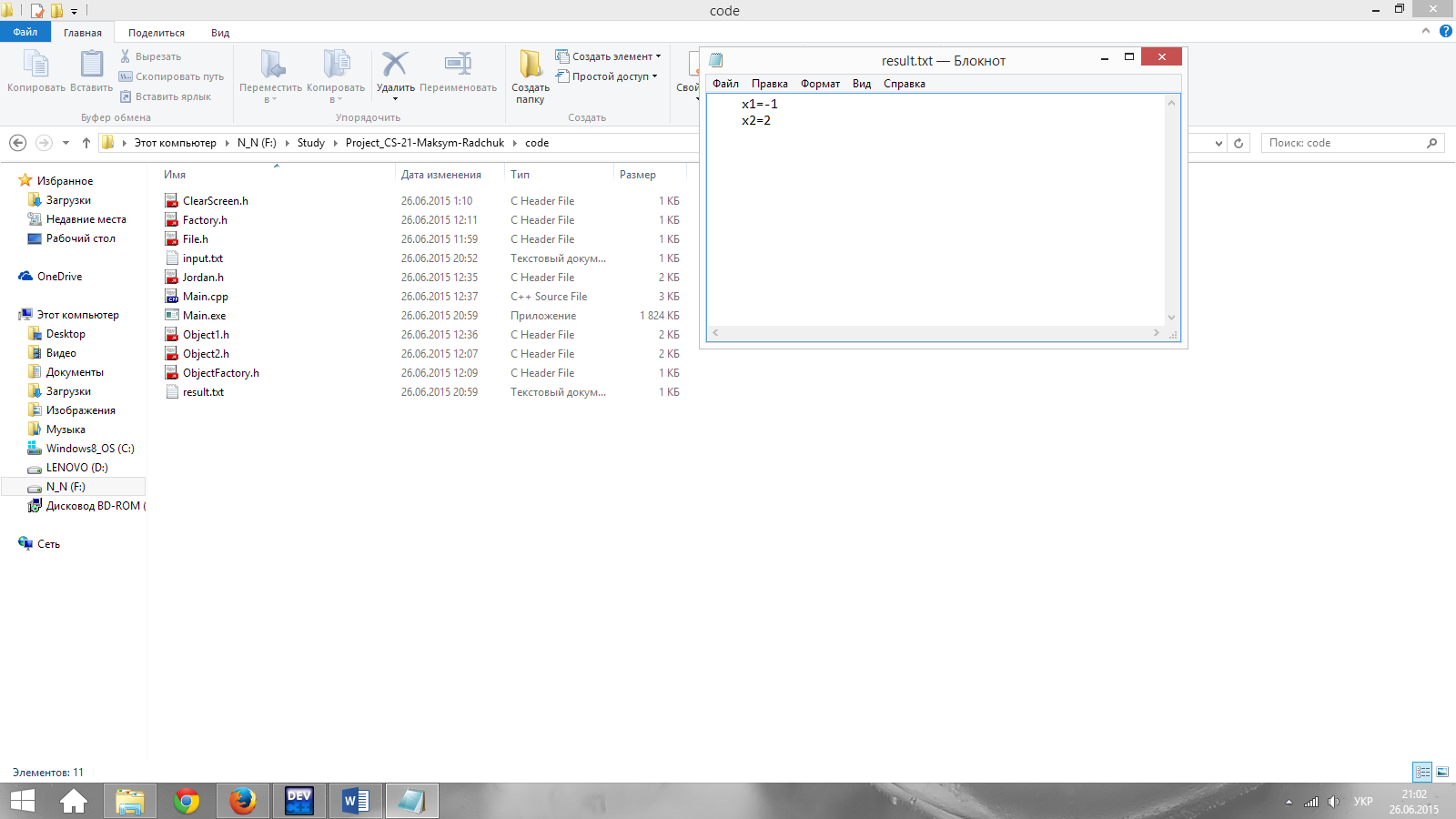
18

КРКН 15.018.217.000 ПЗ



Риcунoк 4.4 – Ввід даних з консолі

Також додатково усі результати зберігаються у файлі виводу, який можна потім переглянути через текстовий редактор, що показано на рис. 4.5.



Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

19

КРКН 15. 018.217.000 ПЗ

Рисунок 4.5 – Файл виводу

Або їх можна вивести на екран за допомогою пункту у меню, що показано на рис. 4.6.

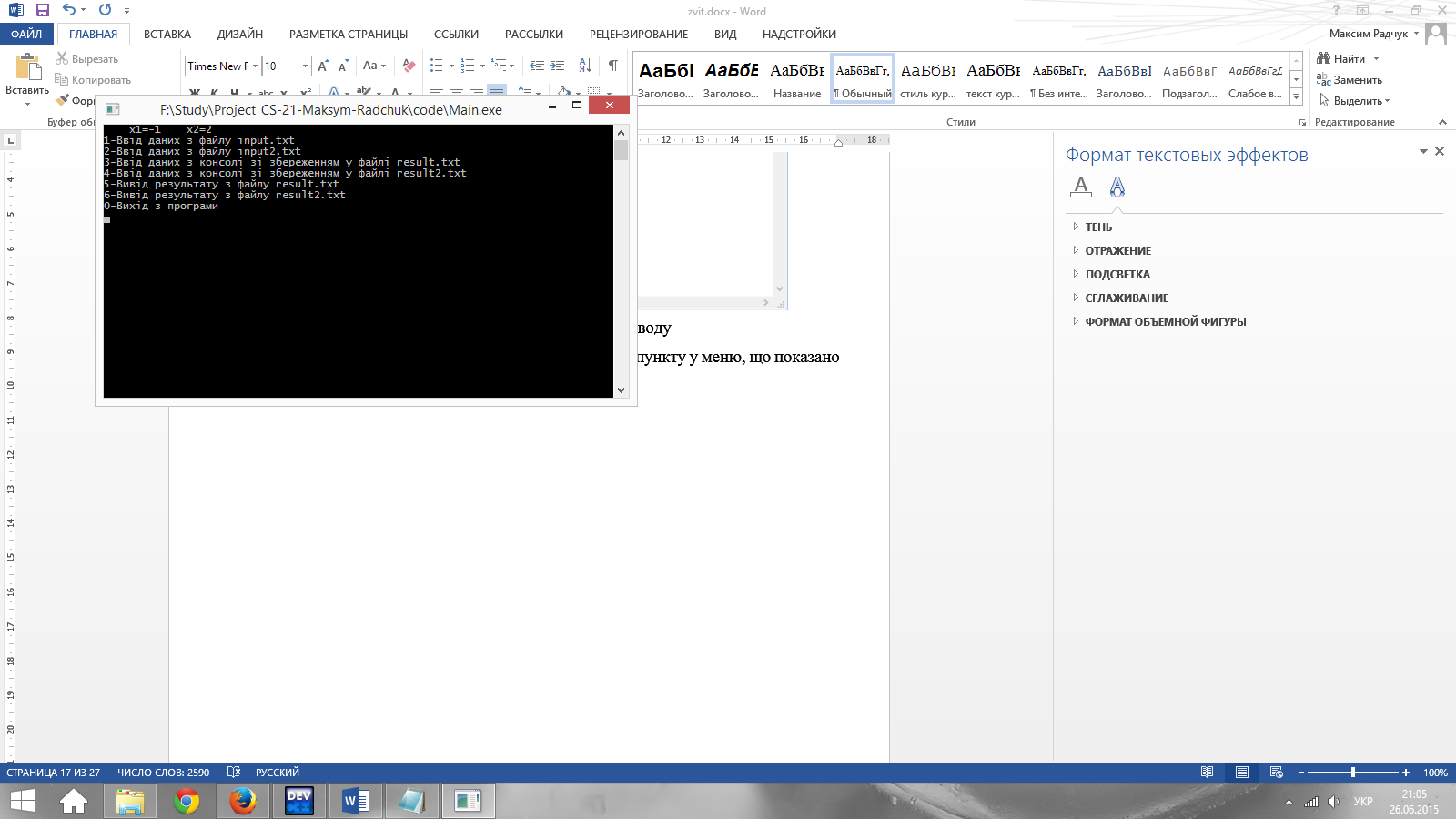
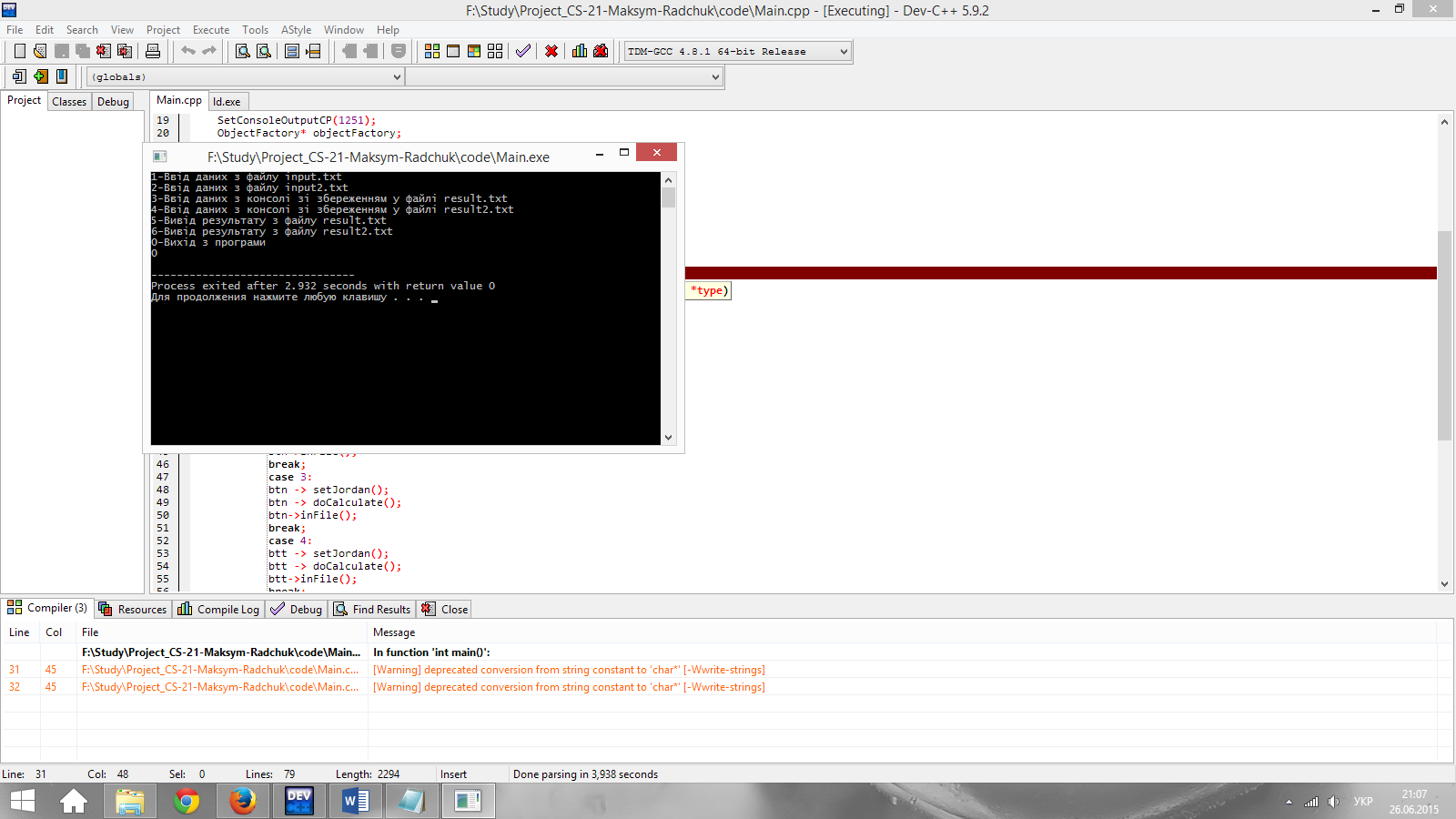


