Сучасні мови та об’єктно-орієнтоване програмування в ядерній фізиці

Інкапсуляція та наслідування об’єктів в C++. Абстрактні класи. Інтерфейси.

Р.В. Єрмоленко

Зміст

1. Інкапсуляція. Модифікатори доступу.
2. Об'явлення класу та реалізація його методів.
3. Поняття наслідування.
4. Множинне наслідування.
5. Віртуальні функції.
6. Абстрактні класи. Інтерфейси та їх призначення.
7. Поліморфізм

Література

- Роберт Мартин. Чистый Код. Создание, анализ и рефакторинг.
- Бьярне Стауструп. Программирование: принципы и практика использования C++
- Эккель Брюс. Философия C++
- Герберт Шилдт. C++. Базовый курс
- Стенли Липпман, Жози Лажойе. C++ для начинающих
- Стивен Прата. Язык программирования C++
- Bjarne Stroustrup - The C++ Programming Language
- Deitel H. M., Deitel P.J. - C++: How to Program
- Bjarne Stroustrup - Programming: Principles and Practice Using C++

Структура та клас

ІСТОРІЯ

СТРУКТУРИ В С ТА С++. РІЗНИЦЯ

```c
struct ParticleDefinition
{
    std::string name;
    float mass;
    float Px;
    float Py;
    float Pz;
    float E;
    bool stable;
    double lifetime;
};
```

СТРУКТУРА В С

```c
struct ParticleDefinition
{
    std::string name;
    float mass;
    float Px;
    float Py;
    float Pz;
    float E;
    bool stable;
    double lifetime;
    float getEnergy() { return E; }
};
```

СТРУКТУРА В С++

В С ++ поняття структури було розширено до класу, тобто існує додана можливість включення в структуру функцій-методів.

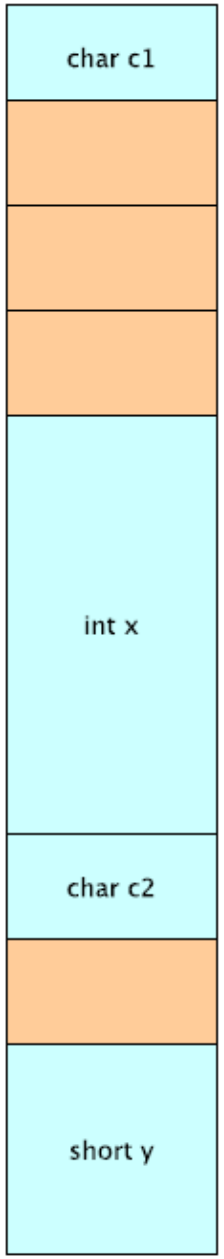
РОЗМІЩЕННЯ СТРУКТУР В ПАМ'ЯТІ. ВИРІВНЮВАННЯ

- В залежності від методу розміщення в памяті даних визначає швидкість доступу до них.
- На більшості сучасних архітектур процесорів доступ до даних відбувається швидше, якщо вони вирівняні.
- Наприклад, змінні типу int повинні знаходитися за адресами, кратними їх розміру (4 байта).

```
4
5   struct Data
6   {
7       char c1;
8       int x;
9       char c2;
10      short y;
11  };
12
13  int main()
14  {
15      std::cout << sizeof(Data);
16      return 0;
17  }
```

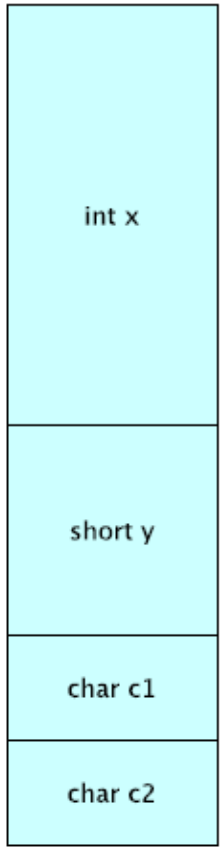
```
5   struct Data
6   {
7       char c1;
8       char c2;
9       short y;
10      int x;
11  };
12
13  int main()
14  {
15      std::cout << sizeof(Data);
16      return 0;
17  }
18
19
```

