Лабораторна робота №4. одинарне ТА МНОЖИННЕ Успадкування. ФАЙЛОВІ ТА РЯДКОВІ ПОТОКИ

МЕТА РОБОТИ: засвоїти основні принципи та поняття успадкування, здобути практичні навики при роботі з «батьківськими», та «нащадковими» класами, навчитись контролювати файлові потоки, освоїти механізм запису і зчитування даних із файлу в різних режимах.

4.1. Програма роботи

4.1.1 Отримати завдання.

4.1.2. Написати програми відповідних класів, основну та відповідні допоміжні функції, згідно з вказівками до виконання роботи.

4.1.3. Підготувати власні коректні вхідні дані (вказати їх формат і значення) і проаналізувати їх.

4.1.4. Оформити електронний звіт про роботу та захистити її.

4.2. Вказівки до виконання роботи

4.2.1. Студент, згідно з індивідуальним номером, вибирає своє завдання.

4.2.2. Оголошення класу (структури), основну та відповідні допоміжні функції необхідно запрограмувати так, як це показано.

4.2.3. Власних вхідних даних необхідно підготувати не менше двох комплектів. Їхні значення мають бути коректними, знаходитися в розумних межах і відповідати тим умовам, які стосуються індивідуального завдання. Після проведення розрахунків отримані результати необхідно проаналізувати і занести до звіту.

4.2.4. Звіт має містити такі розділи:

* мету роботи та завдання з записаною умовою задачі;
* коди всіх використовуваних .h і .ccp файлів, а також пояснення до них;
* результати реалізації програми;
* діаграму класів та діаграму варіантів використання з поясненням;
* висновки, в яких наводиться призначення програми, обмеження на її застосування і можливі варіанти удосконалення, якщо такі є.

4.3. Теоретичні відомості

**Одинарне успадкування:**

Успадкування (inheritance) – це процес, завдяки якому об’єкт може набувати властивостей іншого об’єкта з додаванням до них ознак, властивих тільки йому.

Успадкування – один з трьох базових принципів ООП. Завдяки успадкуванню підтримується концепція ієрархії класів. Застосування ієрархії класів робить керованими великі потоки інформації.

Клас, властивості якого успадковуються, називається **базовим**, а клас, який успадковує властивості базового – **похідним**.

Для оголошення успадкування використовується така загальна форма:

class ім’я\_похідного\_класу : доступ ім’я\_базового\_класу {

... // елементи похідного класу

}

Тут доступ – одне з трьох ключових слів: public, private або protected. Специфікатор доступу визначає, як елементи базового класу успадковуються похідним класом. Якщо специфікатором доступу успадкованого базового класу є public, то всі відкриті члени базового класу стають відкритими і в похідному. Якщо специфікатором доступу успадкованого базового класу є private, то всі відкриті члени базового класу стають закритими в похідному. В обох випадках всі закриті члени базового класу залишаються закритими і недосяжними для похідного класу.

Якщо специфікатором доступу є private, то відкриті члени базового класу стають закритими в похідному, проте ці члени залишаються доступними для функцій-членів похідного класу.

Базовий клас, похідний клас або обидва можуть мати конструктори та/або деструктори. Якщо і у базового, і у похідного класів є конструктори та деструктори, то конструктори виконуються у порядку успадкування, а деструктори – у зворотньому порядку. Таким чином, конструктор базового класу виконується раніше, ніж конструктор похідного класу. Для деструкторів правильний зворотній порядок: деструктор похідного класу виконується раніше від деструктора базового класу. Синтаксис передавання аргументів з похідного в базовий клас такий:

ім'я\_похідного\_класу(список\_арг1): ім’я\_базового\_класу (список\_арг2) {

... //тіло конструктора похідного класу

}

**Множинне успадкування:**

Є два способи, завдяки яким похідний клас може успадковувати більше, ніж один базовий клас. По-перше, похідний клас може використовуватися як базовий для іншого похідного класу, створюючи багаторівневу ієрархію класів. В цьому випадку вихідний базовий клас є непрямим (inderect) базовим класом для іншого похідного класу. По-друге, похідний клас може прямо успадковувати більше, ніж один базовий клас. У такій ситуації комбінація двох або більше базових класів допомагає створенню похідного класу. Коли клас використовується як базовий для похідного, який, у свою чергу, є базовим для іншого похідного класу, конструктори цих трьох класів викликаються в порядку успадкування. Деструктори викликаються у зворотньому порядку. Якщо похідний клас напряму успадковує множину базових класів, використовується таке розширене оголошення:

class ім’я\_похідного\_класу:

спец\_доступу1 ім’я\_базового\_класу1,

спец\_доступу2 ім’я\_базового\_класу2,

. . .