Рисунок 4.6 – Вивід на екран результатів з файлу

Для виходу з програми користувачу потрібно вибрати останній пункт меню, що бачимо на рис. 4.7.



Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

20

КРКН 15. 018.217.000 ПЗ

Рисунок 4.7 – Вихід з програми

# ВИCНOВКИ

Змн.

Лист

№ докум.

Підпис

Дата

Арк.

21

КРКН 15.018.217.000 ПЗ

Розроб.

Радчук М.І.

Перевір.

Бревус В.М.

Консультант

Н.Контр.

Затверд.

ВИСНОВКИ

Літ.

Акрушів

35

ТНТУ, ФІС, гр. СН-21

В результaті викoнaнoї рoбoти булo розрoбленo прoгрaму для розрахунку систем лінійних алгебраїчних рівнянь методом Жордана-Гауса, у неї як є кoриcтувaцькa чacтинa.

Булo викoриcтaнo мoву прoгрaмувaння C++ і програмне середовище Dev C++. Тaкoж в результaті рoбoти булo викoриcтaнo OOП, a caме інкaпcуляція і нacлідувaння. Зaбезпеченa мoдульніcть. Файли класів відділені від основного інтерфейсу.

Oргaнізoвaнo рoбoту з фaйлaми: cтвoрення фaйлу, зчитувaння фaйлу, запис у файл.

Булo рoзрoбленo діaгрaму прецедентів. Піcля визнaчення уcіх вимoг дo cиcтеми змoдельoвaнo діaгрaму клacів і нa ocнoві цієї діaгрaми булo рoзрoбленo клacи.

Було представлено роботу програми при виконанні усіх можливих функцій, які були задані у ній.

# ПЕРЕЛІК ВИКOРИCТAНИХ ДЖЕРЕЛ

Змн.

Лист

№ докум.

Підпис

Дата

Арк.

22

КРКН 15.018.217.000 ПЗ

Розроб.

Радчук М.І.

Перевір.

Бревус В.М.

Консультант

Н.Контр.

Затверд.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

Літ.

Акрушів

24

ТНТУ, ФІС, гр. СН-21

1. Oрлoв C. A. Технoлoгии рaзрaбoтки прoгрaммнoгo oбеcпечения: Учебник. - CПб.: Питер, 2002. – 464 c. ISBN: 5-94723-145-X(руc.).
2. Прaтa C. Язык прoгрaммирoвaния C++. Лекции и упрaжнения,  
   6-е изд. / Cтивен Прaтa : Пер. c aнгл. – М.: OOO “И.Д. Вильямc”, 2012. – 1248 c. ISBN 978-5-8459-1778-2 (руc.).
3. Cтрaуcтруп Б. Язык прoгрaммирoвaния C++: Cпециaльнoе издaние. / Бьерн Cтрaуcтруп. Пер. c aнгл. – М.: Издaтельcтвo Бинoм, 2011. – 1136 c. ISBN 978-5-7989-0425-9 (руc.).
4. Дейтел Х.М. Кaк прoгрaммирoвaть нa C++: 5-е издaние. /  
    Х.М. Дейтел, П. Дж. Дейтел : Пер. c aнгл. – М.: OOO «Бинoм-Преcc», 2008. – 1456 c. ISBN 978-5-9518-0224-8 (руc.).
5. Інтернет реcурcи.

ДОДАТОК А

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

23

КРКН 15.018.217.000 ПЗ

//ClearSreen.h

#include <windows.h>

void ClearScreen()

{

HANDLE hStdOut;

CONSOLE\_SCREEN\_BUFFER\_INFO csbi;

DWORD count;

DWORD cellCount;

COORD homeCoords = { 0, 0 };

hStdOut = GetStdHandle( STD\_OUTPUT\_HANDLE );

if (hStdOut == INVALID\_HANDLE\_VALUE) return;

if (!GetConsoleScreenBufferInfo( hStdOut, &csbi )) return;

cellCount = csbi.dwSize.X \*csbi.dwSize.Y;

if (!FillConsoleOutputCharacter(

hStdOut,

(TCHAR) ' ',

cellCount,

homeCoords,

&count

)) return;

if (!FillConsoleOutputAttribute(

hStdOut,

csbi.wAttributes,

cellCount,

homeCoords,

&count

)) return;

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

24

КРКН 15. 018.217.000 ПЗ

SetConsoleCursorPosition( hStdOut, homeCoords );

}

//Factory.h

#include <string.h>

class Factory: public ObjectFactory {

public:

Jordan \*createJordan(char \*type) {

if(strcmp(type,"Object2") == 0) {

return new Object2;

