**Лабораторна робота №14**

**Робота з файлами та багатофайлові проекти**

**Мета роботи**

Ознайомитись з принципами роботи з файловими даними та отримати практичні навички застосування файлових функцій.

**Хід роботи**

1) Ознайомитись з методичними вказівками до лабораторної роботи та темами "Багатофайлові проекти" і "Робота з файлами";

2) розробити схеми алгоритмів методів класу згідно із завданням за варіантом;

3) за схемами алгоритму виконати оголошення та визначення класу роботи з текстовими файлами;

4) розробити схему алгоритму програми, що демонструє роботу зі сформованим класом;

5) за схемою алгоритму розробити програму на мові С++;

6) розділити програмний код на заголовний та програмний файли;

7) зробити висновки;

8) підготувати звіт про виконання лабораторної роботи, що включає наступні пункти: номер, тема, мета та хід лабораторної роботи, завдання, схема алгоритму програми, лістинг програми з коментуванням кожної інструкції, результат роботи програми (скріншот), висновки.

**Теоретичні відомості та рекомендації до виконання**

Файли виступають основною одиницею зберігання інформації на жорстких дисках. Майже вся робота з персональним комп'ютером зводиться до створення, копіювання, видалення, переміщення файлів та до запису/зчитування вкладеної у них інформації. Файли також дозволяють зберігати поточні дані виконання програм та кінцеві результати проектів для повторного їх зчитування та використання у майбутньому. Тому кожному програмісту необхідно мати навики застосування файлів у своїх проектах.

Достатньо часто деякі класи, функції та змінні відокремлюються у заголовкових файлах з розширенням ".h". Це виконується для забезпечення можливості повторного використання вже розроблених програмних фрагментів без їх повного копіювання у нові проекти через підключення заголовних файлів до основної програми у якості бібліотек. При цьому програмісту немає потреби осягати особливості написання тих чи інших функцій – він їх лише використовує. Завдяки цьому забезпечується можливість створення програмного продукту групою програмістів, кожен з котрих виконує лише свою частину згідно зі заздалегідь запланованими правилами формування функцій.

При виконанні завдання до лабораторної роботи студент повинен засвоїти основні принципи потокової роботи з файлами та навчитися створювати багатофайлові програмні проекти.

Виконання лабораторної роботи починається ознайомленням з методичними вказівками та теоретичними відомостями про можливості потокової роботи з файлами, забезпечені мовою С++. Засвоївши їх, студент приступає до розробки схеми алгоритму програми згідно із завданням за варіантом. За схемою алгоритму виконується оголошення та визначення класу та написання програми, що демонструє роботу з методами двома методами роботи з файлом.

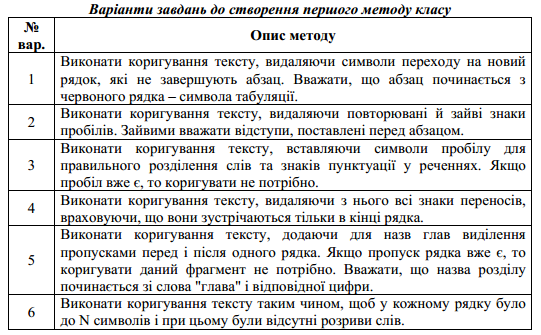
Після перевірки роботи програми її код розділяється на два файли, щоб перетворити проект на багатофайловий.

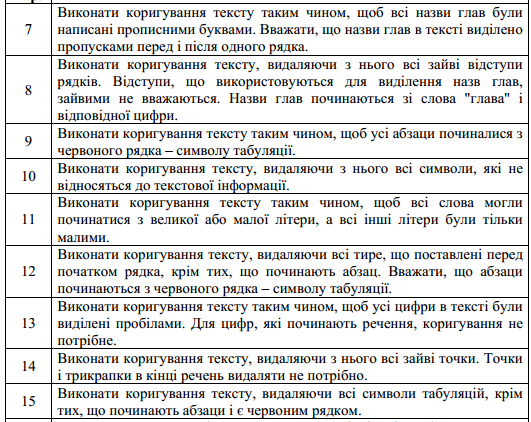
У висновках до звіту студент повинен розкрити призначення теми "Робота з файлами та багатофайлові проекти". Висновок має бути поданий в формі пояснення сфери застосування вивченого матеріалу згідно із розумінням його студентом.

