Programowanie, Laboratorium 6.

Szablony wspomagają bezpośrednio programowanie z użyciem typów jako parametrów. Mechanizm szablonów umożliwia używanie typów i wartości jako parametrów m.in. definicji funkcji oraz klas. Szablony można łatwo reprezentować i łączyć ze sobą ogólne koncepcje programistyczne.

Szablony umożliwiają definiowanie funkcji uniwersalnych, bez podawania typów. Kompilator C++ podczas kompilacji tworzy funkcje odpowiednich typów. Definiując szablony funkcji, zmniejszamy liczbę funkcji, a program może używać tej samej nazwy dla funkcji realizujących konkretną operację, niezależnie od typu zwracanej wartości i typów parametrów. Wszystkie szablony funkcji zaczynają się od słowa kluczowego template poprzedzającego zawartą w nawiasach ostrokątnych < > listę parametrów formalnych wzorca funkcji. Każdy z tych parametrów, reprezentujący typ, musi być poprzedzony słowem class (lub typename), tak jak tutaj:

template<class T> lub template <typename T>

Litera T oznacza uniwersalny typ szablonu. Parametry formalne w definicji wzorca wykorzystywane są (podobnie jak argumenty typów wbudowanych i zdefiniowanych przez użytkownika) do określenia typów argumentów funkcji i typu zwracanej przez nią wartości oraz do deklarowania przez nią zmiennych. Po definicji wzorca występuje definicja funkcji. Słowo kluczowe class (lub typename) wykorzystywane do określenia typów parametrów wzorca funkcji oznacza tutaj „każdy typ wbudowany lub zdefiniowany przez użytkownika”.

Szablony w języku C++ są podstawą programowania ogólnego (ang. generic programming). Jest to technika programowania skupiona na projektowaniu, implementowaniu i stosowaniu ogólnych algorytmów. Algorytm może przyjmować argumenty różnego typu, pod warunkiem że spełniają one wymagania tego algorytmu.

Ćwiczenie 1

Napisz program zawierający prosty szablon funkcji *max()*, która zwraca większą z dwóch wartości typu T (użyj typu int, float oraz string).

Ćwiczenie 2

Napisz program zawierający szablon funkcji *wypiszTablice()*, który wyświetla zadeklarowane w programie elementy tablicy. W definicji szablonu typ T oznacza typ tablicy, a T1 typ licznika tablicy. Stworzona za pomocą szablonu funkcja wypisuje elementy tablic typów int, double oraz char

Ćwiczenie 3

Napisz program zawierający szablon klasy dla tablic, który definiuje metody służące do obliczania sumy dla elementów przechowywanych w tablicy

```
class SumowanieTablic
{
        public:
        SumowanieTablic(int rozmiar);
        T1 suma(void);
        void wypisz_tablice(void);
        int dodaj_element(T);
        private:
        T* dane;
        int rozmiar;
        int indeks;
};
```

Ćwiczenie 4

Napisz program, który z wykorzystaniem instrukcji warunkowej if oblicza pierwiastki równania kwadratowego ax2 + bx + c = 0, gdzie zmienne a, b, c to liczby rzeczywiste wprowadzane z klawiatury. Dla zmiennych a, b, c, x1 oraz x2 należy przyjąć format wyświetlania ich z dokładnością do dwóch miejsc po kropce

Ćwiczenie 5

Napisz zgodnie z zasadami programowania obiektowego program, który wczytuje z klawiatury imię i nazwisko, zapisuje te dane do pliku tekstowego dane.txt, a następnie odczytuje je z niego i wyświetla na ekranie komputera. Klasa powinna zawierać trzy metody:

- czytaj_dane() — wczytuje z klawiatury imię i nazwisko;
- zapisz_dane_do_pliku() — zapisuje imię i nazwisko do pliku tekstowego dane.txt;

- czytaj_dane_z_pliku() — odczytuje dane z pliku tekstowego dane.txt i wyświetla je na ekranie komputera.

Ćwiczenie 6

Napisz zgodnie z zasadami programowania obiektowego program, który tablicę a o wymiarach 10×10 odwraca symetrycznie w osi przekątnej diagonalnej. Klasa powinna zawierać cztery metody:

- czytaj_dane() — tworzy tablicę a o wymiarach 10×10;
- przetwórz_dane() — przepisuje elementy tablicy a o wymiarach 10×10 do tablicy b o tej samej wielkości;
- zapisz_dane_do_pliku() — zapisuje tablicę b o wymiarach 10×10 do pliku;
- czytaj_dane_z_pliku() — odczytuje tablicę c o wymiarach 10×10 z pliku i wyświetla ją na ekranie.