

Лабораторна робота № 11

Тема. Шаблонні класи.

Мета роботи. Поширити знання у шаблонізації (узагальненні) на основі вивчення шаблонних класів та створення власних шаблонних типів.

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:

- Кулик Данііл Ігорович
- НТУ “ХПІ” КІТ 102.8а
- Варіант 12

1.2 Загальне завдання

Модернізувати клас, що був розроблений у попередній роботі наступним шляхом:

– зробити його шаблонним;

– додати поле – шаблонний масив;

– видалити з аргументів існуючих методів масив, а замість цього використовувати масив-поле класу. Необхідно продемонструвати роботу програми як з використанням стандартних типів даних, так і типів, які створені користувачем.

2 ОПИС ПРОГРАМИ

2.1 Функціональне призначення

За допомогою цієї програми можна створити масив об'єктів, додавати та видаляти об'єкти, виводити вміст масиву на екран та вивід об'єкта по індексу. Також у цій програмі реалізоване зручне меню спілкування з користувачем.

2.2 Важливі фрагменти програми

На рисунку №1 зображено діаграми класів.

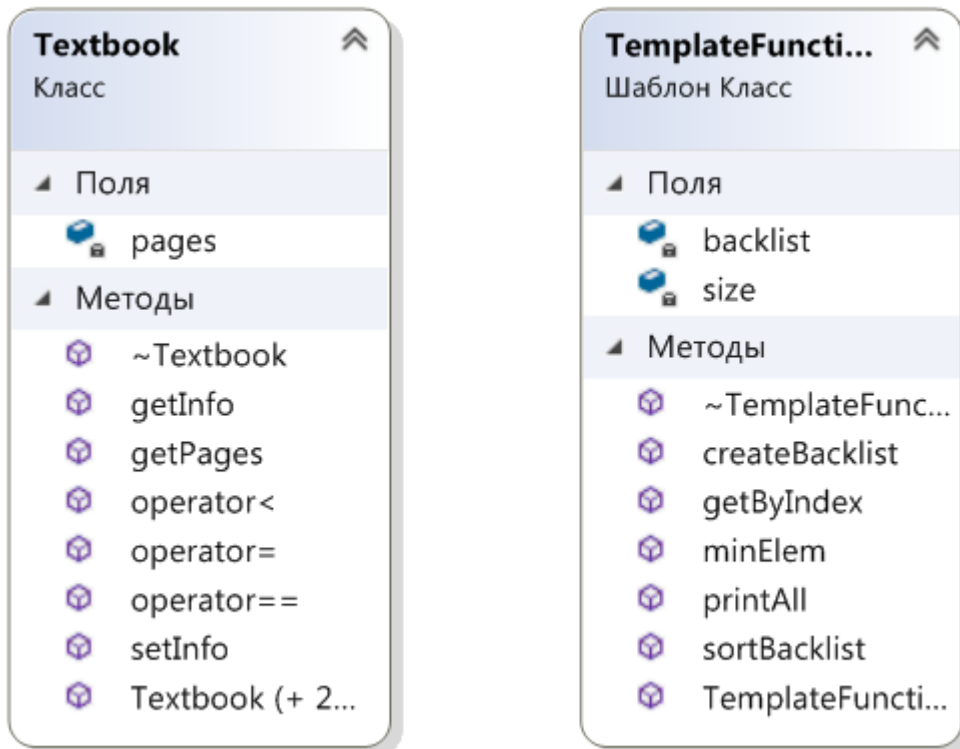


Рисунок №1 – діаграма класу Textbook та TemplateFunctions

Методи класу **Textbook**:

- **Textbook()** - Конструктор за замовчуванням ;
- **Textbook(int pages)** – конструктор з параметрами;
- **Textbook(const Textbook &obj)** – конструктор копіювання ;
- **void setInfo(int pages)** – заповнення поля **Textbook:: pages**;
- **int getPages()const** – читання значення поля **pages** (інші методи **get** роблять теж саме, але з іншими полями).
- **Textbook& operator = (Textbook &obj)** - перевантажений оператор присвоювання;
- **bool operator < (Textbook &obj)** - перевантажений оператор «менше»;
- **bool operator == (Textbook &obj)** - перевантажений оператор порівняння.

Методи класу `TemplateFunctions`:

- `TemplateFunctions()` - Конструктор за замовчуванням ;
- `void createBacklist(T* backlist, int size)` – функція, що дозволяє створити масив;
- `void printAll(T* arr, int size)` – функція, що виводить вміст масиву на екран;
- `void sortBacklist(T* arr, int size)` - функція, виконує сортування елементів масиву;
- `T minElem(T* arr, int size)` – функція, що визначає значення мінімального елементу масиву;
- `int getByIndex(T* arr, int size, T values)` – функція, що визначає індекс переданого елементу в масиві.

3 ВАРІАНТИ ВИКОРИСТАННЯ

Програма може бути використана для створення масиву об'єктів. Програма має методи додавання, видалення об'єктів, вивід об'єкта по індексу з масиву та усіх об'єктів на екран, читання та запис об'єктів масиву з файлу, а також сортування за одним із властивостей об'єкта. Меню робить роботу з цією програмою зручною.

1. Меню спілкування з користувачем зображено на рисунку №5

```
Enter size: 3
Enter number of pages:
228
322
160

Welcome to our menu list. What do you want?:
0 - Exit
1 - Print array
2 - Sort array
3 - Search by index
4 - Search by min values
```

Рисунок №5 – меню спілкування з користувачем

2. Результат виводу об'єктів після сортування зображено на рисунку №6.

```
Number of pages in 1 textbook is: 160
Number of pages in 2 textbook is: 228
Number of pages in 3 textbook is: 322

Welcome to our menu list. What do you want?:
0 - Exit
1 - Print array
2 - Sort array
3 - Search by index
4 - Search by min values
```

Рисунок №6 - результат виводу об'єктів масиву після сортування

ВИСНОВКИ

В інтегрованому середовищі VisualStudio розроблена програма мовою C++. Засоби налагодження дозволяють за допомогою меню спілкування створити масив об'єктів та змінювати його за допомогою методів класів.