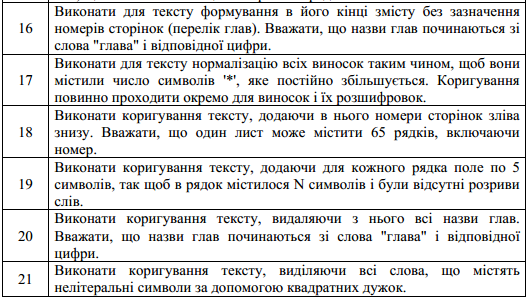
**Завдання для самостійного виконання**

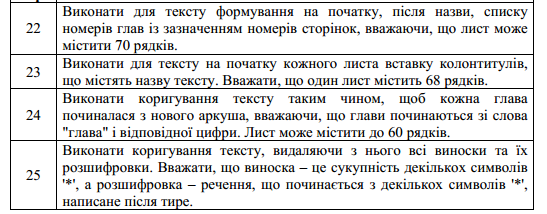
Розробити схему алгоритму та програму, що демонструє використання класу, який містить два методи роботи з текстовою інформацією: форматування тексту та запис результату у файл "OUTPUT1.TXT" (табл. 14.1); створення з літер тексту псевдографічного зображення та запис результату у файл "OUTPUT2.TXT" (табл. 14.2). Текст для передавання через аргументи у методи класу повинен бути взятий з файлу "INPUT.TXT" (шлях до файлу вводиться з консолі). Програмний код поділити на два файли – заголовний та програмний.

Таблиця 14.1

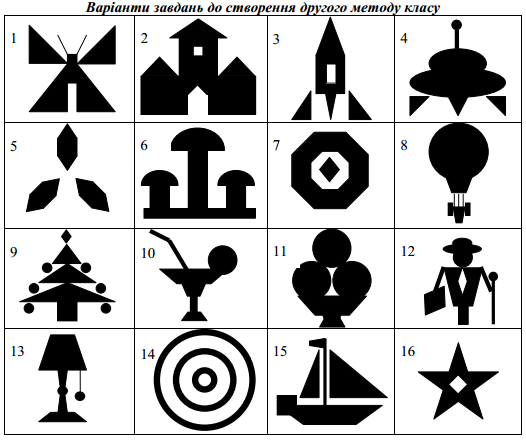


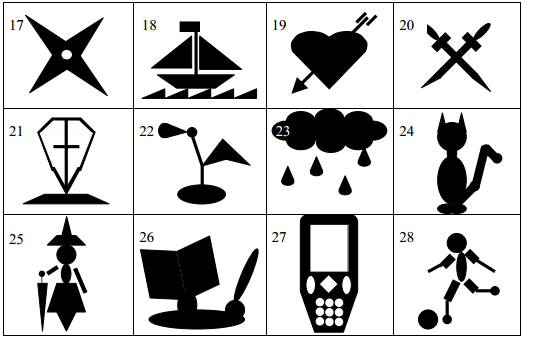






Таблиця 14.2





**Приклад виконання завдання**

Розробити програму, що демонструє використання класу, який містить два методи роботи з текстовою інформацією:

– видалення з тексту символів ',' та запис результату до файлу "OUTPUT1.TXT";

– створення з літер тексту псевдографічного зображення



та запис результату до файлу "OUTPUT2.TXT".

Текст для передавання через аргументи у методи класу треба взяти з файлу "INPUT.TXT". Програмний код поділити на два файли – заголовний та програмний.

Розробимо програму на мові С++.