Не оптимально



12 байт

Оптимально



8 байт

Структура та клас в C++

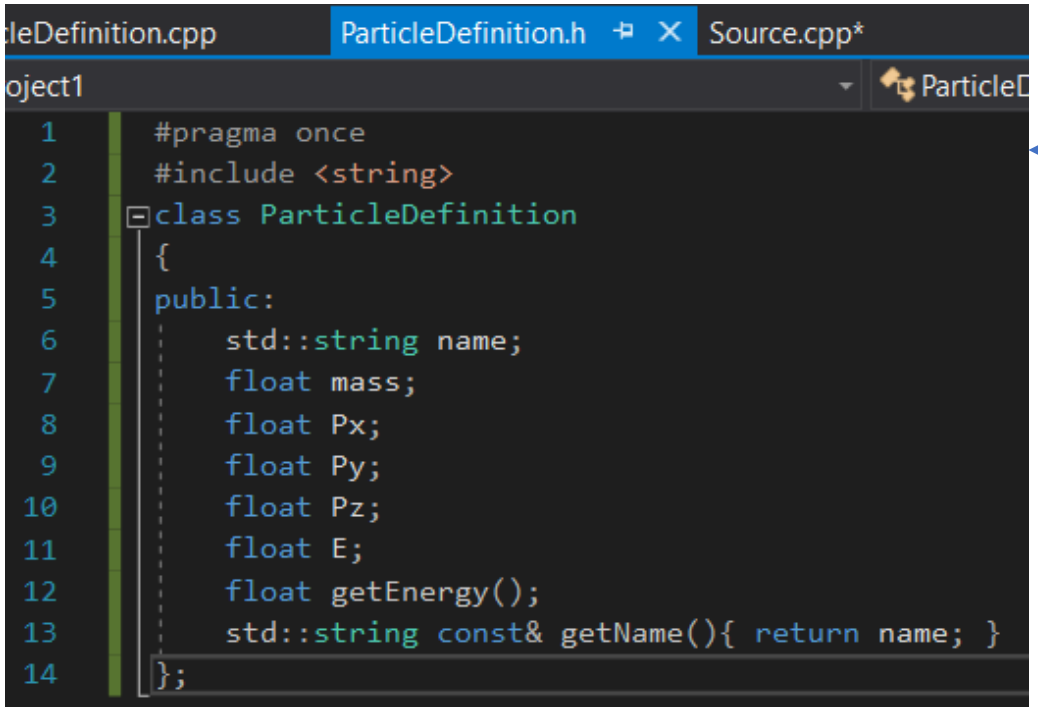
```cpp
struct ParticleDefinition
{
    std::string name;
    float mass;
    float Px;
    float Py;
    float Pz;
    float E;
    bool stable;
    double lifetime;
    float getEnergy() { return E; }
};
```

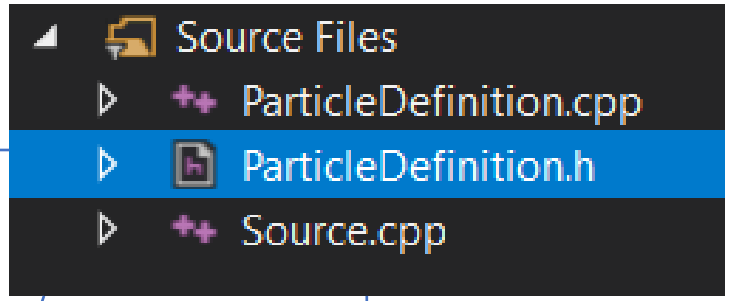
```cpp
class ParticleDefinition
{
public:
    std::string name;
    float mass;
    float Px;
    float Py;
    float Pz;
    float E;
    float getEnergy() {return E;}
};
```

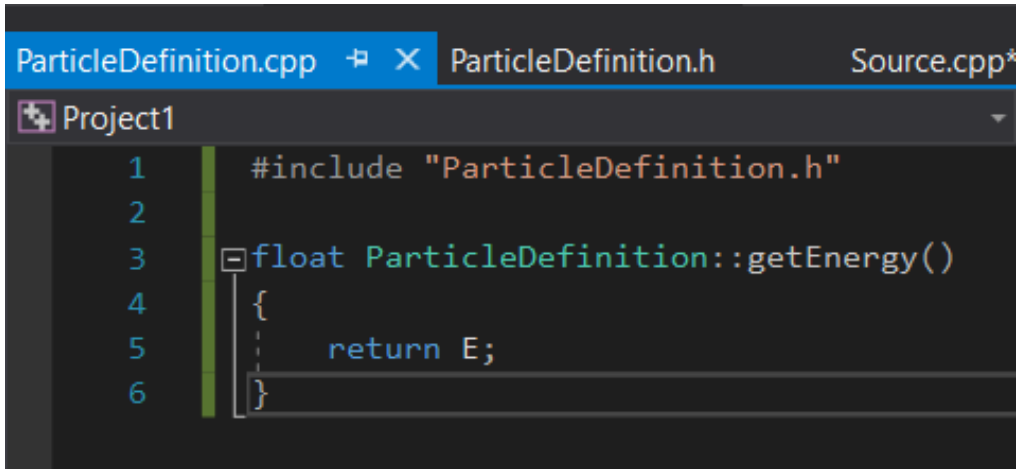
Основна відмінність структури від класу – наявність модифікатора доступу (принцип інкапсуляції) члени класу є закритими, а члени структури – відкритими.

