

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ І ЗВ'ЯЗКУ

Кафедра інженерії програмного забезпечення

ОБРОБКА СТРУКТУРОВАНИХ ДАНИХ ЗА ДОПОМОГОЮ МОВИ C++

з дисципліни
«Технології програмування»

Методичні вказівки
з виконання курсової роботи
для студентів бакалаврської підготовки
зі спеціальності 125 - Кібербезпека

Одеса 2023

Рецензент: д.т.н., професор Ісмаїлова Н.П.

“Обробка структурованих даних за допомогою мови С++” з дисципліни
“Технології програмування” : методичні вказівки з виконання курсової роботи /
Укладачі: Л.В. Глазунова, Л.М. Буката. Одеса:ДУІТЗ, 2023, с. 34.

Метою курсової роботи є набуття практичних навичок розробки алгоритму та програмування мовою С++ власної задачі з використанням таких елементів програмування, як складені структури даних, тип даних дата-час, обробка помилок за допомогою виключень, робота з файлами, а також застосування різних алгоритмів сортування.

Курсове завдання містить 50 варіантів. Студент за вказівкою викладача отримує власний варіант.

Методичні вказівки з виконання курсової роботи для студентів бакалаврської підготовки спеціальності 125 Кібербезпека.

СХВАЛЕНО

на засіданні кафедри
інженерія програмного забезпечення
та рекомендовано до друку.
Протокол № 6
від 27 січня 2023 р.

ЗАТВЕРДЖЕНО

методичною радою
ДУІТЗ.
Протокол № 2
від 08 лютого 2023 р.

Вступ

Дисципліна “Технології програмування” викладається у семестрах 2.1 і 2.2 студентам другого курсу, які навчаються за спеціальністю “Кібербезпека”.

Метою викладання навчальної дисципліни “Технології програмування” є навчання студентів основам програмування процедурними та об’єктно-орієнтованими мовами високого рівня C, C++, C# та Python, здобуття практичних навичок складання та налагодження програм за допомогою комп’ютера, основам алгоритмізації і програмування алгоритмів обробки структурованих даних, алгоритмів обчислювальної математики для застосування здобутих навичок у процесі навчання, повсякденній і майбутній професійній діяльності, пов’язаній з захистом інформації.

Основними завданнями вивчення дисципліни “Технології програмування” є вивчення сутності, основ процедурного та об’єктно-орієнтованого програмування на прикладі найбільш поширених мов програмування, вміння проектувати алгоритми і структури даних. У результаті вивчення навчальної дисципліни студент повинен отримати наступні компетенції відповідно до Стандарту вищої освіти за спеціальністю 125 Кібербезпека галузі знань 12 Інформаційні технології для першого (бакалаврського) рівня вищої освіти, затвердженого Наказом МОН №1382 від 12.12.2018 р.:

Загальні компетентності

КЗ 5. Здатність до пошуку, оброблення та аналізу інформації.

Спеціальні (фахові, предметні) компетентності

КФ 2. Здатність до використання інформаційно-комунікаційних технологій, сучасних методів і моделей інформаційної безпеки та/або кібербезпеки.

КФ 3. Здатність до використання програмних та програмно-апаратних комплексів засобів захисту інформації в інформаційно-телекомунікаційних (автоматизованих) системах.

Програмні результати навчання

ПРН-3 – використовувати результати самостійного пошуку, аналізу та синтезу інформації з різних джерел для ефективного рішення спеціалізованих задач професійної діяльності;

ПРН-15 – використовувати сучасне програмно-апаратне забезпечення інформаційно-комунікаційних технологій;

ПРН-18 – використовувати програмні та програмно-апаратні комплекси захисту

інформаційних ресурсів;

ПРН-20 – забезпечувати функціонування спеціального програмного забезпечення, щодо захисту інформації від руйнуючих програмних впливів, руйнуючих кодів в інформаційно-телекомунікаційних системах.

Програма курсу складається з трьох модулів:

Модуль 1 - Динамічні структури даних.

Модуль 2 - Алгоритми програмування поширених задач.

Модуль 3 - Скриптова мова Python.

Тема курсової роботи з дисципліни «Технології програмування» - «ОБРОБКА СТРУКТУРОВАНИХ ДАНИХ ЗА ДОПОМОГОЮ МОВИ C++».

Мета курсової роботи: здобуття практичних навичок проектування та програмування інженерних задач за допомогою мови C++ з використанням складених структур та алгоритмів сортування в середовищі VISUAL STUDIO.

У результаті виконання курсової роботи студент повинен вміти проектувати програмну систему, застосовувати сучасні середовища розробки програм, кодувати на одній з високорівневих мов програмування, документувати та тестувати розроблену програмну систему.

Завдання на курсовий проект видається на початку 3 семестру. На його виконання відводиться 20 годин індивідуальної роботи студента. Після виконання та оформлення КР студент захищає її за 100 бальною системою. При цьому береться до уваги компетентність студента, оригінальність та творчість мислення, обґрунтованість прийнятих рішень, ритмічність у роботі (дотримання термінів здачі роботи або її складових частин).

СТРУКТУРА КУРСОВОЇ РОБОТИ

ЗМІСТ

1 ВСТУП

2 ТЕОРЕТИЧНА ЧАСТИНА

- 2.1 Складовий тип даних – структура.
- 2.2 Робота з файлами у C/C++: текстові та бінарні.
- 2.3 Робота з датою/часом у C++.
- 2.4 Алгоритми сортування: вставки, вибору, бульбашковий.
- 2.5 Обробка виключень у C++.

3. КУРСОВЕ ЗАВДАННЯ

4. ПРОЕКТУВАННЯ ПРОГРАМИ

- 4.1 Опис основних структурних елементів (функцій) програми.
- 4.2 Блок-схема алгоритму сортування.
- 4.3 Блок-схема алгоритму вибору за умовою.

5. РОЗРОБКА ПРОГРАМНОГО КОДУ

6. ТЕСТУВАННЯ ПРОГРАМИ

ВИСНОВКИ

СПИСОК ЛІТЕРАТУРИ

ВИМОГИ ЩОДО ОФОРМЛЕННЯ КУРСОВОЇ РОБОТИ

Правила оформлення сформульовані згідно з державним стандартом. Орієнтовний обсяг КР повинен складати 20-30 сторінок формату А4 (210x297 мм). Сторінки повинні бути заповнені хоча б на 50%. Розміри полів сторінок: верхнє і нижнє – 20 мм, праве – 15 мм, ліве – 25 мм. Основний текст повинен бути надрукований без переносів шрифтом Times New Roman Cyr розміром 14 пунктів, вирівнювання – по ширині сторінки, міжрядковий інтервал – 1,25. Фрагменти вихідного коду програм надаються шрифтом Courier New розміром 12 пунктів, вирівнювання – по лівому краю, міжрядковий інтервал – 1. Абзацний відступ повинен бути однаковим впродовж усього тексту і приблизно дорівнювати п'яти знакам (12,5 мм).

Сторінки слід нумерувати арабськими цифрами, дотримуючись наскрізної нумерації впродовж усього тексту. Номер проставляють праворуч у верхньому куті сторінки без крапки в кінці, починаючи з листа, наступного за першою сторінкою змісту. У загальну нумерацію включають всі сторінки, починаючи з титульної.

Розділи і підрозділи, пункти і підпункти основної частини повинні мати заголовки. Заголовки СТРУКТУРНИХ ЕЛЕМЕНТІВ та РОЗДІЛІВ слід розташовувати у середині рядка нової сторінки, друкувати великими літерами напівжирним шрифтом без крапки в кінці, не підкреслюючи. Заголовки підрозділів, пунктів і підпунктів слід починати з абзацного відступу і друкувати напівжирним шрифтом малими літерами, крім першої великої, не підкреслюючи, без крапки в кінці. Відстань між заголовком і подальшим чи попереднім текстом має дорівнювати одному рядку. Розділи і підрозділи слід нумерувати арабськими цифрами. Номер підрозділу складається з номера розділу і порядкового номера підрозділу, розділених крапкою. Після номера крапку не ставлять.

