Лабораторна робота №6. Шаблони функцій і класів

МЕТА РОБОТИ: навчитись створювати та використовувати шаблони функцій та узагальнені класи.

6.1. Програма роботи

6.1.1. Отримати завдання.

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

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

6.1.4. Оформити електронний звіт про роботу та захистити її.

6.2. Вказівки до виконання роботи

6.2.1. Студент, згідно з індивідуальним номером, вибирає своє завдання з розд. 6.5 і записує його до звіту.

6.2.2. Оголошення класу (структури), основну та відповідні допоміжні функції необхідно запрограмувати так, як це показано у розд. 6.4.

6.2.3. 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.

6.2.4. 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.

6.3. Теоретичні відомості

Шаблонні функції – узагальнене визначення функції, за яким компілятор автоматично згенерує код конкретного екземпляра функції. Шаблонні функції інакше називаються параметризованими або template – функціями. Використовуються у випадку, коли необхідно виконати обчислення для декількох різних типів даних за одним в тим самим алгоритмом.

Шаблонні функції оголошують за допомогою кваліфікатора template:

template <class T1 /\*,….., class Tn\*/>

тип\_функції ім’я\_функції (T1 t1 /\*,….., Tn tn, параметри інших типів \*/);

У кутових дужках після слів class вказуються формальні параметри типів. Замість слова class можна використовувати typename. Параметри типів можуть бути використані для оголошень формальних параметрів та локальних даних функції.

Розглянемо для прикладу функцію, яка повертає максимум з двох своїх параметрів:

// шаблон функції. Type – це параметр шаблону,

// який є типом параметрів

template <typename Type>

Type maxi (Type a, Type b)

{ return (a < b)? b : a; }

int main( )

{

int m = 123, n = 10;

double x = 1.0e-2, y = 1.0e+2;

char v = 'A', w = 'B';

cout <<"integer after max:"<< maxi (m, n)<< endl;

cout <<"double after max:"<< maxi (x, y)<< endl;

cout <<"char after max:" << maxi (v, w) << endl;

return 0;

}

**Виклики шаблонних функцій:**

Під час виклику шаблонної функції її формальні параметри типів набувають значення відповідних фактичних аргументів. Якщо фактичний аргумент є об’єктом класу, то, як правило, виникає необхідність у перевантаженні деяких операцій.

Порядок виклику функцій:

1. знайти нешаблонну функцію, параметри якої мають типи, вказані при виклику;
2. якщо функцію не знайдено, то знайти шаблон з точною відповідністю параметрів (для шаблонних параметрів перетворення типів не відбувається);
3. якщо жоден шаблон не забезпечує точної відповідності типі, то розглянути звичайні функції на предмет можливого неявного перетворення типів.

Шаблонний клас – це узагальнене визначення множини класів, з якого компілятор автоматично згенерує код конкретного класу. Шаблонні класи інакше називаються парметризованими або template – класами. Використовуються у випадку, коли програма будується на основі декількох класів з одинаковою структурою, які відрвзняються типами даних або типами параметрів методів. Шаблонні класи С++ реалізують один із варіантів поліморфізму – поліморфізм типів.

Оголошення шаблонного класу:

template <список формальних параметрів шаблону>

class ім’я\_класу

{

протокол класу, який використовує параметри шаблону;

};

Параметри шаблону оголошуються одним із двох способів:

**Відповідність параметрів шаблону класу**

|  |  |
| --- | --- |
| **Формальний параметр** | **Фактичний аргумент** |
| тип ідентифікатор | константа відповідного типу (цілий, переліковий, вказівник, посилання або вказівник на елемент класу). Значення дійсних та класових типів не можуть бути аргументами. Дозволяється вказівник на класовий тип. Константний рядок символів не може бути аргументом шаблону. |
| сlass ідентифікатор параметризованого типу | ідентифікатор будь – якого типу |

Шаблонний клас з двома можливими способами оголошення формальних параметрів:

template <class T, int k>

class A {

T \*p;

int size;

public:

A (int n);

~A ( );

//…..

};

**Статичні елементи шаблонних класів**

Поля статичних даних є спільними для усіх об’єктів конкретного екземпляра шаблонного класу (об’єктів з одинаковими значеннями аргументів шаблону).

Статичні дані визначаються у тому самому файлі, де було зроблено оголошення шаблонного класу.

Наприклад:

template <class T>

class A {

public:

static int x;

static T y;

public:

void print ( ) { cout<<x<< ‘ ’<<y<<endl; }

void inc ( ) { x++; y++; }

};

template <class T> int A<T>::x=5;

template <class T> T A<T>::y=7;

void main ( )

{

A<int> a;

a.print ( ); // 5 7

cout<<A<int>::x<< ‘ ’<<A<int>::y<<endl; // 5 7

a/inc ( );

a.print ( ); // 6 8

cout<<A<int>::x<< ‘ ’<<A<int>::y<<endl; // 6 8

A<float> b;