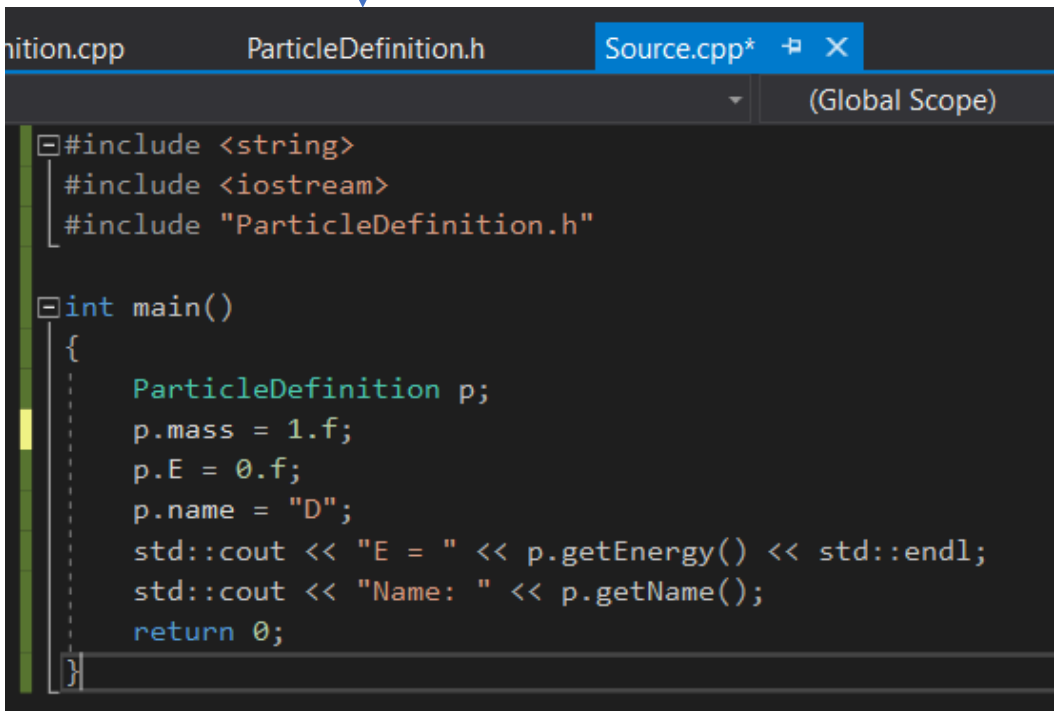
Заголовкові файли *.h та файли реалізації *.cpp