спец\_доступуN ім’я базового\_класуN {

... // тіло класу

}

Тут ім’я\_базового класу1, ..., ім’я\_базового класуN – імена базових класів, спец\_доступу – специфікатор доступу, який може бути різним для кожного базового класу. Коли успадковується множина базових класів, конструктори використовуються зліва направо у порядку, що задається в оголошенні похідного класу. Деструктори виконуються у зворотньому порядку. Коли клас успадковує множину базових класів, конструкторам яких необхідні аргументи, похідний клас передає ці аргументи, використовуючи розширену форму оголошення конструктора похідного класу:

ім’я\_похідного\_класу (список\_арг):

ім’я\_базового\_класу1(список\_арг1),

ім’я\_базового\_класу2(список\_арг2),

. . .

ім’я\_базового\_класуN(список\_аргN) {

... // тіло конструктора похідного класу

}

Коли похідний клас успадковує ієрархію класів, кожний похідний клас повинен передавати попередньому базовому класу по ланцюжку необхідні аргументи.

**Файлові та рядкові потоки:**

Файлові операції введення – виведення даних можна реалізувати після внесення у програму заголовка <**fstream**>, у якому визначено всі необхідні для цього класи і значення. Створивши потік, його потрібно пов'язати з файлом. Це можна зробити за допомогою функції **open**(), причому у кожному з трьох потокових класів є своя функція-член **open**(). Їх прототипи мають такий вигляд:

**void** **ifstream**::**open**(**const** **char** \**filename*,

**ios**::**openmode** *mode* = **ios**::**in**);

**void** **ofstream**::**open**(**const** **char** \* *filename*,

**ios**::**openmode** *mode* = **ios**::**out** | **ios**::**trunc**);

**void** **fstream**::**open**(**const** **char** \**filename*,

**ios**::**openmode** *mode* = **ios**::**in** | **ios**::**out**);

У цих записах елемент *filename* означає ім'я файлу, яке може містити специфікатор шляху, який вказує доступ до нього. Елемент *mode* називається специфікатором режиму, який визначає спосіб відкриття файлу. Він повинен приймати одне або декілька значень перерахунку **openmode**, який визначено у класі **ios**:

* **ios::арр** – приєднує до кінця файлу усі дані, що виводяться;
* **ios::ate** – пошук потрібних даних починатиметься з кінця файлу;
* **ios::binary** – відкриває файл у двійковому режимі;
* **ios::in –** забезпечує відкриття файлу для введення даних;
* **ios::out –** забезпечує відкриття файлу для виведення даних;
* **ios::trunc** – призводить до руйнування вмісту файлу.

Декілька значень перерахунку **openmode** можна об'єднувати за допомогою логічного додавання (АБО).

**За замовчуванням усі файли відкриваються в текстовому режимі.**

Щоб закрити файл, використовується функція – член **close**().

