

KURS JĘZYKA C++

LISTA NAPISÓW

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zadanie.

Zdefiniuj klasę `lista`, która będzie opakowywać listę jednokierunkową. Sama lista ma być homogeniczną strukturą zbudowaną z węzłów — obiektów klasy `lista::wezel` (zagnieżdżona definicja klasy).

```
class lista {
    class wezel {
        wezel *nast; // następny węzeł w liście
        string napis; // napis przechowywany w węźle
        // ...
    public:
        friend ostream& operator<< (ostream &wy, const wezel &wzl) {
            if (wzl.nast) return wy << wzl.napis << ", " << *wzl.nast;
            else return wy << wzl.napis;
        }
    };
    wezel *poczatek; // wskaźnik na początek listy
public:
    lista();
    lista(const lista &lst);
    lista(lista &&lst);
    lista(initializer_list<string> lst);
    lista& operator= (const lista &lst);
    lista& operator= (lista &&lst);
    ~lista();
    // ...
public:
    friend ostream& operator<< (ostream &wy, const lista &lst) {
        if (lst.poczatek) return wy << "(" << *lst.poczatek << ")";
        else return wy << "()";
    }
};
```

Całą funkcjonalność listy zaprogramuj w węźle w prosty rekurencyjny sposób; metody listy mają tylko aktywować odpowiadające im metody w węźle początkowym. Funkcjonalność listy ma być ograniczona do: wstawienia do listy nowego napisu na zadaną pozycję (metoda `void wstaw (const string &napis, int poz)`), odczytania napisu z zadanej pozycji (metoda `const string& czytaj (int poz)`), usunięcia z listy napisu z zadanej pozycji (metoda `void usun (int poz)`) oraz ustalenia liczby wszystkich elementów na liście (metoda `int rozmiar()`).

W liście zaimplementuj kopiowanie i przenoszenie oraz inicjalizację za pomocą listy wartości `initializer_list<string>`.

Dodatkowo napisz program rzetelnie testujący wszystkie zdefiniowane w liście metody oraz mechanizmy inicjalizacji i kopiowania.

Uwaga.

Podziel program na pliki nagłówkowe i źródłowe.