Ілюстрації (схеми, діаграми, знімки екранних форм) слід розташовувати безпосередньо після тексту, в якому вони згадуються вперше, або на наступній сторінці. На всі ілюстрації мають бути посилання в тексті. Ілюстрації розміщуються з вирівнюванням по центру і повинні мати назву, яку розміщують під ілюстрацією через один рядок після неї також з вирівнюванням по центру. Відстань між текстом і ілюстрацією становить один рядок. Ілюстрації слід нумерувати арабськими цифрами порядковою нумерацією в межах розділу.

Номер ілюстрації складається з номера розділу і порядкового номера ілюстрації, розділених крапкою. Номер ілюстрації від її назви відділяється тире, крапка в кінці назви не ставиться. Наприклад, другий рисунок третього розділу слід зазначити наступним чином: Рисунок 3.2 – Структурна схема алгоритму методу

На всі рисунки повинні бути посилання в тексті. Посилання на таблиці в тексті Times New Roman скорочують і пишуть з малої літери, наприклад, див. рис. 3.2 або, за неявного посилання, (рис. 3.2).

Таблицю слід розташовувати безпосередньо після тексту, в якому вона згадується вперше, або на наступній сторінці. Таблиця має назву, яку друкують малими літерами (крім першої великої) і вміщують над таблицею з вирівнюванням вліво. Нумеруються таблиці арабськими цифрами порядковою нумерацією в межах розділу, за винятком таблиць, що надані в додатках. Номер таблиці складається з номера розділу і порядкового номера таблиці, між якими ставиться крапка. На всі таблиці повинні бути посилання в тексті. Посилання на таблиці в тексті Times New Roman скорочують і пишуть з малої літери, наприклад, див. табл. 4.2 або, при неявному посиланні, (табл. 4.2).

Тексти програм (лістинги) слід розташовувати безпосередньо після тексту, в якому вони згадуються вперше, або на наступній сторінці. Відстань між лістингом і текстом становить один рядок. Код лістинга надається шрифтом Courier New 12 з міжстроковим інтервалом 1.0. Лістинг слід нумерувати арабськими цифрами порядковою нумерацією в межах розділу. Номер лістинга складається з номера розділу і порядкового номера лістинга, розділених крапкою. Номер лістинга від його назви відділяється тире, крапка в кінці назви не ставиться. Наприклад, перший лістинг третього розділу слід зазначити наступним чином:

Лістинг 3.1 – SQL-запит на виведення даних.

На всі лістинги повинні бути дані посилання в тексті, наприклад, див. Лістинг 4.2 або, при неявному посиланні, (лістинг 4.2).

Перед переліком ставлять двокрапку. Перед кожною позицією переліку слід ставити малу літеру або цифру з дужкою або, не нумеруючи, коротке тире (перший рівень деталізації). Для подальшої деталізації переліку слід використовувати літери з дужкою (другий рівень деталізації). Переліки першого рівня деталізації друкують малими літерами з абзацного відступу, другого рівня – з відступом щодо місця розташування переліків першого рівня. наприклад:

1) інтегроване середовище для розробки програмного забезпечення Microsoft Visual Studio;

2) засоби баз даних:

а) система керування базами даних з відкритим вихідним кодом PostgreSQL;

б) інструмент для корпоративного моделювання PowerDesigner;

3) компілятори.

В усіх розділах роботи необхідно надавати посилання на літературні джерела, з яких запозичуються матеріали. Не допускається використання тексту інших авторів без посилань на них, а також пряме цитування без використання лапок.

Список використаних джерел надається в кінці пояснювальної записки, починаючи з нової сторінки. Він включає описи тільки тих джерел, які були використані при виконанні роботи і на які є посилання в тексті. Джерела нумеруються і друкуються з абзацного відступу. Відомості про джерела слід надавати і розташовувати у порядку появи посилань, на них у тексті. Неприпустимо вказувати джерела, на які в тексті немає жодного посилання. У тексті записки посилання на використані літературні джерела слід позначати порядковим номером за переліком посилань, виділеним двома квадратними дужками, наприклад: «... в роботі [2] ...» або «... в роботах [1-7,10] ...». Вимоги до оформлення бібліографічних описів сформульовані у відповідності з державним стандартом [9].

Приклади оформлення бібліографічних посилань:

1. Дацюк В.М., Букатов А.А., Жигуленко А.І. Багатопроцесорні системи та паралельне програмування: мет. посіб. Частина 1,2,3.

URL: <http://rsusu1.rnd.runnet.ru/ncube/koi8/method/m1/context.html>.

2. Таненбаум Е. Архітектура комп'ютера. 4-е вид. Х.: Харків, 2003. 697 с.

ЗАВДАННЯ ДО КУРСОВОЇ РОБОТИ

1 Створити бінарний файл для зберігання записів структури за власним варіантом (таблиця 1).

Примітка. Ввести не менше 10 записів, вивести на екран вміст файла. Зробити скріншот вмісту файла у вікні виконання програми.

2 Відсортувати вказане поле у файлі за допомогою заданого алгоритму.

Примітка. Визначити кількість записів у файлі, оголосити і заповнити динамічний масив, вивести його, впорядкувати цей масив, вивести знову, повернути дані у файл, вивести вміст файла. Зробити скріншоти вмісту масиву до і після сортування, вмісту файла після сортування.

3 Вивести дані за умовою.

Примітка. Створити скріншот даних за умовою.

4 Створити текстовий файл з інформацією за умовою.

Примітка. Вивести на екран вміст файла. Зробити скріншот вмісту файла у вікні виконання програми і в Блокноті.

5 При програмуванні завдання використовувати обробку винятків.

Примітка. Зробити скріншот повідомлення у випадку спрацювання виключення.

Важливо. Програма розробляється, як набір функцій, що вирішують завдання курсової роботи, як це показано нижче у прикладі розробки коду.

Таблиця 1 – Варіанти індивідуальних завдань

№ вар.	Вміст бінарного файла	Параметри сортування (алгоритм сортування)	Відбір даних за умовою
1	«Бібліотека»: прізвище читача; назва книги, взятої читачем; дата видачі книги; термін, на який видано книгу; реальна дата здачі читачем книги	За зростанням дати видачі (вставки)	Роздрукувати відомості про читачів старше 60 років, які не повернули книгу в зазначений термін
2	«Молочна продукція»: назва молочного продукту; дата надходження товару до магазину; термін зберігання продукту за накладною; дата перевірки наявності продукту у магазині.	За назвою продукції (бульбашковий)	Визначити продукцію з назвою «молоко» та «сир», що зберігаються у магазині більше відведеного терміну
3	«Кінофільми»: назва фільму; країна-виробник; дата виходу на екран; кількість серій; тривалість однієї серії; якими днями проходить серіал.	За зменшенням кількості серій (вставки)	Визначити, скільки днів буде демонструватися самий багатосерійний фільмвиробником якого є Франція. Вивести дані про цей серіал
4	«Ліки від хвороби»: назва хвороби; назва ліків; країна-виробник; форма (таблетки, сироп, краплі ...); вартість ліків; термін зберігання (місяць, рік)	За назвою хвороби (вибору)	Роздрукувати відомості про ліки, здатні лікувати грип або ГРЗ, термін зберігання яких закінчиться через місяць
5	Список робітників підприємства: табельний номер, ПІБ, посада, дата народження, наявність вищої освіти (так/ні), розмір зарплатні	За зростанням розміру зарплатні (бульбашковий)	Кількість співробітників без вищої освіти, вік яких до 40 років
6	Дані в ЖЕКу про мешканців: адреса, назва ділянки чи району, ПІБ власника, дата укладення договору про надання послуг, наявність субсидії (так/ні), розмір заборгованості	За спаданням розміру заборгованості (вибору)	Кількість боржників без субсидії, сума боргу яких понад 500 грн.
7	Список товарів на складі: код, назва, категорія, кількість, ціна за одиницю товару, дата завезення на склад, ознака того чи є товар що швидко псується (так/ні)	В алфавітному порядку назв товарів (вставки)	Товари, що швидко псуються та завезені понад місяць тому
8	Товари в магазині електроніки: код, назва, фірма-виробник, рік випуску, вартість, наявність гарантії (так/ні)	За спаданням вартості (бульбашковий)	Дані про телевізори фірми Samsung з наявною гарантією