4.4. Зразок виконання роботи

**Код програми:**

**Результат виконання програми:**

Рис. 1. Вигляд вікна в результаті компіляції програми

4.5. Індивідуальні завдання

1. Створити класи транспортних засобів: АВТОМОБІЛЬ, ВАНТАЖІВКА, ПАРОПЛАВ і ЛІТАК. Створити з них ієрархії класів. В основу ієрархії покласти клас ТРАНСПОРТНИЙ ЗАСІБ зі спільними для усіх цих класів елементами. Визначити функції виведення, успадковані методи, конструктори та деструктори. Інформацію про класи вивести у файл, використовуючи файлові потоки.
2. Створити ієрархію класів СПОРТИВНА ГРА і ФУТБОЛ. Визначити функції виведення, успадковані методи, конструктори та деструктори. Інформацію про класи вивести у файл, використовуючи файлові потоки.
3. Створити ієрархію класів НОСІЙ ДАНИХ – має обсяг (Гбайт), ВІНЧЕСТЕР, ОПТИЧНИЙ ДИСК – додатково мають кількість циліндрів, доріжок, секторів та марку. В основу ієрархії покласти клас НОСІЙ ДАНИХ. Визначити успадковані методи, конструктори та деструктори. Перевантажити операції потокового введення – виведення. Інформацію про класи вивести у файл, використовуючи файлові потоки.
4. Створити ієрархію класів ОСОБА, СТУДУНТ І СТУДЕНТ – ДИПЛОМНИК. Перевизначити виведення у потік і введення з потоку, операцію присвоювання через відповідні функції базового класу. Визначити успадковані методи. Інформацію про класи вивести у файл, використовуючи файлові потоки.
5. Створити ієрархію класів ПРОЦЕСОР – має марку і тактову частоту (ГГц), ЖОРСТКИЙ ДИСК – має обсяг(Гбайт), КОМП’ЮТЕР – має модель, ціну. В основу ієрархії покласти клас КОМП’ЮТЕР. Визначити функції виведення, успадковані методи, конструктори та деструктори. Інформацію про класи вивести у файл, використовуючи файлові потоки.
6. За допомогою множинного успадкування визначити клас КОЛО, який вписаний в КВАДРАТ. Класи КОЛО та КВАДРАТ повинні мати методи для обчислення площі та виводу інформації про фігури. Клас КОЛО повинен містити метод для обчислення довжини кола, клас КВАДРАТ – для обчислення периметра квадрата. Інформацію про класи вивести у файл, використовуючи файлові потоки.
7. За допомогою множинного успадкування визначити клас Прямокутник, вписаний у Коло. Класи Коло та Прямокутник повинні мати методи для обчислення площі та виводу інформації про фігури. Клас Коло повинен містити метод для обчислення довжини кола, клас Прямокутник – для обчислення периметра прямокутника. Інформацію про класи вивести у файл, використовуючи файлові потоки.
8. За допомогою множинного успадкування визначити клас Сфера, вписана у Куб. Класи Сфера та Куб повинні мати методи для обчислення об’єму, площі поверхні та виводу інформації про фігури. Інформацію про класи вивести у файл, використовуючи файлові потоки.
9. За допомогою множинного успадкування визначити клас Коло, вписане у Ромб. Класи Коло та Ромб повинні мати методи для обчислення площі та виводу інформації про фігури. Клас Коло повинен містити метод для обчислення довжини кола. Інформацію про класи вивести у файл, використовуючи файлові потоки.
10. Створити ієрархію класів ОСОБА і СЛУЖБОВЕЦЬ. Перевизначити виведення у потік і введення з потоку. Визначити успадковані методи. Інформацію про класи вивести у файл, використовуючи файлові потоки.
11. За допомогою множинного успадкування визначити клас ПРЯМОКУТНИЙ ТРИКУТНИК, вписаний у КОЛО. Класи КОЛО та ПРЯМОКУТНИЙ ТРИКУТНИК повинні мати методи для обчислення площі та виводу інформації про фігури. Клас КОЛО повинен містити метод для обчислення довжини кола, клас ПРЯМОКУТНИЙ ТРИКУТНИК – для обчислення периметра прямокутника. Інформацію про класи вивести у файл, використовуючи файлові потоки.
12. Створити ієрархію класів СПОРТСМЕН, ВИКЛАДАЧ та ТРЕНЕР. Визначити функції виведення, успадковані методи, конструктори та деструктори. Інформацію про класи вивести у файл, використовуючи файлові потоки.
13. Створити ієрархію класів СТУДЕНТ, АКТОР та УЧАСНИК КВН. Визначити функції виведення, успадковані методи, конструктори та деструктори. Інформацію про класи вивести у файл, використовуючи файлові потоки.
14. Створити ієрархію класів ІГРАШКА, МАШИНА та М’ЯКА ІРАШКА. Визначити функції виведення, успадковані методи, конструктори та деструктори. Інформацію про класи вивести у файл, використовуючи файлові потоки.
15. Описати клас ФОРТЕЦЯ. Для початку, напишіть клас СТРІЛЕЦЬ, який буде містити два поля: назва амуніції та її кількість. Він може виконати дію "постріл" і витратити для цього одиницю амуніції, повідомляючи в консоль, як вона називалась. А також може здійснити дію "дати амуніцію", яка встановлюватиме назву і кількість амуніції, отримуючи ці дані через параметри. Стрілець має три дочірні класи: ЛУЧНИК, РЕЙНДЖЕР та КАТАПУЛЬТА, які заміщають дію "постріл", доповнюючи повідомлення в консоль інформацією про те, хто саме з них вистрілив. Перші два повинні стріляти стрілами, а катапульта – каменями. Від перших двох породжується клас СТРІЛКОВА ВЕЖА, у якої запас амуніції має бути один (спільний) для обох: лучника і рейнджера. Від класів СТРІЛКОВА ВЕЖА і КАТАПУЛЬТА має походити клас ФОРТЕЦЯ. Лише в її конструкторі мають поставлятись стріли для стрілкової вежі і камені для катапульти. Додайте до фортеці ще кілька полів - товщина стін, стан воріт у відсотках, і т.д. – по вашому бажанню та фантазії. Більше ніяких допоміжних методів чи об'єктів створювати не потрібно. В кожного з полів чи методів у цій ієрархії має бути встановлений максимальний рівень захисту, і успадкування має відбуватись по найвищо-допустимому по захисту специфікаторові доступу. Але таким, щоб в main() можна було виконати наступне:

* створити об'єкт фортеці;
* вистрілити тричі лучником, двічі рейнджером і раз з катапульти.

Отже, всього має бути 6 класів, фінальним серед них є клас ФОРТЕЦЯ, лише один об'єкт якої буде створено. Зверніть увагу – 6 класів описано, всі використовуються. Але об'єкт у всій програмі один. Коли створюватиметься фортеця, доставте в стрілкову вежу лише 4 стріли, тоді буде явно видно, що на другий постріл рейнджеру стріл не вистачить, оскільки лучник перед тим витратив 3 із 4-х стріл. Інформацію про класи вивести у файл, використовуючи файлові потоки.

4.6. Контрольні запитання

1. Що таке механізм успадкування?
2. Які переваги використання успадкування?
3. Опишіть синтаксис використання успадкування.
4. Чи можуть функції похідного класу використовувати закриті члени базового класу?
5. Які типи успадкування Ви знаєте?
6. Який порядок виклику конструкторів та деструкторів при успадкуванні?
7. Що таке заміщення методу?
8. Які різновиди ієрархій Ви знаєте?
9. Що таке невизначеність при успадкуванні?
10. Наведіть приклад множинного успадкування?
11. Наведіть синстаксис множинного успадкування для мови С++.
12. Який за замовчуванням модифікатор доступу використовується при успадкуванні?
13. Як можна уникнути невизначеності?
14. Чи є обмеження у кількості рівнів у ієрархії успадкування? Чому?
15. Охарактеризуйте модифікатори доступу при успадкуванні.
16. Як відбувається зв’язування і відокремлення потоку з файлом?
17. Який потік необхідно відкрити для роботи з файловими операціями?
18. Яка функція використовується для відкриття і закриття файлу і які параметри вона має?
19. В якому режимі відкриваються файли за замовчуванням?
20. Як перевірити чи відкритий потік і файл?
21. Яка функція використовується для отримання кількості зчитаних символів за останню операцію?
22. Як працює оператор >> при зчитуванні рядків?
23. Яка різниця між функцією get() i getline()?
24. Яка функція використовується для запису вмісту файлових буферів на диск і як вона працює?
25. Як можна перевірити статус введення-виведення?
26. Які функції використовуються для отримання доступу до вмісту файлу в довільному порядку (не послідовно байт за байтом)?