10. Створення шаблонів функцій та шаблонів класів

Контрольні запитання:

- Як створити функцію-шаблон? В яких ситуаціях вона корисна?
- Як створити клас-шаблон? Що потрібно зробити якщо шаблоном є лише єдиний метод класу?
- Навіщо потрібні простори імен та що таке стандартний простір імен? Як його підключити та що робити коли не можна його підключати на весь файл програми?
- Як створити власний простір імен що містить власні математичні функції sin, cos, pow. Як їх коректно використати разом зі стандартними функціями?
- Створіть вкладені простори імен та функції з однаковими ідентифікаторами в них та функцію з таким самим ідентифікатором глобально. Як правильно використати ці функції використовуючи ключове слово using? Як без нього?

Завдання для аудиторної роботи:

В завданнях цього циклу створіть власний простір імен та в цьому просторі потрібні функції та класи.

1) Перепишіть функцію шаблон для пошуку максимуму двох чисел, так щоб вона працювала для всіх стандартних числових типів. Чи запрацює вона для рядків? Що потрібно зробити, щоб вона запрацювала і для типу Раціонального дробу з попередніх лекцій? (Вказівка: щось потрібно визначити для класу Раціональний дріб)
2) Написати функцію, що вводить масив цілих чисел доки не буде введений нуль та повертає результат через змінний аргумент та кількість елементів масиву повертається як результат роботи функції. Для невідомої заздалегідь кількості елементів потрібно робити реалізацію стеку. Створіть власну реалізацію класу шаблону Стек для будь-якого типу. Перевірте її роботу за допомогою стандартного класу Stack з STL для даної задачі та іншого типу чисел що вводяться.

Завдання для самостійної роботи:

3) Створити клас-шаблон BlackBox, який містить конструктор (порожній та від масиву (вказівника) будь-якого типу), метод push(), що

дозволяє додати елемент певного типу, та метод pop(), що видає та видаляє випадковий елемент, що вже міститься в класі та виключення, якщо BlackBox порожній, метод xpop(), що просто повертає випадковий елемент цього класу. Кількість елементів обмежена 100.

4) Створити клас-шаблон Mediana, який містить конструктор (порожній та від масиву (вказівника) будь-якого типу), що містить операції порівняння, метод push() який дозволяє додати елемент будь-якого типу, що містить операції порівняння, метод pop(int n), що видає та видаляє елемент за номером n за порядком, або виключення якщо n більше розміру всіх елементів та метод mediana(), що повертає медіану елементів цього класу. Кількість елементів обмежена 100.

5) Визначити клас Масив, який містить розмір масиву та відповідний масив даних довільного типу.

   Реалізувати в ньому методи сортування як для самого масиву та як статичні методи (inplace):

   а) обмінне сортування (метод бульбашки);

   б) обмінне сортування «Шейкер-сортування»;

   в) сортування за допомогою вибору (метод простого вибору);

   г) сортування вставками;

   д) сортування методом хешування (сортування з обчисленням адреси);

   е) сортування вставками (метод простих вставок);

   є) сортування бінарним злиттям;

   ж) сортування Шелла (сортування зі спадним кроком);

   з) швидке сортування;

   і) сортування купою.

6) Створіть клас раціональне число на базі шаблону пари для довільних типів знаменника та чисельника. Перевантажте методи для всіх арифметичних операцій та порівнянь (зокрема, остача від ділення – це ділення після якого видаляється ціла частина). Зробіть наступну спеціалізацію, якщо знаменник або чисельник – рядок: створюється рядок вигляду "чисельник /знаменник" з виключеннями на всі арифметичні операції, крім додавання (для нього – це конкатенація), але коректною роботою з порівнянням/введенням/виведенням/доступом.

7) Створіть клас рядок, що приймає у якості символу будь-який тип (зокрема інший рядок) та роздільник(того самого типу) - що відокремлює в запису ці символи. Методи класу:

- перезавантажте методи введення/виведення в/з консолі та в/з текстового файлу;
- введення та заміна роздільника;
- метод конкатенації (з додаванням між рядками роздільника);
- довжина рядку;
- злиття символів – тобто перетворення масиву символів на єдиний символ типу рядок;
- доступ до даного символу за квадратними дужками;
- видалення данного символу.

Забезпечити ініціювання помилки при неправильному введенні та роботі з рядками та роботі з файлами та спеціалізацію як звичайний рядок при символі типу char.

8) Визначити клас Інтервал с урахуванням включення/невключення країв та нескінченості на інтервалах, на базі шаблону пара. Якщо тип на одному з країв – рядок, то вважається що це відповідна нескінченість. Створити методи по знаходженню перетину і об'єднанню інтервалів, причому інтервали, що не мають спільних точок, перетинатися /об'єднуватися не можуть. Створить масив з n інтервалів та знайдіть їх спільний перетин.

9) Реалізуйте функцію sum(T* x, size_t n), яка рахує суму будь-якого масиву, що передається їй як аргумент. При цьому тип char сумується як символ, а тип вказівник вважається масивом розміру 1 та сумується утворюючи масив розміру n (нульові вказівники просто ігноруються в додаванні):

```
int v1[]  = { 1, 2, 3 }; // sum(v1,3) =6
double v2[] = { 1, 2, 3 };//sum(v2,3) =6.0
string v3[] = { "a", "bc", "def" };// sum(v3,3) ="abcdef"
char v4[]  = { 'a', 'b', 'c' }; // sum(v4,3) ="abc"
int* v5[] = { {1,4}, {2}, {3} }; // sum(v5,3) ={1,2,3}
```

10) Визначить клас Функція. Клас дозволяє задавати інтервал де шукається корінь та створювати функцію від ступнів дійсних чисел та від функцій косинус, корінь та логарифм. Створити методи для обчислення значення за формулою лівих прямокутників, за формулою правих прямокутників, формулою середніх прямокутників, по формулі трапецій, по формулі Сімпсона (параболічних трапецій).

Створіть метод для семплювання функції – задаються межі інтервалу та кількість семплів на інтервалі, обчислюються дискретні значення в даних точках і будується й виводиться таблиця, що містить пари точки - значення.