№ вар.	Вміст бінарного файла	Параметри сортування (алгоритм сортировки)	Відбір даних за умовою
9	Список студентів: номер, ПІБ, група, дата народження, рейтинг з історії, математики, основ програмування, наявність пропусків занять без поважної причини (так/ні)	В алфавітному порядку прізвищ (вибору)	Студенти, які претендують на стипендію: мають середній бал понад 75 балів та не мають пропусків занять без поважної причини (так/ні)
10	Прокат автомобілів: державний номер, марка, кілометраж пробігу, дата здачі у прокат, сума платні, наявність сигналізації (так/ні)	За спаданням кілометражу (вставки)	Автомобілі Ford без сигналізації
11	«Область»: назва області; обласний центр; населення; площа території; губернатор	За назвою (вставки)	Вивести дані про області, кількість населення, яких менше 2 млн. жителів, а площа більше середньоарифметичної
12	"Мотоцикл": марка; колір; серійний номер; реєстраційний номер; рік випуску; рік техогляду; ціна	За зменшенням року випуску (вибору)	Вивести дані про мотоцикли марки Harley-Davidson, які пройшли техогляд у цьому році
13	«Манекенщиця»: прізвище; ім'я; по батькові; стать; національність; зріст; вага; дата народження (рік, місяць число); домашня адреса (країна, місто)	За зростанням віку (вибору)	Вивести дані про манекенниць з України, яким менше 19 та мають день народження у квітні.
14	«Вчений»: прізвище; ім'я; по батькові; стать; національність; дата народження (рік, місяць число); вчений ступінь, посада, номер телефону; домашню адресу (країна, місто).	За країною мешкання (бульбашковий)	Вивести інформацію про вчених кандидатів технічних наук віком до 35
15	Список книг у бібліотеці: інвентарний номер, назва книги, автор, жанр, рік видання, ціна, дата інвентаризації, наявність CD/DVD для книги (так/ні)	За зростанням року видання (вибору)	Книги, у назві яких є "C++", які в комплекті мають CD чи DVD
16	Прайс-лист ноутбуків у магазині: фірма-виробник, тип: ноутбук, ультрабук, нетбук, трансформер тощо), діагональ дисплея у дюймах, ціна, дата поставки, наявність сенсорного дисплея (так/ні)	За зростанням ціни (вставки)	Ноутбуки типу ультрабук з сенсорним дисплеєм

Продовження таблиці 1

№ вар.	Вміст бінарного файла	Параметри сортування (алгоритм сортировки)	Відбір даних за умовою
17	Список команди "Чорноморець": номер гравця, ПІБ, дата народження, амплуа: воротар, захисник тощо), кількість зіграних ігор за минулий сезон, наявність важких травм у цьому сезоні (так/ні)	За спаданням кількості зіграних ігор за минулий сезон (вставки)	Дані про нападників команди, які не мали важких травм у цьому сезоні
18	Список принтерів для продажу: тип принтера, фірма-виробник, марка, ціна принтера, дата поставлення, наявність у комплекті запасного картриджа (так/ні)	За зростанням ціни (бульбашковий)	Принтери фірми HP з ціною до 1000 грн., які мають запасний картридж
19	Дані про наявність квитків на автостанції: пункт призначення, модель автобусу, дата відправлення, час відправлення, ціна, наявність вільних місць (так/ні)	В алфавітному порядку назв пунктів призначення (вибору)	Дані про рейси до Києва, які відправляються у першій половині дня
20	Каталог ПК: фірма-виробник, тип : настільний, моноблок, неттоп, ігровий), ємність RAM, ємність диска, наявність ТВ-тюнера (так/ні)	За зростанням ємності диска (бульбашковий)	Настільні ПК з ТВ-тюнером
21	Заявки на ремонт комп'ютерної техніки: найменування комп'ютера, назва ремонту, ПІБ власника, дата взяття на ремонт, кількість днів для ремонтних робіт, вартість ремонту, наявність передплати (так/ні)	За спаданням вартості (вставки)	Передплачені заявки, для яких кількість днів ремонту є меншими семи
22	Дані про успішність студентів: прізвище та ініціали, дата народження, назва групи, оцінки з фізики, математики, історії, наявність стипендії у минулому семестрі (так/ні)	За зростанням оцінок з математики (вибору)	Студентів, які не здали сесію – мають двійку хоча б з одного предмета – та мали стипендію у минулому семестрі
23	Абоненти телефонної станції: ПІБ абонента, номер телефону, дата встановлення телефону, розмір фіксованої абонплати в місяць, наявність пільг (так/ні), сума заборгованості	За спаданням заборгованості (бульбашковий)	Абонентів без пільг і боргом понад 200 грн.

№ вар.	Вміст бінарного файла	Параметри сортування (алгоритм сортування)	Відбір даних за умовою
24	«Міжнародна компанія»: назва; інтернет сайт; адреса головного офісу (поштовий індекс, країна, місто, вулиця, дім, квартира), час існування на світовому ринку; кількість співробітників; кількість філіалів у Європі.	За зростанням кількості філіалів (вставки)	Вивести міжнародні компанії за кількістю співробітників, у яких понад 10000, і головний офіс знаходиться у Німеччині
25	"Зоопарк": назва тварини; вага денної норми; кількість виду; адреса зоопарку (поштовий індекс, країна, місто, вулиця); загальна кількість тварин, кількість працівників	За зменшенням кількості тварин (вибору)	Вивести інформацію про зоопарки в яких є тигри в кількості більше 5
26	«Мультфільм»: назва; режисер (прізвище; ім'я); рік виходу; країна; вартість; дохід; прибуток. Вивести дані про мультфільми компанії Walt Disney	За зростанням прибутку (бульбашковий)	Вивести дані про мультфільми компанії Walt Disney з бюджетом понад \$10000000
27	Список послуг операторів мобільного телефонного зв'язку: номер телефону, назва тарифу, дата ініціалізації пакета, наявність підключення до Інтернету (так/ні), баланс рахунку (грн.)	За спаданням номерів телефонів (вставки)	Дані про абонентів без підключення до Інтернету і балансом рахунку понад 100 грн.
28	Список аварійних будинків у районі міста: вулиця, номер будинку, кількість мешканців у будинку, рік побудови, наявність проведення ремонту за останні 20 років (так/ні)	За зростанням року побудови на облік (бульбашковий)	Вивести відомості про будинки без ремонту за останні 20 років, побудовані до 1941 року, в яких понад 50 мешканців
29	Прайс-лист планшетів у магазині: фірма-виробник, модель, діагональ екрана в дюймах, ціна, дата поставлення, наявність 3G-модуля (так/ні)	За спаданням значень розміру діагоналі екрана (вибору)	Планшети Apple з 3G-модулем
30	Список підприємств, які виготовляють електронне обладнання: назва підприємства, дата реєстрації підприємства, вид продукції, кількість продукції за останній квартал, рентабельність (так/ні)	За зростанням кількості продукції за останній квартал (вставки)	Рентабельні підприємства, які виробляють TV-тюнери