ВИКОРИСТАННЯ ФАЙЛІВ-ЗАГОЛОВКІВ *.h

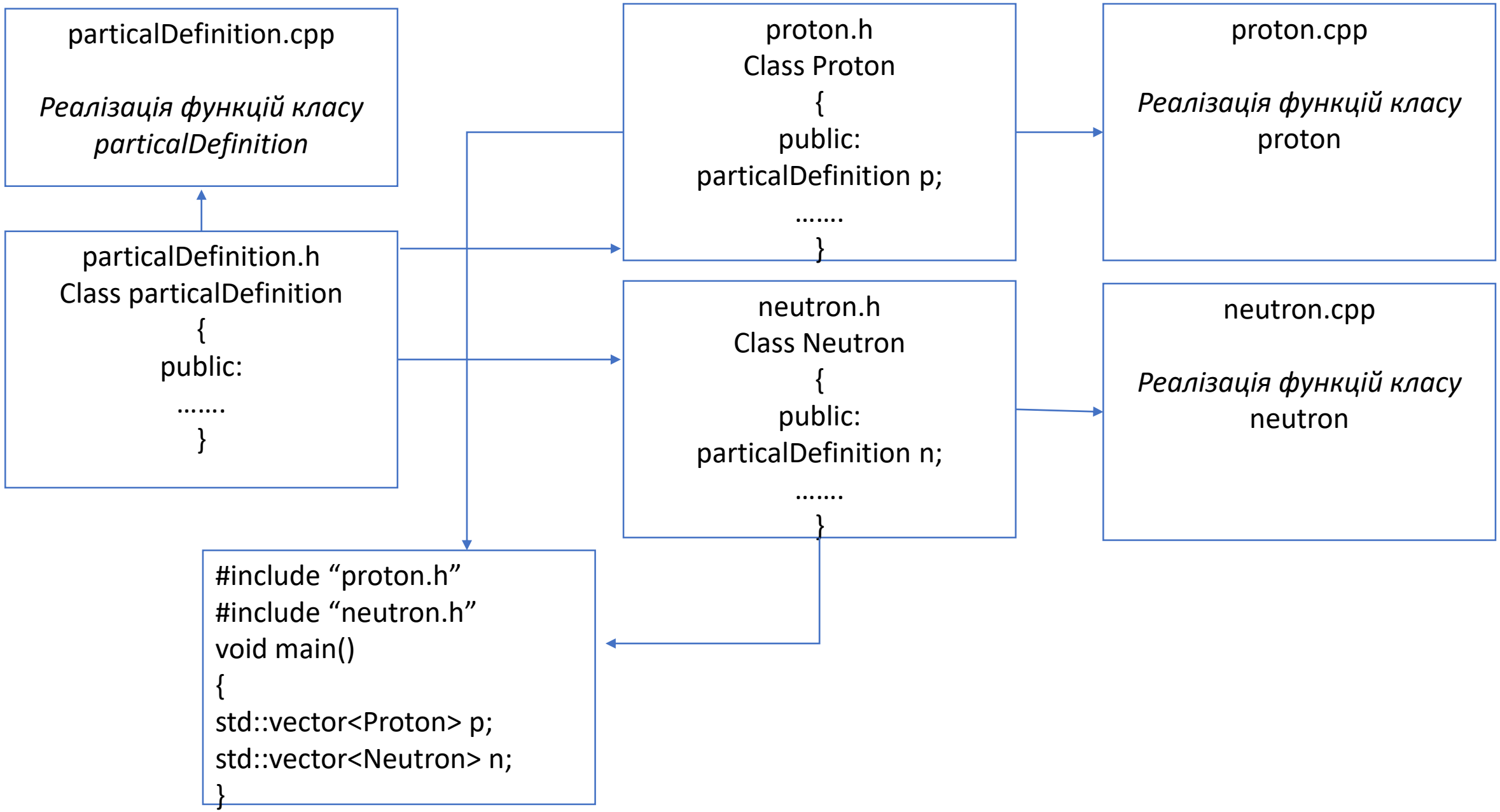




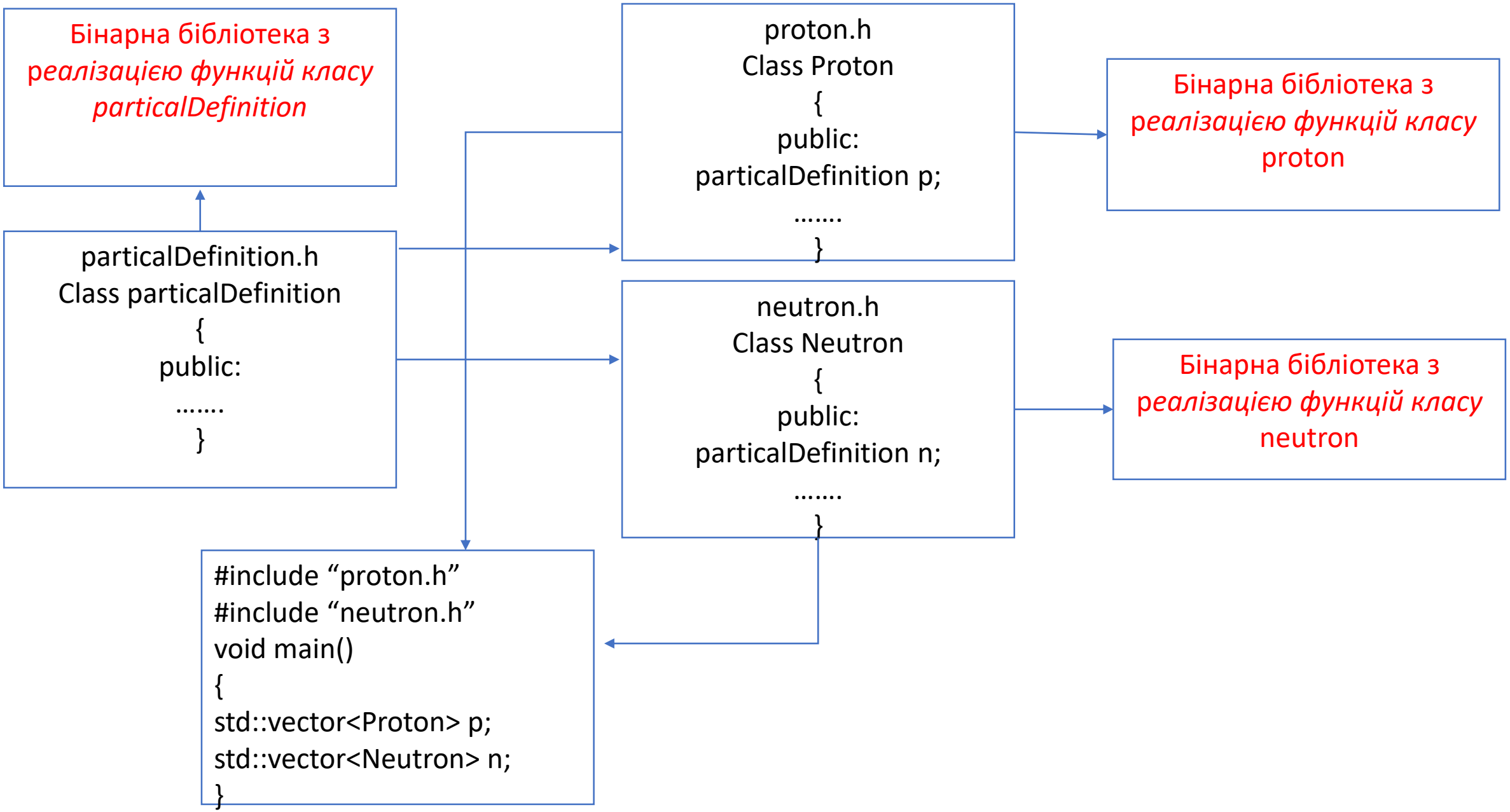




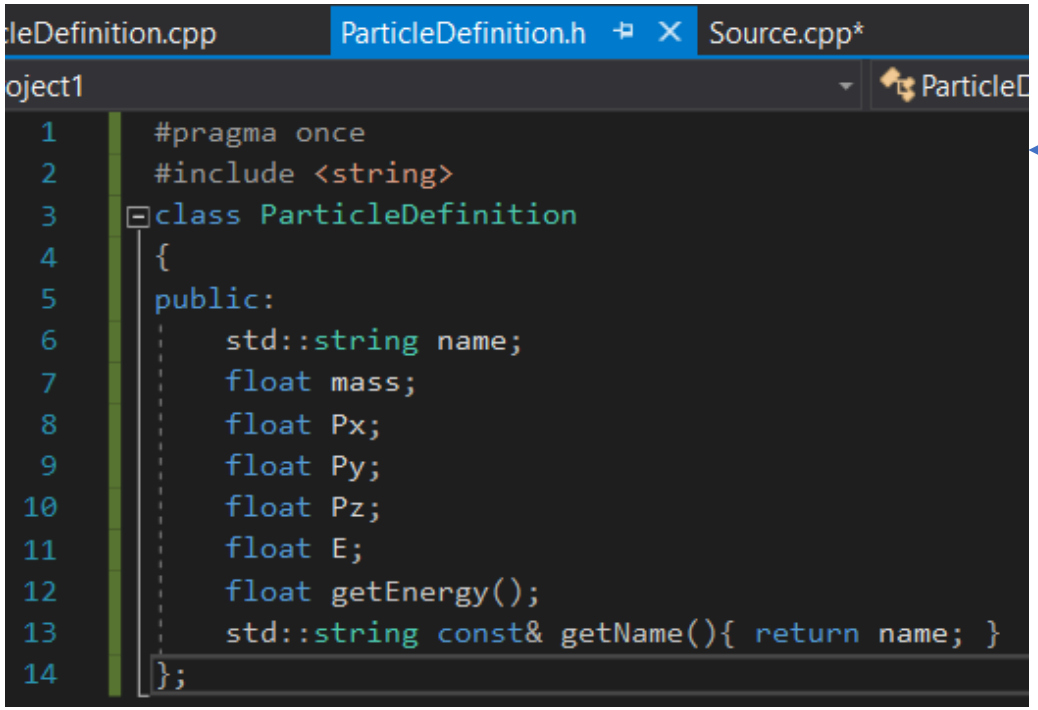
ВИКОРИСТАННЯ ФАЙЛІВ-ЗАГОЛОВКІВ *.h

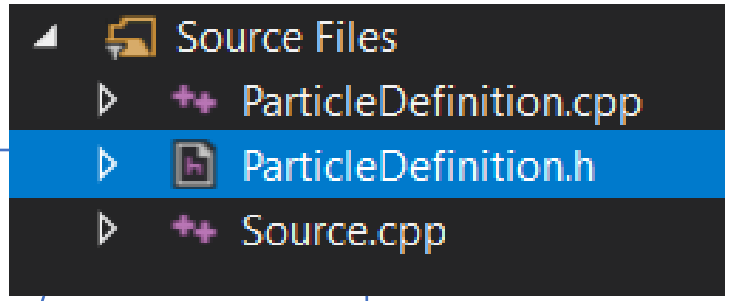


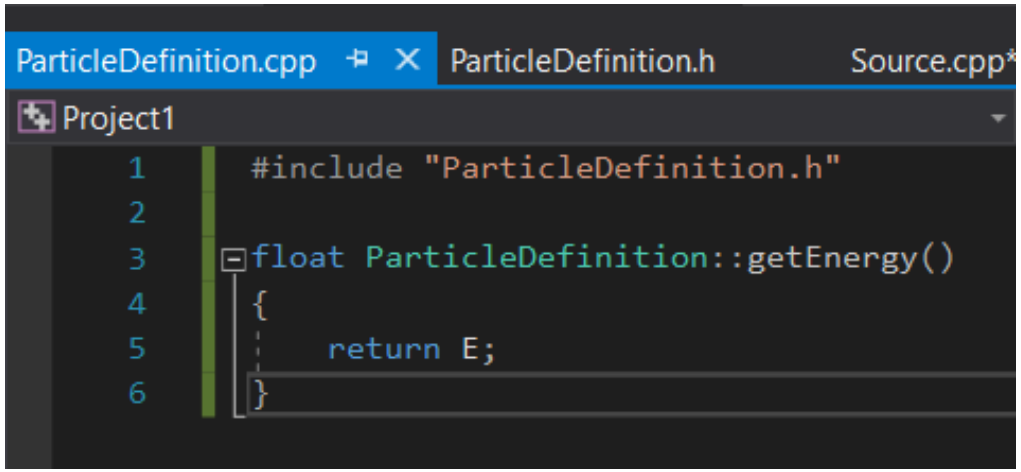
ВИКОРИСТАННЯ ФАЙЛІВ-ЗАГОЛОВКІВ *.h