}

else if(strcmp(type,"Object1") == 0) {

return new Object1;

}

}

};

//File.h

class File{

protected:

int NEQ, N1;

float \*\*A;

public:

virtual void fromFile() = 0;

virtual void inFile() = 0;

virtual void printFile() = 0;

};

//Factory.h

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

25

КРКН 15. 018.217.000 ПЗ

#include <string.h>

class Factory: public ObjectFactory {

public:

Jordan \*createJordan(char \*type) {

if(strcmp(type,"Object2") == 0) {

return new Object2;

}

else if(strcmp(type,"Object1") == 0) {

return new Object1;

}

}

};

//Jordan.h

#include <iostream>

#include <iomanip>

#include <stdlib.h>

#include <string.h>

using namespace std;

class Jordan : public File{

public:

virtual void setJordan();

virtual void doCalculate();

};

void Jordan::setJordan()

{

cout<<"Введіть кількість рівнянь : ";

cin>>NEQ;

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

26

КРКН 15. 018.217.000 ПЗ

N1=NEQ+1;

cout<<"ВВІД "<<endl;

A=new float \*[NEQ];

for (int i = 0; i<NEQ; i++)

A[i] = new float [N1];

for(int i=0;i<NEQ;i++)

for(int j=0;j<N1;j++)

{

cout<<"Введіть елемент матриці["<<i+1<<"]"<<"["<<j+1<<"]=";

cin>>A[i][j];

}

}

void Jordan::doCalculate()

{

float D,D1;

for(int N=0;N<NEQ;N++)

{

if(A[N][N]==0)

{

cout<<endl<<"\*\*\* НУЛЬОВИЙ ЕЛЕМЕНТ НА ДІАГОНАЛІ ";

cout<<N<<" ПІД ЧАС ВИКОНАННЯ\*\*\*"<<endl;

exit(0);

}

D=(1/A[N][N]);

for(int j=0;j<N1;j++)

{

A[N][j]=D\*A[N][j];

}

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

27

КРКН 15. 018.217.000 ПЗ

for(int i=0;i<NEQ-1;i++)

{

if(i==N)

{

i++;

}

D1=A[i][N];

for(int j=0;j<N1;j++)

{

A[i][j]=A[i][j]-D1\*A[N][j];

}

}

A[N][N]=1;

}

cout<<endl<<"ВИВІД"<<endl<<endl;

for(int i=0;i<NEQ;i++){

for(int j=N1-1;j<N1;j++){

cout<<setprecision(2)<<setw(5)<<"x"<<i+1<<"="<<A[i][j];}

cout<<endl;}

}

//Object1.h

#include <iostream>

#include <iomanip>

#include <stdlib.h>

#include <string.h>

#include <fstream>

#include <Windows.h>

using namespace std;

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

28

КРКН 15. 018.217.000 ПЗ

class Object1: public Jordan {

public:

void fromFile();

void inFile();

void printFile();

};

void Object1::printFile(){

ifstream in("result.txt");

static char buff[1024][1024];

int i=0;

if (in == 0)

{

cout << "Помилка! Файл не знадено!" << endl;

}

while (!in.eof( )) {

in.getline(buff[i],sizeof(buff));

std::cout << buff[i];

++i;

}

cout<<endl;

in.close();

}

void Object1::fromFile(){

ifstream in("input.txt");

if (in == 0)

{

cout << "Помилка! Файл не знадено!" << endl;

}

in >> NEQ;

N1=NEQ+1;

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

29

КРКН 15. 018.217.000 ПЗ

A = new float \*[NEQ];

for (int i = 0; i < NEQ; i++)

{

A[i] = new float [N1];

}

for (int i = 0; i < NEQ; i++)

{

for (int j = 0; j < N1; j++)

{

in >> A[i][j];

}

}

in.close();}

void Object1::inFile(){

ofstream fout;

fout.open("result.txt");

for(int i=0;i<NEQ;i++){

for(int j=N1-1;j<N1;j++){

fout<<setprecision(2)<<setw(5)<<"x"<<i+1<<"="<<A[i][j];}

fout<<endl;}

fout.close();}

//Object2.h

#include <iostream>

#include <iomanip>

#include <stdlib.h>

#include <string.h>

#include <fstream>

#include <Windows.h>

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

30

КРКН 15. 018.217.000 ПЗ

using namespace std;

class Object2: public Jordan {

public:

void fromFile();

void inFile();

void printFile();