№ вар.	Вміст бінарного файла	Параметри сортування (алгоритм сортування)	Відбір даних за умовою
31	Характеристики комп'ютерів: назва, частота, кількість ядер, об'єм оперативної пам'яті, наявність, зовнішньої відеокарти (так/ні), вартість	За спаданням вартості (бульбашковий)	Комп'ютери більше 4-х ядер та зовнішньою відеокартою
32	Прайс-лист магазину мобільних телефонів: фірма-виробник, модель, діагональ екрана в дюймах, ціна, дата поставлення, наявність двох SIM-карт (так/ні)	За спаданням ціни (вставки)	Телефони Samsung з двома SIM-картами
33	Послуги турагенції: назва туру, країна, дата виїзду, кількість днів, вартість туру, наявність нічних переїздів (так/ні)	За зростанням вартості турів (бульбашковий)	Тури до Польщі без нічних переїздів
34	Список комп'ютерів у комп'ютерному класі: номер, тип ПК, тактова частота, оперативна пам'ять, ємність диска, дата встановлення, наявність підключення до мережі Інтернет (так/ні)	За спаданням значень тактової частоти (вибору)	Підключені до Інтернету ПК, в яких ємність диска не менше 80 Гб
35	Список студентів курсу: ПІБ, дата народження, група, середній бал успішності, проживання у гуртожитку (так/ні)	За зростанням середнього бала (вставки)	Студентів із середнім балом понад 80, які мешкають у гуртожитку
36	Співробітники підприємства: табельний номер, ПІБ, дата народження, посада, відділ, оклад, наявність премії минулого місяця	За спаданням табельних номерів (бульбашковий)	Робітники з відділу реалізації, які отримали премію
37	Список моніторів: модель, фірма-виробник, діагональ у дюймах, можливість підтримки 3D (так/ні), дата поставлення	За зростанням розміру діагоналі (вибору)	За зростанням розміру діагоналі Монітори фірми LG без 3D
38	Список робітників: табельний номер, прізвище та ініціали, дата народження, посада, оклад, наявність нагород (так/ні)	За спаданням окладу (бульбашковий)	Робітники пенсійного віку (понад 60 років), та, які мають нагороди
39	"Рейс": марка автомобіля; номер автомобіля; пункт призначення; вантажопідйомність (в тонах); вартість одиниці вантажу; загальна вартість вантажу	За зменшенням вартості одиниці вантажу (вставки)	Вивести дані про автомобіль, вантажопідйомність якого перевищує 2 тони з мінімальною загальною вартістю вантажу

№ вар.	Вміст бінарного файла	Параметри сортування (алгоритм сортування)	Відбір даних за умовою
40	«Кінострічка»: назва; режисер (прізвище; ім'я); рік виходу; країна; вартість; дохід; прибуток	За зростанням доходу (вибору)	Вивести дані про українські фільми з вартістю понад 1000000 з роком виходу до 2000
41	«Держави Європи»: назва країни; столиця; кількість державних мов; населення; площа території; грошова одиниця; державний лад; Глава держави	За зменшенням площі (вставки)	Вивести дані про держави, населення яких понад 10 млн. жителів, які мають декілька державних мов
42	«Власник автомобіля»: прізвище; ім'я; по батькові; номер телефону; домашня адреса (поштовий індекс, країна, область, район, місто, вулиця, будинок, квартира); марка автомобіля; номер автомобіля; номер техпаспорта	За прізвищем (бульбашковий)	Вивести дані про власників автомобіля марки "Ваз", які проживають в Одесі
43	"Пацієнт": ПІБ; стать; зріст; вага; дата народження (рік, місяць число); номер телефону; домашня адреса (місто, вулиця, будинок, квартира); номер лікарні; відділення; номер медичної картки; діагноз; група крові	За ПІБ (вибору)	Вивести дані про пацієнтів старше 60 років, які мають діагноз «гіпертонія» із 18 відділення
44	«Покупець»: прізвище; ім'я; по батькові; стать; дата народження (рік, місяць число); номер телефону; домашня адрес область, район, місто, вулиця, будинок, квартира); номер кредитної картки; банківського рахунка	За зменшенням номера кредитної картки (вставки)	Вивести дані про покупців із міста Одеса, які народились у липні
45	«Виконавець джазової музики»: виконавець; назва альбому; тираж; рік випуску альбому; вартість альбому.	За зростанням тиражу (бульбашковий)	Вивести відомості про виконавця, який у поточному році випустив альбом тиражем понад 1000 примірників за ціною не менше 50 грн.
46	«Озеро»: найменування озера; країна розташування озера; глибина озера; солоність озера у відсотках	За назвою країни розташування (вибору)	Вивести відомості про озера, глибина яких менше 50 м, а солоність понад 20%

Закінчення таблиці 1

№ вар.	Вміст бінарного файла	Параметри сортування (алгоритм сортування)	Відбір даних за умовою
47	Список автомобілів в автосалоні: марка, модель, рік випуску, дата реєстрації, пробіг, ціна, наявність гарантійного сервісу протягом року (так/ні)	За спаданням ціни (вибору)	Автомобілі марки «Mazda» з гарантійним сервісом
48	Прайс-лист WIFI-адаптерів у магазині: фірма-виробник, модель, швидкість WI-FI у Мбіт/с, ціна, дата поставлення, наявність рознімів USB (так/ні)	За спаданням швидкості (вставки)	WIFI-адаптери фірми «Asus» з наявністю рознімів USB
49	Анкетні дані про співробітників: табельний номер, ПІБ, посада, дата прийому на роботу, сімейний стан, наявність неповнолітніх дітей	В алфавітному порядку ПІБ (бульбашковий)	Співробітники - чоловіки, які працюють на підприємстві понад 10 років
50	Список книг у книжковому магазині: назва книги, автор, категорія, дата надходження, ціна, наявність акційної знижки (так/ні)	За спаданням ціни (вибору)	Книги категорії "Програмування" з акційною знижкою

ВКАЗІВКИ ДО ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Вступ

Потрібно описати важливість знань, які були отримані в межах дисципліни “Технології програмування”, для професійної діяльності - захисту інформації.

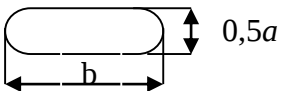
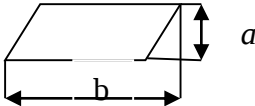
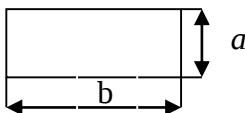
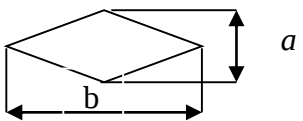
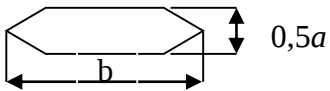
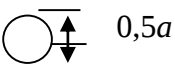
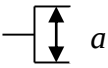
Теоретична частина

Описати основні елементи мови програмування C++, які будуть використані при створенні проекту за власним варіантом: структури, тип дата-час, робота з текстовими та бінарними файлами, обробка помилок за допомогою виключень та базові алгоритми сортування (бульбашковий, вибору, вставки). Опис повинен складатися з визначення елемента, його оголошення та прикладу його застосування. Теоретичний матеріал можливо отримати з електронного конспекту лекцій та рекомендованої літератури.

Проектування основних функцій програми

Перед тим як почати працювати над кодом програми, потрібно спроектувати структуру програми, тобто визначити основні структури даних та функції, які будуть вирішувати технічне завдання, також розробити блок-схеми найбільш складних функцій для розуміння послідовності дій для їх реалізації, це блок-схему для функції сортування, алгоритм якого вказано в індивідуальному варіанті завдання, та блок-схему для вибору елементів з масиву власної структури за умовою варіанта. Блок-схеми створювати строго відповідно до вимог використання елементів блок-схеми для надання різних інструкцій мови програмування та правил їх оформлення. В табл. 2 надані основні блоки для створення блок-схеми, а на рис.1 надано приклад їх застосування.

Таблиця 2 - Графічні елементи блок-схеми

Назва блоку	Графічне надання	Призначення
Початок-кінець		Початок (кінець) програми
Блок введення-виведення		Введення даних і вивід результату
Блок обчислення або присвоєння		Обчислення за формулою або присвоєння
Умовний блок		Вибір за умовою напрямку виконання алгоритму
Блок модифікації (початок циклу)		Визначає початкове, кінцеве та крок циклу
З'єднання блоків		З'єднання перерваних ліній потоку
Блок виклику підпрограми		Виклик підпрограми
Коментарі		Зв'язок між елементами схеми та пояснення

На рис. 1 показано приклад блок-схеми для функції вибору за умовою. У функцію “Вибір за умовою” передається покажчик на масив деяких значень та розмір цього масиву. Перший блок тіла функції – блок модифікації, створює зміну циклу i та присвоює їй початкове та кінцеве значення, наступні два блоки – це тіло циклу, де спочатку перевіряється умова вибору елементів масиву (умовний блок), а потім, якщо значення задовольняє умові, цей елемент виводиться на екран (блок введення-виведення) інакше переходимо до наступного елемента, по завершенню циклу виходимо з функції.

