

KURS JĘZYKA C++

GRAFY

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zadanie.

Zdefiniuj klasę `Graf` reprezentującą graf prosty. W grafie tym zbiór wierzchołków $V = \{0, 1, \dots, n-1\}$ ma być określony na etapie konstrukcji i nie zmieniać się do końca egzystencji obiektu. Krawędzie w takim grafie można jednak dynamicznie dokładać albo usuwać. Klasa ta powinna być wyposażona w konstruktor kopiujący, operator przypisania kopiującego i wirtualny destruktor.

```
class Graf
{
// ...
public:
    Graf (int n)
        throw(ujemna_liczba_wierzchołkow);
    Graf (int n, vector<pair<int,int>> kraw)
        throw(ujemna_liczba_wierzchołkow, nieprawidłowa_krawedz);
    Graf (const Graf &g) throw();
    Graf & operator= (const Graf &g) throw();
    virtual ~Graf () throw();
// ...
};
```

W klasie `Graf` zdefiniuj operatory umożliwiające dołożenie nowej krawędzi oraz usunięcie istniejącej. Zaprogramuj także zaprzyjaźnione operatory do odczytania grafu ze strumienia wejściowego i zapisania grafu do strumienia wyjściowego.

```
class Graf
{
// ...
public:
    friend Graf & operator+= (pair<int,int> kraw)
        throw(nieprawidłowa_krawedz);
    friend Graf & operator-= (pair<int,int> kraw)
        throw(nieprawidłowa_krawedz);
    friend istream operator>> (istream &we, Graf &x);
    friend ostream operator<< (ostream &wy, const Graf &x);
};
```

Uzupełnij funkcjonalność grafu o przeglądanie w głąb i w szerz z możliwością wykonania jakiegoś zadania na odwiedzanych wierzchołkach — argumentem metod przeglądających powinien być wskaźnik do funkcji albo obiektu funkcyjnego, który realizuje jakieś zadanie.

Uzupełnienie.

Na koniec napisz program, który dla zadanego ze standardowego wejścia grafu wyznaczy jakieś drzewo rozpinające dla niego. W swoim programie posłuż się techniką zdobywania zasobów poprzez inicjalizację z wykorzystaniem obiektu `auto_ptr`.