

ELEMENTY JĘZYKA C++: pliki nagłówkowe i źródłowe, kompilacja i konsolidacja.

### 1. Prosty kalkulator 3

Napisz program zawierający funkcje `dodaj`, `odejmij`, `pomnoz` i `podziel`. Każda z nich powinna przyjmować dwie liczby całkowite i zwracać wynik działania. Utwórz prosty kalkulator korzystający z tych funkcji. Deklaracje funkcji umieść w pliku nagłówkowym, a ich kod źródłowy z pliku źródłowym, który będzie niezależny od reszty kodu kalkulatora.

### 2. Lista powiązana

Poniżej znajduje się kod `original.cpp` związany z listą powiązaną. Podziel kod na plik nagłówkowy oraz pliki źródłowe tak, aby nadawał się do wielokrotnego użycia.

```
#include <iostream>
using namespace std;

struct Node {
    Node *p_next;
    int value;};

Node* addNode (Node* p_list, int value) {
    Node *p_new_node = new Node;
    p_new_node->value = value;
    p_new_node->p_next = p_list;
    return p_new_node;}

void printList (const Node* p_list) {
    const Node* p_cur_node = p_list;
    while ( p_cur_node != NULL ) {
        cout << p_cur_node->value << endl;
        p_cur_node = p_cur_node->p_next;}}

int main () {
    Node *p_list = NULL;
    for ( int i = 0; i < 10; ++i ) {
        int value;
        cout << "Enter value for list node: ";
        cin >> value;
        p_list = addNode( p_list, value );}
    printList( p_list );}
```

**3. Kwestionariusz osobowy 2**

Korzystając z klasy `Name` (ćwiczenia 10) napisz program, który wczytuje ze standardowego wejścia listę imion i nazwisk, a potem wypisuje je na standardowe wyjście najpierw w kolejności alfabetycznej, a następnie w kolejności alfabetycznej inicjałów (w tym celu napisz dodatkowo własną funkcję porównującą, której użyjesz jako argumentu funkcji `sort`). Kod projektu podziel na plik nagłówkowy i pliki źródłowe. Plik nagłówkowy z deklaracją klasy:

```
class Name {
public:
    Name ();
    Name (const string &given,const string &family);
    string initials () const;
    friend bool operator< (const Name &first,const Name &second);
    friend ostream &operator<< (ostream &stream,const Name &Name);
    friend istream &operator>> (istream &stream,Name &Name);
private:
    string given,family;};
```

**4. Figury geometryczne**

Dana jest lista figur geometrycznych, będących obiektami wirtualnej klasy bazowej `Figure` (ćwiczenia 11), w następującym formacie:

```
circle red 5
rectangle blue 2 3
```

Napisz program, który wczytuje tę listę ze standardowego wejścia, a następnie wypisuje na standardowe wyjście tę samą listę posortowaną w kolejności rosnącego pola powierzchni. Przy każdej figurze powinno być dodatkowo wypisane jej pole powierzchni. Podziel kod na pliki nagłówkowe (osobne dla klasy bazowej `Figure` oraz każdej z klas pochodnych `Circle` i `Rectangle`) i pliki źródłowe (także osobne dla każdej z klas).

Zadań domowych wyjątkowo brak. Pytania można wysłać na adres: MDABROWSKI@FUW.EDU.PL.