Більшу кількість прикладів створення блок-схем для різних алгоритмів надано у методичних посібниках зі списку літератури. Блок-схеми потрібно створювати за допомогою онлайн застосунків для побудови діаграм

app.diagrams.net або графічними засобами текстового редактора Word меню «Вставка/Фігури».

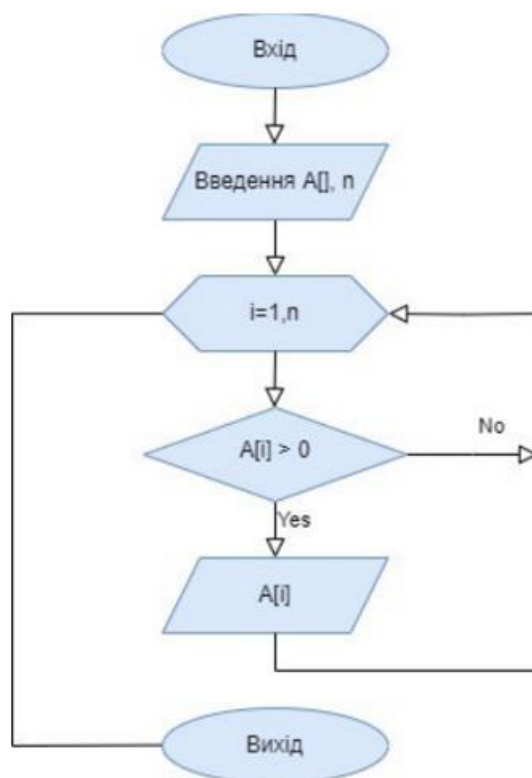


Рисунок 1 – Приклад блок-схеми функції вибору за умовою

Розробка програмного коду

В цьому розділі треба надати увесь код програми з його коментуванням. Код надається шрифтом Courier New 12 з міжрядковим інтервалом 1,0 без відступу.

Тестування програми

Надати шість скріншотів результатів роботи програми згідно з індивідуальним завданням до курсової роботи, надати підписи до скріншотів з поясненнями:

- 1) скріншот змісту бінарного файлу після запису інформації;
- 2) скріншот змісту вихідного масиву;
- 3) скріншот відсортованих за змістом вказаного поля записів вихідного масиву;
- 4) скріншот виведення даних за умовою;
- 5) скріншот змісту текстового файлу з даними за умовою;
- 6) скріншот результату спрацювання виключення.

ПРИКЛАД РОЗРОБКИ ПРОГРАМИ

Технічне завдання:

- створити бінарний файл для зберігання даних про співробітників з наступною структурою: прізвище, ім'я, по батькові, посада, дата прийому на роботу, зарплатня.
- відсортувати співробітників у файлі за спаданням зарплатні з використанням алгоритму Шелла;
- вивести інформацію про інженерів, які працюють понад 5 років;
- створити текстовий файл з інформацією про співробітників, які мають зарплатню нижче середньої.

Проектування програмної системи:

Напочатку надати назви функцій, за допомогою яких реалізується завдання курсової роботи та детальний їх опис.

Програма містить глобальне оголошення структури **employee** і наступні функції:

add_record – додавання одного запису у бінарний файл;

view_file – перегляд змісту (виведення на екран) бінарного файла;

number_of_records1 або **number_of_records2** (на вибір) – визначення кількості записів у файлі;

create_array – заповнення динамічного масиву даними з бінарного файла;

sort_array – сортування динамічного масиву за зменшенням зарплатні;

view_array – перегляд (виведення на екран) динамічного масиву;

array_to_file – заповнення файла даними з динамічного масиву;

select_data або **select_data2** (на вибір) – виведення на екран відомостей про інженерів, які працюють понад 5 років;

create_text_file – створення і заповнення текстового файла даними з бінарного файла;

view_text_file – перегляд (виведення на екран) змісту текстового файла;

main – головна функція, в якій виконуються всі основні оголошення і виклики описаних вище функцій.

Далі створюються блок-схеми алгоритму сортування та вибору за умовою згідно з технічним завданням.

Для сортування буде застосовано алгоритм Шелла, швидкість якого значно більша за швидкість алгоритмів простого сортування (бульбашковий, вибору і вставок). Сортування Шелла було названо в честь його винахідника – Дональда Шелла, який опублікував цей алгоритм у 1959 році. Цей алгоритм використовує мінімум пам'яті і показує високу швидкість при сортуванні. По суті в методі Шелла застосовуються порівняння та перестановки елементів, аналогічні алгоритму вставок, але при цьому порядок порівнюваних елементів зовсім інший. Загальна ідея сортування Шелла полягає в порівнянні на початкових стадіях сортування пар значень, розташованих досить далеко один від одного в впорядкованому наборі даних. Така модифікація методу сортування дозволяє швидко переставляти далекі неупорядковані пари значень (сортування таких пар зазвичай вимагає значної кількості перестановок, якщо використовується порівняння тільки сусідніх елементів).

Загальна схема методу полягає в наступному.

Крок 1. Відбувається упорядкування елементів $n/2$ пар $(x_i, x_{i/2+1})$ для $1 < i < n/2$.

Крок 2. Упорядковуються елементи в $n/4$ групах з чотирьох елементів $(x_i, x_{i/4+i}, x_{i/2+i}, x_{3i/4+i})$ для $1 < i < n/4$.

Крок 3. Упорядковуються елементи вже в $n/4$ групах з восьми елементів і т.д.

На останньому кроці упорядковуються елементи відразу усьому масиві $x_1, x_2, \dots, x_n$. На кожному кроці для упорядкування елементів у групах використовується метод сортування вставками, блок-схема показана на рис. 2.

Середній час роботи алгоритму сортування Шелла дорівнює $O(n^{1.25})$, для найгіршого випадку оцінкою є $O(n^{1.5})$, тобто близький до часу роботи швидких алгоритмів сортування $O(n \log_2 n)$.

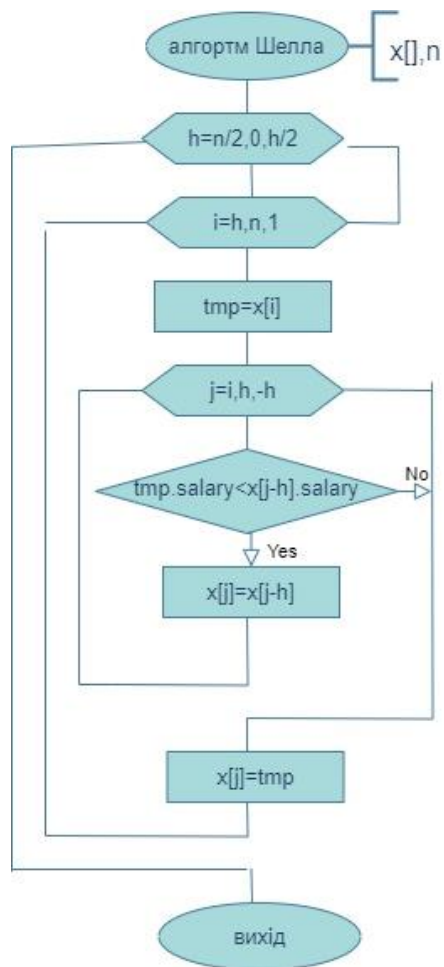


Рисунок 2 – Блок-схема алгоритма Шелла

Блок-схема алгоритму вибору за умовою показана на рис. 3, яка аналогічна блок-схемі рис. 1, тільки реалізована з використанням циклу while.

