Создание материалов

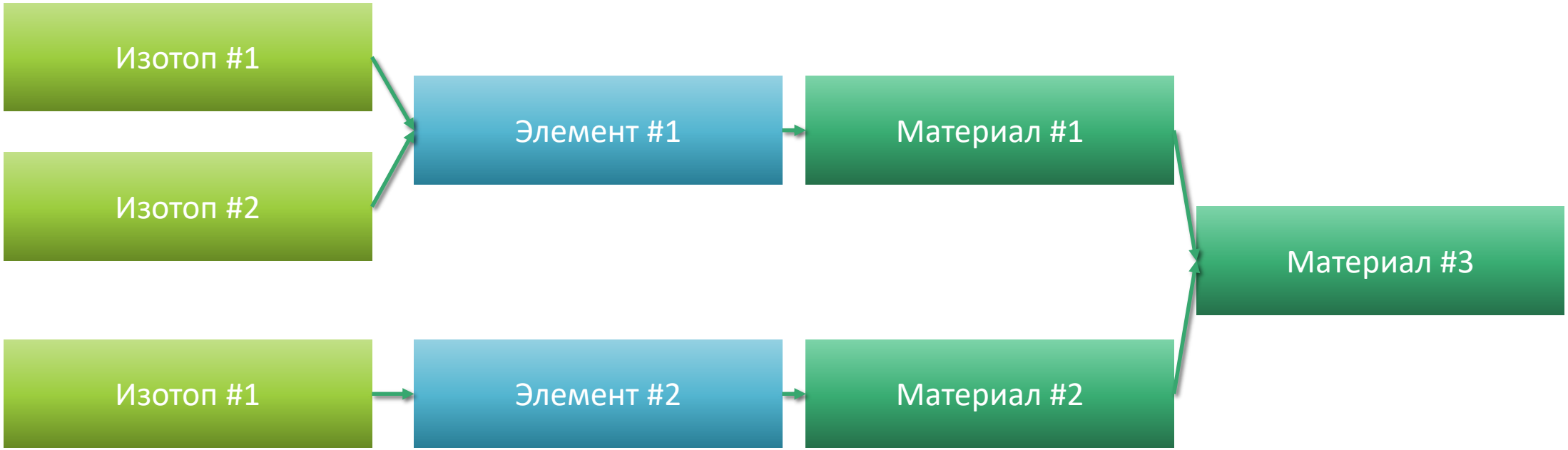
ЛЕКЦИЯ 4

Содержание

- Общая структура материалов
- База материалов NIST
- Вакуум
- Создание не стандартных материалов

Общая структура материалов

Материалы в Geant4 строятся следующим образом: **Материалы** строятся из *элементов* или других *материалов*. **Элементы** строятся из *изотопов*. Материалы должны содержать минимум **1** компонент.



База материалов NIST

В Geant4 представлена база материалов составленная NIST Physical Measurement Laboratory. В данной базе:

- Доступно более 3000 изотопов
- Все представленные элементы составлены исходя из природного баланса изотопов

Официальный сайт (на английском языке):

https://www.nist.gov/pml/productsservices/physical-reference-data

База материалов NIST

- В базе **NIST** представлены элементы от **Водорода** до **Калифорния**(98)

Кроме того в базе доступны различные материалы, к примеру:

- Составные вещества и смеси:
  - Ткань, эквивалентная пластику; морской воздух и т.д.
- Биохимические материалы:
  - Жировая ткань, цитозин, тимин и т.д.
- Композитные материалы:
  - Кевлар и т.д.

