**ЗМІСТ**

Вcтуп……………………………………………………………………………...5

1…………………………………………………………………………………...7

2…………………………………………………………………………………...8

3………………………………………………………………………………….10

3.1……………………………………………………………………………10

3.2…………………………………………………………………………....11

3.3……………..………………………………………...…………..…….....19

3.4...……………..…………..…………………………………………..…...23

4. ……………………………………………….………………………………..24

Висновки ………………………………………………………………………..27

Список використаних джерел………………………………………………….28

Додаток 1……………………...………………………………………………...30

Додаток 2…………………...…………………………………………………...32

Додаток 3……………………...………………………………………………...40

**ВСТУП**

Актуальність: досліджувана тема актуальна, тому що написання характеристик необхідне у сучасному світі через серйозне практичне значення. Постає проблема для студентів, класних керівників та деканатів в наданні оцінки ділових і моральних якостей, щоб кожну особу можна було охарактеризувати відповідно до її особливостей роботи й характеру. Є багато способів вирішення цієї проблеми, але автоматизація написання цього документа методами програмування вирішує її, суттєво полегшує даний процес, виключає неправильність самостійного складання, дозволяє зекономити час та є корисною у повсякденному житті.

Мета: вивчення та аналіз літературних джерел інформації, дослідження методів створення програмного продукту для автоматичного написання характеристик, на основі здобутих навичок програмування мовою С++ розроблення програмної реалізації алгоритмів, вивчення особливостей характеристики як документу, закріплення теоретичних знань та практичних навичок програмування, набутих при вивченні дисципліни “Основи програмування”.

Завдання: розробити програму, яка буде автоматично формувати текст на основі розробленого шаблону та за допомогою введених даних користувачем про особу, на яку писатиметься документ(ім’я, вік, стать, захоплення, особисті якості), та писати характеристику, створити зручний інтуїтивно зрозумілий користувацький інтерфейс.

Об’єктом дослідження курсової роботи є програмне забезпечення для автоматичного написання характеристик.

Предмет дослідження – особливості реалізації алгоритму написання програмного продукту для автоматизації написання документу.

Методи дослідження:

- пошук – збір інформації, вивчення основних джерел інформації, складових частин автоматизації характеристики;

- аналіз – перевірка знайденої інформації, кількісний та якісний аналіз даних, їх систематизація;

- порівняння – встановлення відмінностей між різними алгоритмами реалізації програми та знаходження спільних рис, вибір найбільш підходящого;

- рефлексія – дослідження конкретних явищ;

- експеримент – виконання дій для перевірки гіпотези, правильності алгоритму;

- формалізація – представлення об’єктів шляхом відображення їхнього змісту в знаковій формі, наприклад, у вигляді діаграми або таблиці.

// 2 сторінки

**РОЗДІЛ 1. АНАЛІЗ ЗАВДАННЯ**

Навчаючись або працюючи у школах, вищих навчальних закладах, багатьом доводилось писати характеристику — часто використовуваний документ, у якому в офіційній формі висловлено громадську думку про працівника, учня, студента як члена колективу, про його громадську або трудову діяльність, і який складається на вимогу або письмовий запит іншої установи для подання до цієї установи.

**1.1. Теоретична складова**

Ця програма спрямована на полегшення процесу написання характеристик, тому вона є актуальною для будь-якого віку й часу, адже написання вручну є досить важкою справою. Як і для будь-якого програмного продукту, необхідний чіткий алгоритм реалізації створення програмного забезпечення.

Характеристика вважається офіційним документом, адже надається однією певною установою іншій, тому стиль тексту обов’язково має бути офіційним.

Програмний застосунок повинен відповідати таким вимогам:

- користувач може обирати вид необхідної характеристики;

- при введенні даних про особу програма повинна перевірити їх;

- базою даних слугує файл, в який записуються вхідні дані;

- за бажанням користувача має бути можливість створити нову папку для збереження файлу;

- на основі введених даних має бути сформована готова характеристика та записана у файл;

- інтерфейс програми має бути зрозумілим та зручним для всіх користувачів;

- в кінці документа обов’язково має бути дата написання та місце для підпису.

