Dzielenie programu na pliki (krok po kroku)

1. Deklaracje i definicje razem (punkt wyjścia)

```
class Liczba
{
  private:
    int wartosc;

  public:
    Liczba(int wartosc)
    {
      this->wartosc = wartosc;
    }

    int getWartosc()
    {
      return wartosc;
    }
};
```

2. Oddzielenie definicji od deklaracji

```
class Liczba
{
  private:
    int wartosc;

  public:
    Liczba(int wartosc);
    int getWartosc();
};


Liczba::Liczba(int wartosc)
{
  this->wartosc = wartosc;
}


int Liczba::getWartosc()
{
  return wartosc;
}
```

3. Deklaracje przeniesione do pliku nagłówkowego (.h lub .hpp), definicje przeniesione do pliku źródłowego (.cpp)

```
// plik Liczba.h

class Liczba
{
  private:
    int wartosc;

  public:
    Liczba(int wartosc);
    int getWartosc();
};


// plik Liczba.cpp

Liczba::Liczba(int wartosc)
{
  this->wartosc = wartosc;
}


int Liczba::getWartosc()
{
  return wartosc;
}
```

4. Dyrektywy zabezpieczające przed wielokrotnym przeczytaniem pliku nagłówkowego przez preprocesor

Cały plik nagłówkowy powinien zostać „opakowany” w dyrektywy pozwalające przeczytać go tylko jeden raz. Identyfikator (np. _LICZBA_H_) powinien być inny dla każdego pliku.

```
// plik Liczba.h

#ifndef _LICZBA_H_  // oznacza: czytaj dalej jeśli niezdefiniowane
#define _LICZBA_H_  // oznacza: zdefiniuj _LICZBA_H_

class Liczba
{
  private:
    int wartosc;

  public:
    Liczba(int wartosc);
    int getWartosc();
};

#endif              // koniec warunkowego przetwarzania
```

5. Dodanie do pliku źródłowego (.cpp) dyrektywy #include do pliku nagłówkowego (.h lub .hpp)

Plik źródłowy musi „widzieć” swoje deklaracje. Jeżeli dołączamy nasz plik, znajdujący się w tym samym katalogu, to nazwa pliku powinna znaleźć się w cudzysłowie.

```
// plik Liczba.cpp

#include ”Liczba.h”

Liczba::Liczba(int wartosc)
{
  this->wartosc = wartosc;
}


int Liczba::getWartosc()
{
  return wartosc;
}
```

6. Efekt końcowy

```
// plik Liczba.h

#ifndef _LICZBA_H_
#define _LICZBA_H_

class Liczba
{
  private:
    int wartosc;

  public:
    Liczba(int wartosc);
    int getWartosc();
};

#endif


// plik Liczba.cpp

#include ”Liczba.h”

Liczba::Liczba(int wartosc)
{
  this->wartosc = wartosc;
}


int Liczba::getWartosc()
{
  return wartosc;
}
```

7. Uwagi

Z reguły każda klasa w programie ma „swoje” dwa pliki (jeden źródłowy i jeden nagłówkowy).

Jeżeli dana klasa ma być używana w innym miejscu (np. w plik main.cpp), to należy tam dodać dyrektywę #include do **pliku nagłówkowego (.h lub .hpp)** naszej klasy. **Nie umieszczamy w kodzie dyrektyw #include do plików źródłowych (.cpp)** – teoretycznie jest to możliwe, ale rodzi sporo potencjalnych problemów i należy tego unikać.

Przy takich założeniach, podczas kompilacji należy wymienić wszystkie pliki **źródłowe (.cpp)**. Przykładowo – nasz program składa się z trzech plików:

- Liczba.h
- Liczba.cpp
- main.cpp

Kompilacja z poziomu linii poleceń będzie wyglądała następująco:

g++ Liczba.cpp main.cpp -o program.exe

Ewentualnie (w szczególnych przypadkach można uprościć sobie życie):

g++ *.cpp -o program.exe