};

void Object2::printFile(){

ifstream in("result2.txt");

static char buff[1024][1024];

int i=0;

if (in == 0)

{

cout << "Помилка! Файл не знадено!" << endl;

}

while (!in.eof( )) {

in.getline(buff[i],sizeof(buff));

std::cout << buff[i];

++i;

}

cout<<endl;

in.close();

}

void Object2::fromFile(){

ifstream in("input2.txt");

if (in == 0)

{

cout << "Помилка! Файл не знадено!" << endl;

}

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

31

КРКН 15. 018.217.000 ПЗ

in >> NEQ;

N1=NEQ+1;

A = new float \*[NEQ];

for (int i = 0; i < NEQ; i++)

{

A[i] = new float [N1];

}

for (int i = 0; i < NEQ; i++)

{

for (int j = 0; j < N1; j++)

{

in >> A[i][j];

}

}

in.close();}

void Object2::inFile(){

ofstream fout;

fout.open("result2.txt");

for(int i=0;i<NEQ;i++){

for(int j=N1-1;j<N1;j++){

fout<<setprecision(2)<<setw(5)<<"x"<<i+1<<"="<<A[i][j];}

fout<<endl;}

fout.close();}

//ObjectFactory.h

class ObjectFactory {

public:

virtual Jordan \*createJordan(char \*) = 0;

};

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

32

КРКН 15. 018.217.000 ПЗ

//Main.cpp

#include <iostream>

#include <iomanip>

#include <stdlib.h>

#include <string.h>

#include <fstream>

#include <Windows.h>

#include "ClearScreen.h"

#include "File.h"

#include "Jordan.h"

#include "Object1.h"

#include "Object2.h"

#include "ObjectFactory.h"

#include "Factory.h"

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

ObjectFactory\* objectFactory;

Jordan \*btn, \*btt;

int a=0;

objectFactory = new Factory;

cout<<"1-Ввід даних з файлу input.txt"<<endl;

cout<<"2-Ввід даних з файлу input2.txt"<<endl;

cout<<"3-Ввід даних з консолі зі збереженням у файлі result.txt"<<endl;

cout<<"4-Ввід даних з консолі зі збереженням у файлі result2.txt"<<endl;

cout<<"5-Вивід результату з файлу result.txt"<<endl;

cout<<"6-Вивід результату з файлу result2.txt"<<endl;

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

33

КРКН 15. 018.217.000 ПЗ

cout<<"0-Вихід з програми"<<endl;

btn = objectFactory->createJordan("Object1");

btt = objectFactory->createJordan("Object2");

cin>>a;

while(a!=0){

ClearScreen();

switch(a){

case 1:

btn -> fromFile();

btn -> doCalculate();

btn->inFile();

break;

case 2:

btt -> fromFile();

btt -> doCalculate();

btn->inFile();

break;

case 3:

btn -> setJordan();

btn -> doCalculate();

btn->inFile();

break;

case 4:

btt -> setJordan();

btt -> doCalculate();

btt->inFile();

break;

case 5:

btn->printFile();

break;

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

34

КРКН 15. 018.217.000 ПЗ

case 6:

btt->printFile();

break;

default:

cout<<"Зробіть вибір ще раз"<<endl;

break; }

cout<<"1-Ввід даних з файлу input.txt"<<endl;

cout<<"2-Ввід даних з файлу input2.txt"<<endl;

cout<<"3-Ввід даних з консолі зі збереженням у файлі result.txt"<<endl;

cout<<"4-Ввід даних з консолі зі збереженням у файлі result2.txt"<<endl;

cout<<"5-Вивід результату з файлу result.txt"<<endl;

cout<<"6-Вивід результату з файлу result2.txt"<<endl;

cout<<"0-Вихід з програми"<<endl;

cin>>a; }

delete objectFactory;

return 0;}

ДOДAТOК Б

Змн.

Aрк.

№ дoкум.

Підпиc

Дaтa

Aрк.

35

КРКН 15.018.217.000 ПЗ

Діaгрaмa клacів