**1.2. Аналіз вхідних/вихідних даних**

Для написання характеристики необхідно вказувати такі основні реквізити про особу:

* прізвище, ім’я, по батькові того, кому видається характеристика;
* стать;
* рік народження;
* особисті якості людини;
* захоплення.

Реквізитами для характеристики є:

* рейтинг характеристики(позитивна, негативна, наполовину позитивна);
* мова написання;
* призначення(куди пишеться).

Вище перелічені реквізити виступають вхідними даними у програмі. Комунікація з користувачем здійснюється за допомогою консолі. Програма повинна друкувати запрошення на ввід цих даних користувачем, тому введення відбувається у зручному та зрозумілому форматі. Має відбуватися зчитування вхідних даних, перевірка на правильність, а у випадку неправильного вводу знову дати можливість користувачеві ввести їх, а потім записати перевірені дані у файл.

Вихідними даними є сформована програмою характеристика, яка також записується у файл для зручності у використанні користувачем. Необхідною умовою є виведення дати створення характеристики.

**1.3. Вибір мови програмування та системи розробки**

Для роботи над алгоритмами та безпосереднього написання коду необхідно було обрати підходящу мову(C/C++) та середовище програмування. Було прийняте рішення обрати об’єктно-орієнтовану мову С++, а середовище розробки — Visual Studio. Для колективної розробки було використано веб-сервіс GitHub.

С++ — мова програмування високого рівня, що підтримує декілька парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Була розроблена у 1979 році.

Розвиток мови С ++ відбувався на базі мови С, і, за невеликим винятком, мова С була збережена як підмножина C ++. Мову використовують для системного програмування, розробки програмного забезпечення, написання драйверів, потужних серверних та клієнтських програм, а також для розробки розважальних програм.

Серед основних причин вибору мови С++ є можливості об’єктно-орієнтованого програмування, можливе завдяки трьом основним поняттям на яких базується саме ООП:

1. Інкапсуляція. Також відома як приховування даних. Зміст інкапсуляції полягає у приховуванні від зовнішнього користувача деталей реалізації об'єкта, замість цього надаючи інтерфейс взаємодії з ним.

2. Успадкування. Це означає, що об'єкти (класи) можуть переймати деякі властивості у своїх прабатьків. Це призводить до повторного використання вже написаного одного разу коду. Підкласи успадковують атрибути та поведінку своїх батьківських класів, і можуть мати нові власні атрибути. Тобто утворюється ієрархія з класів, де від батьківського класу походять усі інші дочірні класи.

- Поліморфізм означає залежність поведінки від класу, в якому ця поведінка викликається, тобто, два або більше класів можуть реагувати по різному на однакові повідомлення. Це спричинене зміною в одного з класів якогось методу, шляхом запису іншого алгоритму[ ].

В якості середовища програмування було обране Visual Studio. Це інтегроване середовище розробки представляє собою багатофункціональну програму, яку можна використовувати для різних аспектів розробки програмного забезпечення. У ньому міститься не тільки редактор і відладчик, а й компілятор, засоби автозавершення коду та велика кількість інших функцій для спрощення процесу розробки.

Враховуючи всі переваги мови програмування С++ та середовища розробки Visual Studio, можна зробити висновок, що цей вибір є найоптимальнішим рішенням для створення програми за завданням курсової роботи.

// 8 сторінок

**РОЗДІЛ 2. РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ**

Розроблений проект був створений методами колективної розробки з постійним редагуванням та вдосконаленням програми новим функціоналом. Додавались доопрацювання як для правильного функціонування програми, так і для зручності у використанні.

**2.1. Загальний опис продукту**

Весь код програми було вирішено розподілити в декілька файлів для розбиття на окремі модулі, розділення інтерфейсу з реалізацією та для зручності в роботі.

**2.2. Опис головних класів, змінних та функцій програми**

Вся програма розділена на декілька підрозділів та функції main, яка зв’язує ці підрозділи та утворює повноцінний програмний продукт.

Створено 2 класи:

* Student;
* Characteristic.