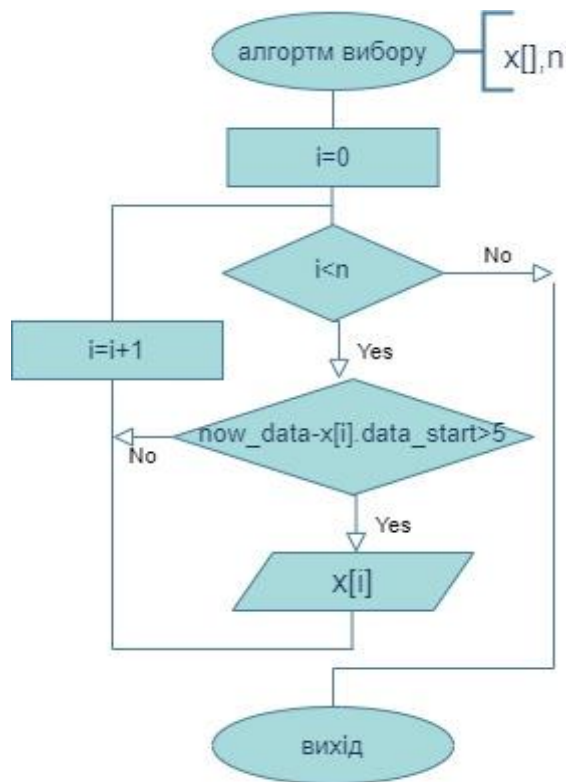


Рисунок 3 – Блок-схема алгоритму вибору всіх співробітників, які працюють понад 5-ти років

Розробка коду програмної системи

Надається код програми з його детальним коментарем.

Код програми

```

//Підключаєм необхідні бібліотеки

#include <iostream>
#include <stdio.h> //Опрацювання файлів
#include <errno.h> //Опрацювання помилок при роботі з файлами
#include <windows.h> //Підтримка кирилиці

using namespace std;

//Оголошення типу Співробітник (employee)

struct employee
{
    char surname[20];
    char name[20];
    char patronymic[20];

```

```

    char position[20];
    char start_date[11];
    float salary;
};

//Додавання одного запису у бінарний файл. Параметри – покажчик на файл
// та ім'я файла

void add_record(char* name)
{
    employee z; FILE* f;
    //Відкриття файла і перевірка успішності відкриття
    errno_t err;
    if ((err = fopen_s(&f, name, "ab")) != 0)
        //Повідомити про стандартну помилку відкриття файла err за
        //допомогою виключення у функції add_record()
        throw " add_record - файл не відкрито!\n";
    //Зчитати запис з клавіатури (дані вводяться в рядок через табуляцію)
    scanf_s("%s\t%s\t%s\t%s\t%s\t%f", z.surname, 20, z.name, 20,
z.patronymic, 20, z.position, 20, z.start_date, 11, &z.salary);
    //Зробити запис у файл
    fwrite(&z, sizeof(employee), 1, f);
    //закрити файл
    fclose(f);
}

// Перегляд файла.Параметри – покажчик на файл та ім'я файла

void view_file(char* name)
{
    employee z; FILE* f;
    errno_t err;
    if ((err = fopen_s(&f, name, "rb")) != 0)
        throw " view_file - файл не відкрито!\n";
    cout << "\nПерегляд файла" << endl;
    cout << "Прізвище\tІм'я\tПо батькові\tПосада\tДата прийому на
роботу\tЗаробітна платня\n" << endl;

    //Зчитувати послідовно записи з файла, доки вони не закінчаться
    while (fread(&z, sizeof(employee), 1, f))
    {
        // Вивести запис на екран у рядок через табуляцію
        printf("%s\t%s\t%s\t%s\t%s\t%.2f\n", z.surname, z.name,
z.patronymic,
        z.position, z.start_date, z.salary);
    }
    fclose(f);
}

```

```

//Визначення кількості записів у файлі. Параметри – покажчик на файл та ім'я файла
// Результат – ціла кількість (кількість записів)
int number_of_records1(char* name)
{
    int n; FILE* f;
    errno_t err;
    if ((err = fopen_s(&f, name, "rb")) != 0)
        throw " number_of_records1 - файл не відкрито!\n";
    //установити курсор на кінець файла (0 – зміститися на 0 байт, 2 – від кінця файла)
    fseek(f, 0, 2);
    //Визначити поточне положення (на скільки байт від початку файла зміщений курсор) та розділити на розмір одного запису
    n = ftell(f) / sizeof(employee);
    fclose(f);
    return n;
}

//Другий спосіб визначення кількості записів у файлі.
//Параметри – покажчик на файл та ім'я файла. Результат – змінна цілого типу кількість (кількість записів)

int number_of_records2(char* name)
{
    int n = 0; employee z; FILE* f;
    errno_t err;
    if ((err = fopen_s(&f, name, "rb")) != 0)
        throw " number_of_records2 - файл не відкрито!\n";
    //Зчитувати по черзі всі записи та збільшувати лічильник на 1
    while (fread(&z, sizeof(z), 1, f))
        n++;
    fclose(f);
    return n;
}

//Заповнення (створення) масиву з бінарного файла.
//Параметри – масив, покажчик файла, ім'я файла.
void create_array(employee* a, int kol, char* name)
{
    FILE* f;
    //Оголошення змінної системної помилки
    errno_t err;
    if ((err = fopen_s(&f, name, "rb")) != 0)
        throw " create_array - файл не відкрито!\n";
    //Захоплення виключення, яке фіксує змінна err
    //Вважати змінну kol записів з файла в масив однією командою
    fread(a, sizeof(employee), kol, f);
    fclose(f);
}

```

//Сортування масиву з використанням алгоритма Шелла

```
void sort_array(employee* a, int kol)
{
    int i, j, h;
    employee tmp;
    for (h = kol / 2; h > 0; h /= 2)
        for (i = h; i < kol; i++)
        {
            tmp = a[i];
            for (j = i; j >= h; j -= h)
            {
                if (tmp.salary > a[j - h].salary)
                    a[j] = a[j - h];
                else
                    break;
            }
            a[j] = tmp;
        }
}
```

//Виведення динамічного масиву на екран

```
void view_array(employee* a, int kol)
{
    cout << "Прізвище\tІм'я\tПо батькові\tПосада\tДата прийому на  
роботу\tЗаробітна платня\n" << endl;
    for (int i = 0; i < kol; i++)
        printf("%s\t%s\t%s\t%s\t%s\t%10.2f\n", a[i].surname, a[i].name,  
a[i].patronymic, a[i].position, a[i].start_date, a[i].salary);
}
```

//Створення бінарного файлу з динамічного масиву (повернення даних у файл)

```
void array_to_file(employee* a, int kol, char* name)
{
    FILE* f;
    errno_t err;
    if ((err = fopen_s(&f, name, "wb")) != 0)
        throw "array_to_file - файл не відкрито!\n";
    fwrite(a, sizeof(employee), kol, f);
    fclose(f);
}
```