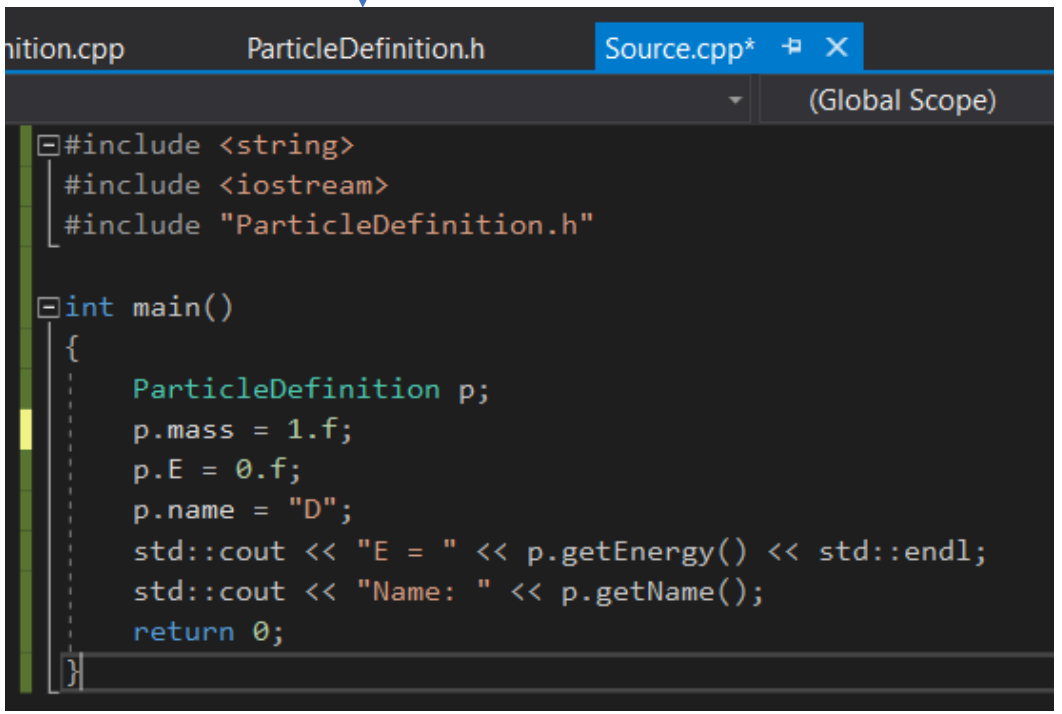


ВИКОРИСТАННЯ ФАЙЛІВ-ЗАГОЛОВКІВ *.h

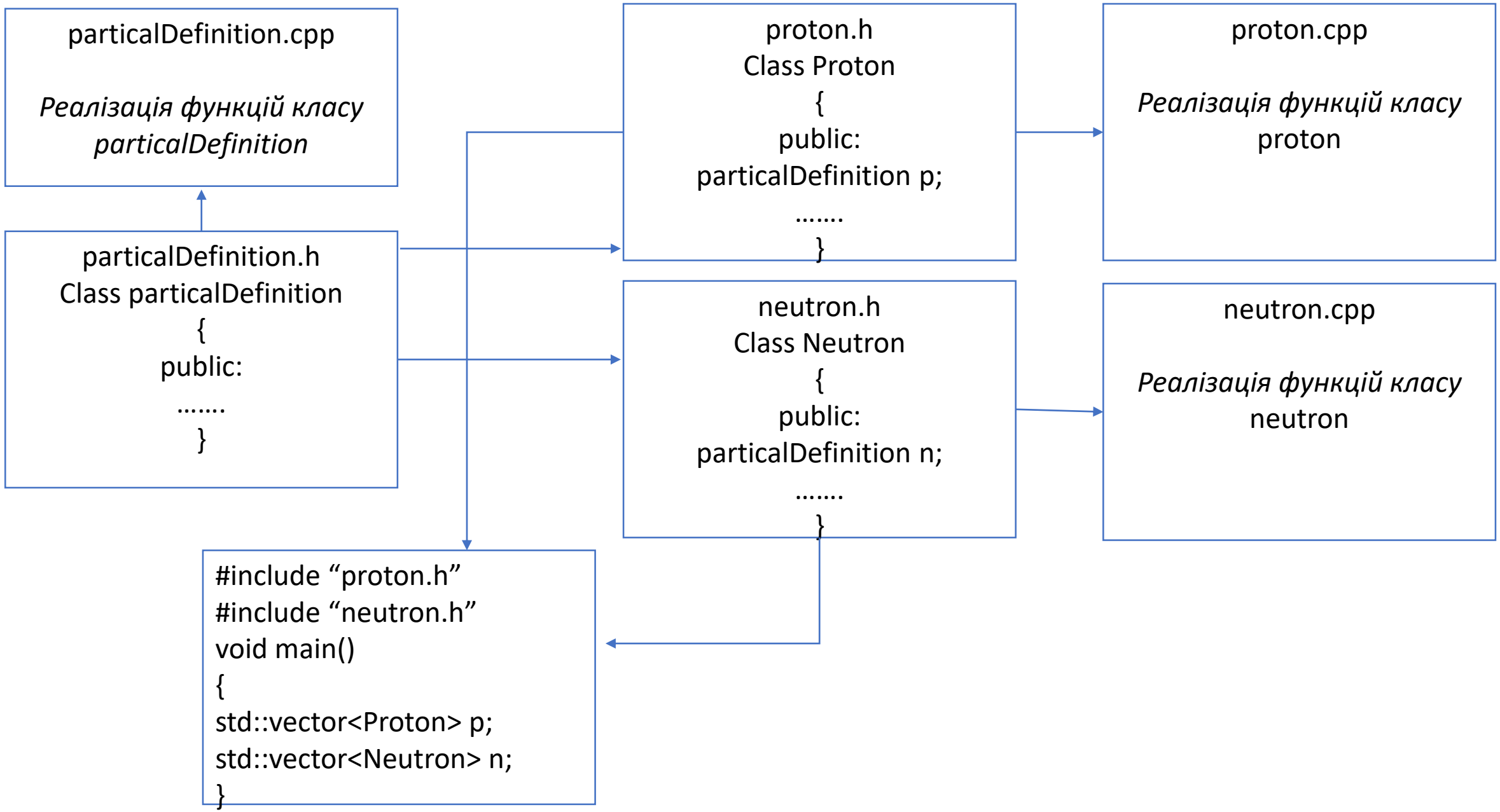




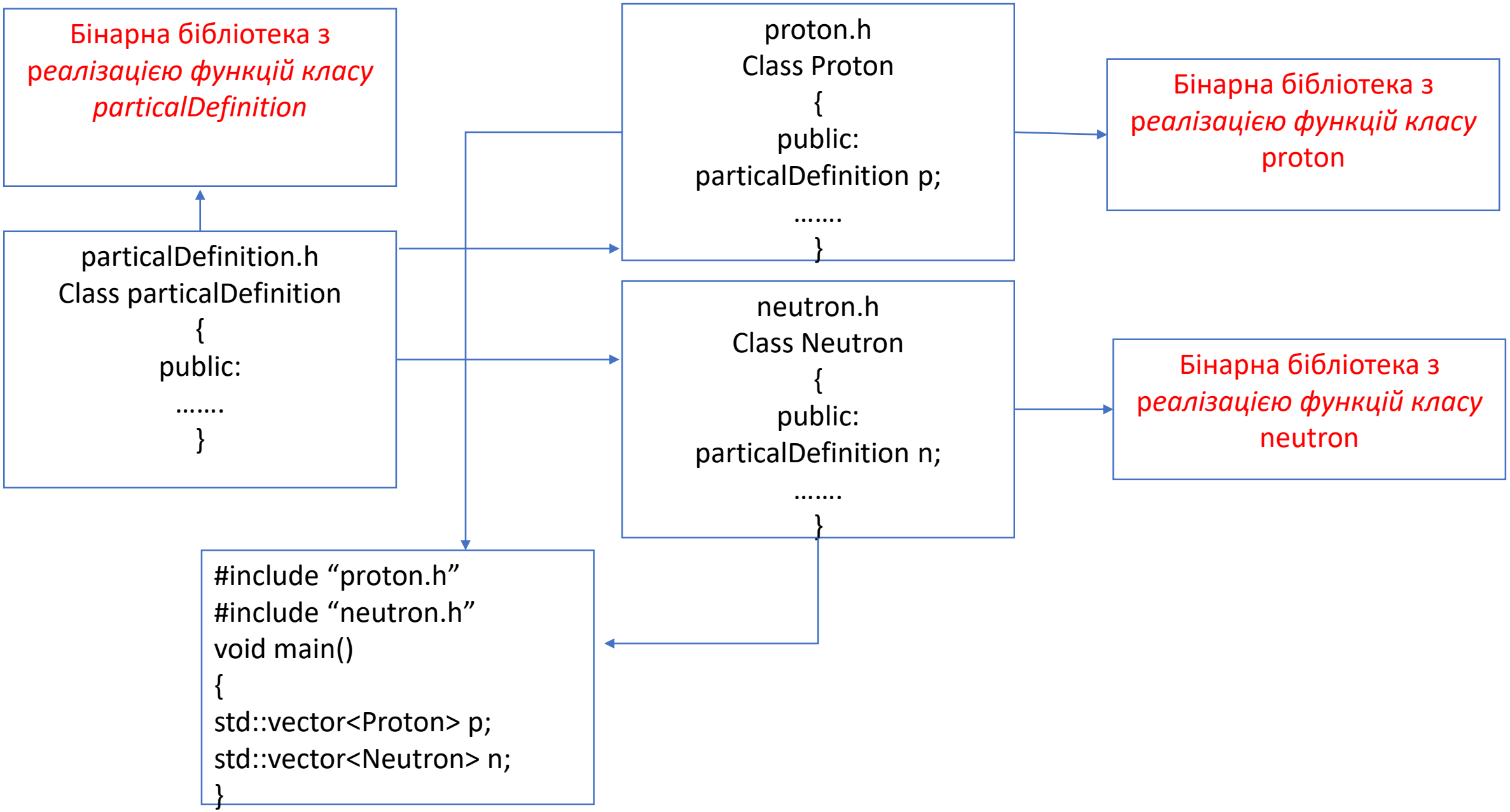




ВИКОРИСТАННЯ ФАЙЛІВ-ЗАГОЛОВКІВ *.h



ВИКОРИСТАННЯ ФАЙЛІВ-ЗАГОЛОВКІВ *.h



ВИКОРИСТАННЯ ШАБЛОНІВ

ParticleDefinition.h | Source.cpp

```
#pragma once
#include <string>

template <typename T>
class ParticleDefinition
{
private:
    std::string m_name;
    T m_mass;
    T m_Px;
    T m_Py;
    T m_Pz;
    T m_E;

public:
    inline T const& getEnergy() { return m_E; }
    std::string const& getName(){ return m_name; }
    T const& getMass() { return m_mass; }
    void setMass(T const& mass) { m_mass = mass; }
    void setName(std::string const& name) { m_name = name; }
};
```

```
1  #include <string>
2  #include <iostream>
3  #include "ParticleDefinition.h"
4  #include <limits>
5
6  int main()
7  {
8      ParticleDefinition<double> p;
9      p.setMass(1.007276466879);
10     p.setName("proton");
11     std::cout.precision(std::numeric_limits< double >::max_digits10);
12     std::cout << "mass = " << p.getMass() << std::endl;
13     std::cout << "Name: " << p.getName() << std::endl;
14     return 0;
15 }
```

template <typename T>- в шаблоні буде використовуватися вбудований тип даних, такий як: int, double, float, char …

template <class T> - в шаблоні функції в якості параметра будуть використовуватися класи.