В класі Student ініціалізовані поля для запису користувачем інформації про людину, на яку буде писатись характеристика. З таблиці 2.1. можна побачити основні поля класу Student та їх призначення.

Таблиця 2.1. Поля класу Student

|  |  |  |
| --- | --- | --- |
| Назва змінної | Тип | Призначення |
| studentName | string | ім'я студента |
| studentSex | int | стать студента |
| studentAge | string | вік студента |
| studentHobby | int | види захоплень |
| numOfQualities | int | кількість особистих якостей |
| studentQualities | string\* | масив для запису особистих якостей |
| studentQuality | string | окремо взята якість студента |

Методи, за допомогою яких можна виконувати дії над даними класу Student:

- void studentInput() — функція, за допомогою якої користувач може ввести дані про студента/учня;

- void studentOutput() — функція для виведення в консоль вхідних даних про особу користувачем;

- bool studentCharact(string inputFile, string outputFile) — функція, що записує сформовану характеристику у файл, де параметр string inputFile є ім’ям файлу, з якого зчитуються дані, а string outputFile є ім’ям файлу, в якому безпосередньо створюється готова характеристика. Функція повертає булеве значення, яке характеризує успішність написання характеристики;

- ~Student() — деструктор для звільнення пам’яті.

В класі Characteristic містяться основні відомості про характеристику, які зображено в таблиці 2.2.

Таблиця 2.2. Поля класу Characteristic

|  |  |  |
| --- | --- | --- |
| Назва змінної | Тип | Призначення |
| charactRating | int | рейтинг характеристики |
| charactLanguage | int | мова |
| charactPurpose | int | призначення характеристики |
| nameOfStudentFile | string | ім'я файлу для запису відомостей про людину |
| fileExtension | string | розширення файлу |
| directoryName | string | назва директорії для збереження файлів |
| allowableExtensions | string | масив із можливих розширень файлів |

В класі Characteristic містяться такі необхідні методи:

- void charactInput() — функція для введення та вибору даних про тип характеристики;

- void charactOutput() — функція для виведення в консоль введених даних про характеристику;

- string charactChoice() — функція для вибору необхідного шаблону на основі введених користувачем даних, повертає назву необхідного шаблону;

- string getNameOfStudentFile() — використовується для отримання назви файлу, в який ведеться запис, повертає його назву;

- void makeDirectory() — для створення папки, в якій будуть зберігатися файли;

- bool matchExtension(string str) — для перевірки дозволених розширень файлів, де параметр string str — задане розширення, повертає булеве значення, яке означає, чи є задане розширення підходящим, чи ні.

У програмі створено 2 окремі функції, які не є членами класів. Функція string getDate() визначає поточну дату написання характеристики та повертає її, після чого ця дата вставляється у вихідний файл. Функція bool hasOnlyDigits(const string s) використовується для перевірки рядка на наявність символів.

**РОЗДІЛ 3. КЕРІВНИЦТВО КОРИСТУВАЧА**

В даній курсовій роботі інтерфейс програми реалізований в консолі середовища розробки Visual Studio.

**3.1. Опис інтерфейсу**

// діаграма використання

**//3.2. Перевірка роботи програми**

**ВИСНОВКИ**

# В результаті виконання курсової роботи було розроблено та реалізовано алгоритм автоматичного написання характеристики. Програма коректно реалізовує всі необхідні функції, результат відповідає вимогам написання документу характеристики.

В першому розділі записки було проведено детальний аналіз поданого завдання, вивчення теоретичних питань, сформовано постановку задачі.

У другому розділі було обґрунтовано вибір середовища розробки та мови програмування, описано структуру, основні підрозділи, функції та класи програми.

У третьому розділі …

Отже, всі пункти розробки були виконані в повному обсязі. Надалі програмне забезпечення має багато можливостей покращення, вдосконалення та оптимізації, наприклад, шляхом додання графічного інтерфейсу.

При виконанні завдання курсової роботи були освоєні та вдосконалені навички розробки та проектування програмного продукту методами програмування С++.

# **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

http://programming.in.ua/programming/basisprogramming/25-oop.html