//Виведення даних за умовою

```
void select_data(char* name)
{
    employee z; FILE* f;
    time_t t, t1; struct tm tmstr;
    double seconds, yearinseconds = 365.25 * 24 * 60 * 60;
    t = time(NULL);
    errno_t err;
```

```

    if ((err = fopen_s(&f, name, "rb")) != 0)
        throw " select_data - файл не відкрито!\n";
    cout << "\nВибір записів" << endl;
    cout << "Прізвище\tІм'я\tПо батькові\tПосада\tДата прийому на
роботу\tЗаробітна платня\n" << endl;
    while (fread(&z, sizeof(employee), 1, f))
    {
        //Розбиваємо рядок на частини за заданим форматом: з цілих чисел
        (%d), розділених крапками
        //і записуємо їх у поля структури tmstr день, місяць та рік
        //при цьому перевіряємо правильність введення дати у змінну
        z.start_date
        //створимо цілу змінну s і надамо їй значення, яке повертає
        функція sscanf_s()
        //у нашому випадку, якщо формат дати перетворено успішно, то s=3
        int s = sscanf_s(z.start_date, "%d.%d.%d", &tmstr.tm_mday,
&tmstr.tm_mon, &tmstr.tm_year);
        cout << "sscanf=" << s << endl;
        //якщо формат дати не перетворено, оброблюємо виключення
        if(s<3)
            throw "при введенні даних про працівників дату початку роботи
треба ввести у форматі 3.12.2023 /n";
        for (int i = 0; i < 5; i++) cout << "\n";
        //Віднімаємо 1 від місяця та 1900 від року, полям згодом
        присвоюємо нулі
        tmstr.tm_mon -= 1; tmstr.tm_year -= 1900; tmstr.tm_hour = 0;
        tmstr.tm_min = 0; tmstr.tm_sec = 0;
        //Перетворюємо під час типу time_t та обчислюємо різницю дат у
        секундах
        t1 = mktime(&tmstr);
        seconds = difftime(t, t1);
        //Якщо різниця в часі понад 5-ти років (у секундах) і посада -
        «інженер»
        if (seconds > yearinseconds * 5 && strcmp(z.position, "інженер")
== 0)
            //Виводимо запис на екран
            printf("%s\t%s\t%s\t%s\t%s\t%6.2f\n", z.surname, z.name,
z.patronymic, z.position, z.start_date, z.salary);
    }
    fclose(f);
}

//Створення текстового файла
void create_text_file(char* bname, char* tname)
{
    employee z; float sum = 0, avg; FILE* fb; FILE* ft;
    int k = number_of_records1(bname);
    //Відкриваємо бінарний файл для читання
    errno_t err;
    if ((err = fopen_s(&fb, bname, "rb")) != 0)

```

```

        throw " create_text_file - файл не відкрито!\n";
//Проходимо по бінарному файлу та обчислюємо сумарну зарплатню
while (fread(&z, sizeof(employee), 1, fb))
    sum += z.salary;
//Обчислюємо середню зарплатню як відношення сумарної зарплатні та
кількості записів
    avg = sum / k;

//Переходимо на початок файлу, щоб знову по ньому пройти
fseek(fb, 0, 0);
//Відкриваємо текстовий файл для запису
if ((err = fopen_s(&ft, tname, "wt")) != 0)
    throw " create_text_file - файл не відкрито!\n";
//проглядаємо бінарний файл та зчитуємо записи

while (fread(&z, sizeof(employee), 1, fb))
{
    //Якщо зарплатня нижча за середню, то виводимо запис у текстовий файл

        if (z.salary < avg)
        {
            fprintf(ft, "%s\t%s\t%s\t%s\t%s\t%g\n", z.surname, z.name,
z.patronymic,
                z.position, z.start_date, z.salary);
        }
    }
    fclose(ft);
    fclose(fb);
}

//Перегляд текстового файла
void view_text_file(char* name)
{
    employee z; FILE* f;
    errno_t err;
    if ((err = fopen_s(&f, name, "rt")) != 0)
        throw " view_text_file - файл не відкрито!\n";
    cout << "\nПерегляд текстового файла" << endl;
    cout << "Прізвище\tІм'я\tПо батькові\tПосада\tДата прийому на
роботу\tЗаробітна платня\n" << endl;
    while (!feof(f))
    {
        fscanf_s(f, "%s\t%s\t%s\t%s\t%s\t%f\n", z.surname, 20, z.name,
20, z.patronymic, 20, z.position, 20, z.start_date, 11, &z.salary);
        printf("%s\t%s\t%s\t%s\t%s\t%6.2f\n", z.surname, z.name,
z.patronymic, z.position, z.start_date, z.salary);
    }
    fclose(f);
}

```

```

//Головна функція
int main()
{ //підтримка кирилиці при введенні та виведенні даних
    SetConsoleOutputCP(1251);
    SetConsoleCP(1251);
    char name[] = "kursfile.dat"; //ім'я бінарного файла
    char tname[] = "textfile.txt"; //ім'я текстового файла
    //Введення даних у бінарний файл
    cout << "Введення інформації про співробітників:" << endl;
    cout << "Прізвище\tІм'я\tПо батькові\tПосада\tДата прийому на
роботу\tЗаробітна платня\n" << endl;
    try {
        for (int i = 0; i < 3; i++)
            add_record(name);
        //Виведення файлу на екран
        view_file(name);

        //Визначення кількості записів у файлі
        int kol = number_of_records1(name);
        cout << "Кількість записів у файлі: " << kol << endl;
        //cout<<"Number of records in the file:
"<<number_of_records2(name)<<endl;
        //пропуск 5 рядків для розділу при виведенні даних
        for (int i = 0; i < 5; i++) cout << "\n";
        //Оголошення динамічного масиву
        employee* a = new employee[kol];
        //Заповнення динамічного масиву значеннями
        create_array(a, kol, name);

        //Виведення динамічного масиву на екран
        cout << "Масив:" << endl;
        view_array(a, kol);
        for (int i = 0; i < 5; i++) cout << "\n";
        //Сортування масиву
        sort_array(a, kol);
        //Виведення динамічного масиву на екран
        cout << "\nВідсортований масив:" << endl;
        view_array(a, kol);
        for (int i = 0; i < 5; i++) cout << "\n";

        //Повернення відсортованих даних до бінарного файла
        array_to_file(a, kol, name);
        //Звільнення пам'яті, яку займає масив
        delete[]a;
        //Виведення файлу на екран
        view_file(name);
        for (int i = 0; i < 5; i++) cout << "\n";

        //Виведення даних з файлу на екран за умовою
        select_data(name);
    }
}

```

```

view_file(name);
for (int i = 0; i < 5; i++) cout << "\n";

//Створення текстового файла з бінарного
create_text_file(name, tname);
//Виведення змісту текстового файла
view_text_file(tname);
}
//опрацювання виключень, які є у функціях програми
catch (const char* err)
{
    printf(err);
    exit(1);
}
system("pause");
return 0;
}

```

Тестування коду

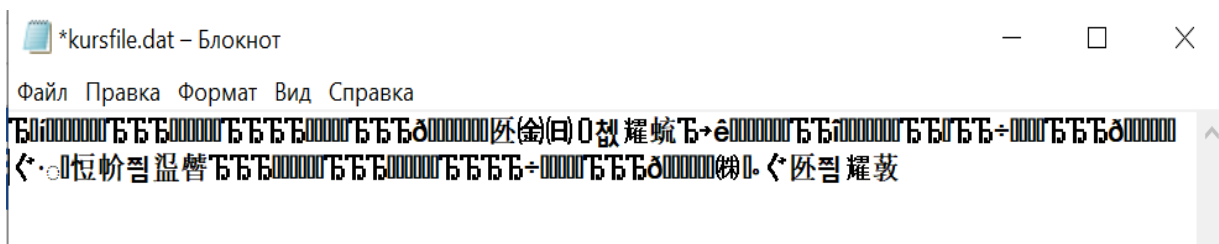


Рисунок 1 – Перегляд бінарного файла з вихідними даними програми у Блокноті.

