

KURS PROGRAMOWANIA W C++

STOS I KOLEJKA

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zadanie 1

Poniżej przedstawiono definicję szablonu klasy reprezentującego listę jednokierunkową. Dopisz definicje wszystkich zadeklarowanych w tym szablonie metod.

```
template <typename T>
class Lista
{
    friend ostream & operator<< <T> (ostream &wy, const Lista<T> &lista);
protected:
    class Wezel // węzeł wewnętrzny listy
    {
        friend ostream & operator<< <T> (ostream &wy, const Wezel<T> &wezel);
protected:
        Wezel *nastepny;
        T wartosc;
public:
        Wezel (const T &wart, Wezel *nast=0);
        Wezel (const Wezel &wezel);
        ~Wezel ();
public:
        void wstaw (const T &wart, int poz);
        void usun (int poz);
        const T & szukaj (int poz);
    };
protected:
    Wezel *pocz; // wskaźnik na początkowy węzeł listy
    int dl; // długość listy (liczba węzłów wewnętrznych)
public:
    Lista ();
    Lista (const Lista &lista);
    virtual ~Lista ();
    Lista & operator= (const Lista &lista);
public:
    int dlugosc () const;
    void wstaw (T wart, int poz);
    void usun (int poz);
    void usunWszystko ();
    const T & operator[] (int poz);
};
```

Następnie na podstawie szablonu listy napisz definicję szablonów klas dla stosu i kolejki, stosując dziedziczenie niepubliczne.

Napisz zestaw kilku programów testowych do sprawdzenia, czy zdefiniowane klasy działają poprawnie. Przetestuj je dla danych typu `int`, `string`, `string*` i `char*`. Jeśli zaistnieje potrzeba, to napisz specjalizowane wersje szablonu dla wybranych typów.

Zadanie 2

Zdefiniuj szablony klas dla stosu i kolejki, stosując kompozycję i wykorzystując szablon `vector<>`. Zamortyzowany czas działania operacji wstawiania i usuwania elementu z kolejki i stosu powinien być stały (zasymuluj działanie kolejki za pomocą dwóch stosów).

Napisz zestaw kilku programów testowych do sprawdzenia, czy zdefiniowane klasy działają poprawnie. Przetestuj je dla danych typu `int`, `string`, `string*` i `char*`. Jeśli zaistnieje potrzeba, to napisz specjalizowane wersje szablonu dla wybranych typów.

Uwaga

Porównaj czasy działania zaprogramowanych przez Ciebie w zadaniach 1 i 2 stosów i kolejek. Przetestuj ich działanie dla dużej liczby operacji.