Заголовний файл "main.h":

#**include** <stdio.h> //*підключення бібліотеки вводу/виводу*

#**include** <string> //*підключення бібліотеки роботи з рядком*

**using namespace** std; //*застосування простору імен*

**class** CFWork //*оголошення класу роботи з файлами*

{

**public**: //*відкритий доступ до членів*

//*оголошення методу формування псевдографічного зображення*

**void** PGraph(string s);

**void** Format(string s); //*оголошення методу видалення ком з тексту*

};

Файл програми "main.cpp":

#**include** "main.h" //*підключення заголовного файлу з оголошенням класу*

//*визначення методу формування псевдографічного зображення*

**void** CFWork::PGraph(string s)

{

**int** a=0; //*оголошення та визначення цілоїзмінноїлічильника*

string st; //*оголошення об'єкта рядка*

**for** (int i=1; i<=4; i++) //*цикл за рівнем малюнка (4 – кількість рядків)*

{

**for**(int j=1; j<=((7-(2\*i-1))/2); j++) //*цикл додавання відступів*

st+=' '; //*додавання символа " "*

**for**(j=1; j<=(2\*i-1); j++) //*цикл додавання символу рядка*

st+=s.at(a++); //*додавання символу рядка*

**for**(j=1; j<=((7-(2\*i-1))+2); j++) //*цикл додавання відступів*

st+=' '; //*додавання символа " "*

**for**(j=1; j<=(2\*i-1); j++) //*цикл додавання символу рядка*

st+=s.at(a++); //*додавання символу рядка*

**for**(j=1; j<=((7-(2\*i-1))/2); j++) //*цикл додавання відступів*

st+=' '; //*додавання символа " "*

st+="\r\n"; //*додавання переходу на новий рядок*

}

//*оголошення змінної-покажчика на структуру роботи з файлом*

FILE \*f;

f=fopen("output2.txt", "w"); //*створення файлу для запису*

fprintf(f, "%s", st.c\_str()); //*запис рядка до файлу*

fclose(f); //*закриття файлу*

}

//*визначення методу видалення ком з тексту*

**void** CFWork::Format(string s)

{

string ns; //*оголошення об'єкта рядка*

**for**(**int** i=0; i<s.length(); i++) //*цикл перебору символів рядка*

**if**(s.at(i)!=',') //*якщо символ не є комою*

ns+=s.at(i); //*додати символ до нового рядка*

//*оголошення змінної-покажчика на структуру роботи з файлом*

FILE \*f;

f=fopen("output1.txt", "w"); //*створення файлу для запису*

fprintf(f, "%s", ns.c\_str()); //*запис рядка до файлу*

fclose(f); //*закриття файлу*

}

**void** main() //*оголошення та визначення головноїфункції*

{

//*оголошення змінної-покажчика на структуру роботи з файлом*

FILE \*f;

string st; //*оголошення об'єкта класу роботи з рядками*

**int** n=0;//*оголошення та визначення змінноїлічильника номера символа*

f=fopen("input.txt", "r"); //*відкриття файлу для читання*

**while** (!feof(f)) //*цикл до визначення кінця рядка*

{

st+=fgetc(f); //*формуємо рядок*

}

fclose(f); //*закриття файлу*

CFWork fw; //*оголошення об'єкта класу*

//*виклик методу формування псевдографічного зображення*

fw.PGraph(st);

fw.Format(st); //*виклик методу видалення ком з тексту*

}

Результатом роботи програми буде створення двох вихідних файлів.

**Питання для підготовки до захисту лабораторної роботи**

1) За допомогою якої методики мови С++ є можливість потокового вводу/виводу інформації файлу?

2) Для чого може використовуватись відокремлення оголошень класів, функцій та змінних у заголовний файл?

3) Як можна виконати виклик функції, що оголошена в іншому файлі?

4) За допомогою яких функцій виконується потокове зчитування та запис тексту у файл?

5) Яким чином виконується переміщення курсору файлу?