Перегляд файлу

Прізвище	Ім'я	Побатькові	Посада	Дата прийому на роботу	Заробітна платня
Куляк	Петро	Васильович	інженер	20.03.1998	5000.00
Стасюк	Микола	Антонович	майстер	12.10.2005	3000.00
Мутян	Виктор	Петрович	інженер	4.05.2020	6000.00

Кількість записів у файлах: 3

Рисунок 2 – Виведення даних бінарного файла та кількості записів у файлі на екран за допомогою функцій **view_file**, **number_of_records1**

Вибір записів

Прізвище	Ім'я	Побатькові	Посада	Дата прийому на роботу	Заробітна платня
Куляк	Петро	Васильович	інженер	20.03.1998	5000.00

Рисунок 3 – Вибір записів бінарного файла за умовою технічного завдання за допомогою функції **select_data**

Перегляд файлу					
Прізвище	Ім'я	Побатькові	Посада	Дата прийому на роботу	Заробітна платня
Мутян	Виктор	Петрович	інженер	4.05.2020	6000.00
Куляк	Петро	Васильович	інженер	20.03.1998	5000.00
Стасюк	Микола	Антонович	майстер	12.10.2005	3000.00

Рисунок 4 – Сортування вихідного масиву за допомогою алгоритму, який є у технічному завданні, функція **sort_array**

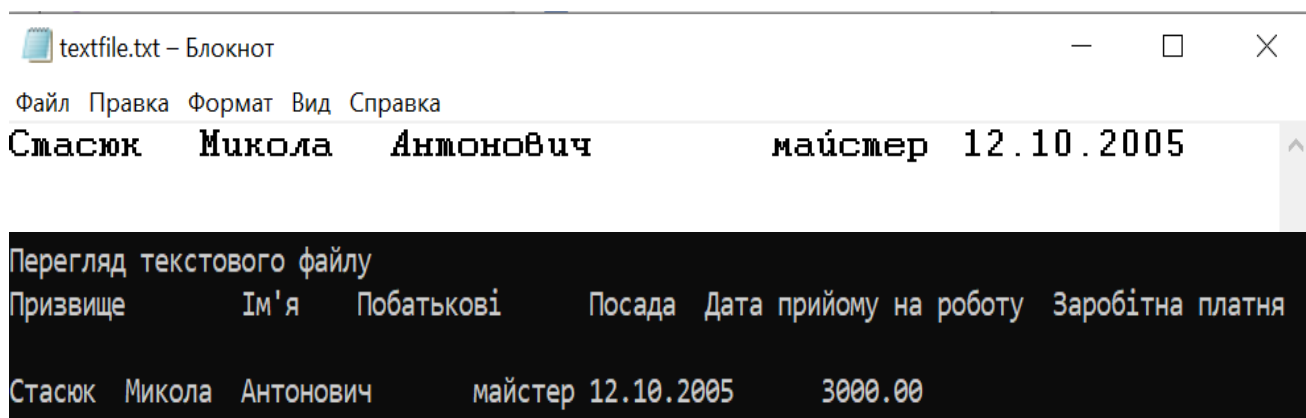


Рисунок 5 – Вигляд змісту текстового файлу у Блокноті та перегляд текстового файлу за допомогою функції **view_text_file**

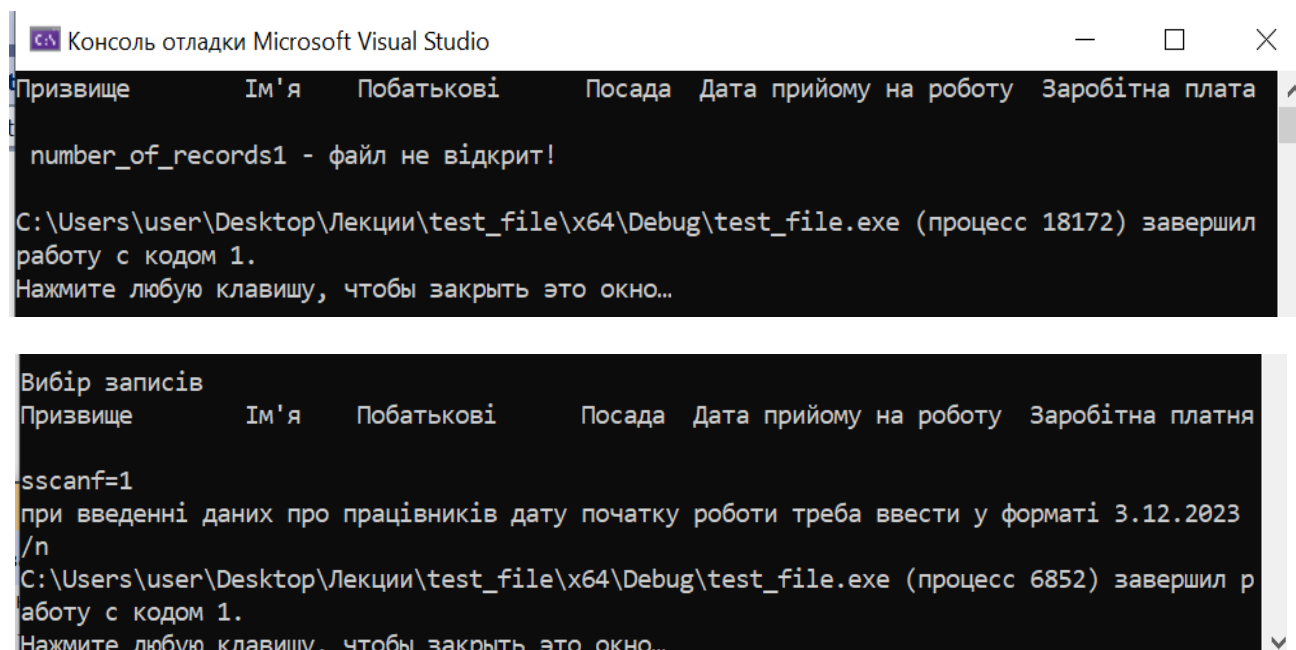


Рисунок 6 – Перегляд повідомлень, які з'являються при обробленні виключень у функціях програми

ЛІТЕРАТУРА

1. Конспект лекцій:
URL: <https://e-learning.suitt.edu.ua/login/index.php#section-1>
<https://metod.suitt.edu.ua/>
2. Комп'ютерні технології та програмування. Ч.1. Базові алгоритми : метод. вказ. до лаб. і практич. робіт. /Трофименко О.Г., Прокоп Ю.В., Буката Л.М. Одеса:ОНАЗ ім.О.С. Попова, 2016. 119 с. URL: <https://metod.suitt.edu.ua/>
3. Комп'ютерні технології та програмування. Ч.2. Програмування структурованих даних: метод. вказ. /укладачі: Трофименко О.Г., Прокоп Ю.В., Буката Л.М. Одеса:ОНАЗ ім.О.С. Попова, 2016. 154 с. URL: <https://metod.suitt.edu.ua/>
4. Комп'ютерні технології та програмування. Ч.3. Програмне опрацювання файлів : методичні вказівки. / укладачі: Трофименко О.Г., Прокоп Ю.В., Буката Л.М. Одеса:ОНАЗ ім.О.С. Попова, 2016. 92 с. URL: <https://metod.suitt.edu.ua/>
5. Алгоритми та структури даних: конспект лекцій. Частина 2. Алгоритми пошуку, стиснення даних, внутрішнього та зовнішнього сортування, алгоритми на графах / Укладачі: О.Д. Воробйова, Л.В. Глазунова – Одеса:ОНАЗ ім.О.С. Попова, 2017. 48с. URL: <https://metod.suitt.edu.ua/>
6. Седжвик Н. Фундаментальные алгоритмы на C++. Части 1-4. Киев: Диасофт, 2001. 688 с.
7. Кнут Д. Искусство программирования. Т. 1-4. Кнут Д.– М.: Вильямс, 2006. 682 с.
8. Сборник задач по программированию. /Ред. Л.В. Глазунова. Одесса: ОНАС им. А.С. Попова, 2011. 211 с.

ЗМІСТ

ВСТУП.....	3
1. СТРУКТУРА КУРСОВОЇ РОБОТИ	5
2. ВИМОГИ ЩОДО ОФОРМЛЕННЯ КУРСОВОЇ РОБОТИ	6
3. КУРСОВЕ ЗАВДАННЯ.....	9
4. ВКАЗІВКИ ДО ВИКОНАННЯ КУРСОВОЇ РОБОТИ.....	17
5. ПРИКЛАД РОЗРОБКИ ПРОГРАМИ.....	20
6. ЛІТЕРАТУРА.....	32

Навчальне видання

Укладачі: Глазунова Людмила Володимирівна
Буката Людмила Миколаївна

**“Обробка структурованих даних за допомогою мови С++” з дисципліни
“Технології програмування” : методичні вказівки з виконання курсової роботи**

Видається в авторській редакції

ДУІТЗ
65023, Одеса, вул. Кузнечна, 1
обсяг 2.0 др.а., електр. версія