b.print ( ); // 5 7

cout<<A<int>::x<< ‘ ’<<A<int>::y<<endl; // 6 8

cout<<A<float>::x<< ‘ ’<<A<float>::y<<endl; // 5 7

}

6.4. Індивідуальні завдання

**Завдання 1**

1. Напишіть шаблон функції, що завжди повертає свій аргумент, помножений на два.
2. Створити шаблонну функцію для заміни одного елементу масиву на інший.
3. Створити шаблонну функцію для знаходження максимального значення елементів масиву.
4. Створити шаблонну функцію для знаходження елементу у впорядкованому масиві.
5. Створити шаблонну функцію для знаходження мінімального значення елементів масиву.
6. Створити шаблонну функцію для знаходження середнього арифметичного елементів масиву.
7. Створити шаблонну функцію, яка повертає модуль числа.
8. Створити шаблонну функцію для сортування елементів вибіркою.
9. Створити шаблонну функцію для сортування елементів за методом перестановок.
10. Створити шаблонну функцію для сортування елементів за методом бульбашки.
11. Створити шаблонну функцію для знаходження елементу в неупорядкованому масиві.
12. Створити шаблонну функцію для інверсії елементів масиву.
13. Створити шаблонну функцію для знаходження середнього геометричного елементів масиву.
14. Створити шаблонну функцію, яка при заданих значеннях елементів масиву і його розмірі виконує ротацію елементів: а[1]=a[0], a[2]= а[1], …, a[size-1]=a[size-2], a[0]= a[size-1].
15. Створити шаблонну функцію для пошуку в масиві заданого числа, яка має три параметра, два з них шаблонні, а один базового типу. Функція повертає індекс знайденого числа або -1 в випадку його відсутності у масиві. Аргументи функції є вказівними на масив, значення яке необхідно знайти, а також розмір масиву.

**Завдання 2**

1. Написати шаблонний клас для сортування одновимірного масиву за зростанням значень елементів.
2. Створити шаблонний клас для роботи із ФАЙЛАМИ даних різних типів. Перевизначити методи відкривання файла, запису даних у файл, зчитування даних з файла.
3. Створити шаблонний клас для сортування масиву по спаданню.
4. Створити шаблонний клас для перетворення заданого масиву таким чином, щоб кожний елемент масиву був отриманий як ехр кожного елемента (для типів int, float, long).
5. Створити шаблонний клас для видалення з текстового файлу заданої стрічки із заданої позиції.
6. Створити шаблонний клас для перетворення масивів з типу int в char.
7. Створити шаблонний клас для послідовного пошуку елемента масиву за заданим ключем. Якщо елемент знайдено, то на екран виводиться уся інформація, що відповідає заданому ключу.
8. Написати шаблонний клас, який знаходить контрольну суму елемента довільного типу. Контрольна сума – це кількість одиниць у машинному зображення заданого елемента.
9. Створити шаблонний клас для перетворення заданого масиву даних таким чином, щоб кожний елемент масиву був отриманий як квадрат кожного елемента (для типів int, float, long).
10. Створити параметризований шаблонний клас – множина, який призначений для збереження елементів і виконання операцій над ними. Реалізувати класичні операції над множинами – об’єднання, перетину, різниці, порівняння та операції введення та виведення.
11. Створити шаблонний клас vector з двома параметрами: перший є параметром типу, а другий – цілочисельним значенням. Визначити конструктор, деструктор, методи пошуку екстремуму, впорядкування, обчислення евклідової норми. Перевантажити операції +, =, [], <<, >> відповідно для додавання, присвоєння, контролю діапазону індексу, виведення та введення об’єктів.
12. Створити шаблонний клас – чергу, що базується на структурі зв’язаного списку динамічної пам’яті. Тип елементів черги визначається параметром шаблону. Передбачити функції для виконання таких операцій: занесення елемента у чергу; вилучення елемента з черги; виведення усіх елементів черги на екран; визначення кількості елементів черги.
13. Створити шаблонний клас для копіювання двовимірного динамічного масиву довільного типу. Передбачити операції копіювання за рядками, копіювання за стовпцями, копіювання вибраного рядка або стовпця, копіювання головної діагоналі. Розробити функції потокового введення і виведення масивів.
14. Створити шаблонний клас – стек, що базується на статичному масиві показчиків. Тип елементів стеку визначається параметром шаблону. Передбачити функції для виконання таких операцій: занесення елемента у стек; вилучення значення із вершини стеку; виведення усіх значень стеку на екран; повернення кількості елементів стеку.
15. Створити параметризований шаблонний клас бінарного дерева з методами – додати елемент до дерева проходження по дереву в спадному і висхідному порядку. Здійснити пошук за деревом.

6.5. Контрольні запитання

1. Яка функція називається шаблонною (узагальнюючою)?
2. З якого ключового слова визначення оголошення шаблонної функції?
3. Що таке спеціалізація чи конкретизація шаблонної функції?
4. Що таке породжена функція?
5. Скільки типів можна визначити в шаблонній функції і від чого це залежить?
6. Що таке безпосередня спеціалізація?
7. Чи можна перезавантажувати шаблонну функцію і якщо можна то як?
8. Як у шаблонних функціях використовують поряд із узагальненими стандартні параметри? Приклад.
9. Які обмеження є на застосування шаблонних функцій?
10. Як записати конкретний примірник узагальненого класу? Приклад.
11. Чи є функції-члени узагальненого класу узагальненими і як записують їх визначення? Приклад.
12. В яких випадках доцільно створювати узагальнені класи?
13. Чи можна створювати узагальнені класи з декількома узагальненими типами даних і як це зробити, якщо можна? Приклад.
14. Які обмеження накладаються при використанні аргументу в узагальнених класах, який не є типом?
15. Як визначити в шаблонних класах аргумент за замовчуванням для узагальненого типу? Приклад.
16. Як визначити в шаблонних класах аргумент за замовчуванням для неузагальненого типу? Приклад.
17. Що таке спеціалізація класу? Приклад.