Полный список материалов представленных в Geant4: Geant4 Material Database

Создание элементов с использованием базы NIST

- Для того чтобы использовать материалы из базы NIST, нужно инициировать на неё указатель:

```
G4NistManager* nist = G4NistManager::Instance();
```

- Для создания элементов из базы NIST нужно вызывать метод **G4Element**::**FindOrBuildElement**() в качестве аргумента передав либо номер элемента либо его имя:

```
G4Element *el = nist->FindOrBuildElement("Fe");
```

Получение свойств изотопов, в составе элемента

По умолчанию, при построении элемента из базы NIST так же строятся все входящие в его состав изотопы исходя из природного соотношения. К примеру, следующим образом можно получить доступ к изотопам в составе элемента по их индексу:

```
20      G4cout << "For element " << el->GetName() << " :" << G4endl;
21
22      for (unsigned int i = 0; i < static_cast<unsigned int>(el->GetNumberOfIsotopes()); i++) {
23          G4cout << "Isotope : " << el->GetIsotope(i)->GetName() <<
24                  " | Relative Abundance : " << el->GetRelativeAbundanceVector()[i] <<
25                  " | A = " << el->GetIsotope(i)->GetA() / (g / mole) <<
26                  " | Z = " << el->GetIsotope(i)->GetZ() << G4endl;
27      }
```

В результате в консоли:

```
For element Fe :
Isotope : Fe54 | Relative Abundance : 0.05845 | A = 53.9396 | Z = 26
Isotope : Fe56 | Relative Abundance : 0.91754 | A = 55.9349 | Z = 26
Isotope : Fe57 | Relative Abundance : 0.02119 | A = 56.9354 | Z = 26
Isotope : Fe58 | Relative Abundance : 0.00282 | A = 57.9333 | Z = 26
```

Примечание: Ссодержание того или иного изотопа записано в отдельном контейнере RelativeAbundanceVector и никак не связано с классом G4Isotope

Создание материалов с использованием базы NIST

- Чтобы создать материал используя базу NIST необходимо воспользоваться методом **G4Material**::**FindOrBuildMaterial**()

```
G4Material *material = nist->FindOrBuildMaterial("G4_WATER");
```

Аналогично элементам и изотопам, для материала можно определить содержание в нем элементов:

```
33      G4cout << "For material " << material->GetName() << " :" << G4endl;
34
35      for (unsigned int i = 0; i < static_cast<unsigned int>(material->GetNumberOfElements()); i++) {
36          G4cout << "Element : " << material->GetElement(i)->GetName() <<
37                 " | Atoms : " << material->GetAtomsVector()[i] << G4endl;
38      }
```

Где содержание элементов уже задается в количестве атомов:

```
For material G4_WATER :
Element : H | Atoms : 2
Element : O | Atoms : 1
```

Создание материалов с использованием базы NIST

Кроме того в базе материалов представлены «материалы – элементы», к примеру:

```
G4Material *material = nist->FindOrBuildMaterial("G4_Fe");
```

Эти материалы состоят из одного атома, и могут быть использованы для построения материалов, для которых не известна химическая формула, но доступно процентное содержание того или иного элемента:

```
For material G4_Fe :
Element : Fe | Atoms : 1
```

Создание материалов с использованием базы NIST: вакуум

В базе NIST существует специальный материал для создания вакуума:

```
G4Material* vacuum = nist->FindOrBuildMaterial("G4_Galactic");

G4cout<< "Density = " << vacuum->GetDensity()/(g/cm3)<<G4endl;
```

Соответственно выводимая плотность материала:

```
Density = 1e-25
```

Создание материалов «с нуля»: Шаг 1

Если существует необходимость задать не стандартный материал, то можно создать его «с нуля», следуя общей схеме: *изотопы -> элементы -> материал -> материал -> и т.д.*

Создадим UF_6:

- Создадим входящие в состав урана изотопы **G4Isotope**. Конструктор выглядит следующим образом:

```
G4Isotope(const G4String &name,          //имя
          G4int z,                       //атомный номер
          G4int n,                       //число нуклонов
          G4double a = 0.,               //молярная масса
          G4int m = 0)                   //изомерный сдвиг
```

Тогда, используя данный конструктор:

```
G4Isotope *u235 = new G4Isotope("U235", 92, 235, 235.044 * g / mole);
G4Isotope *u238 = new G4Isotope("U238", 92, 238, 238.051 * g / mole);
```

Создание материалов «с нуля»: Шаг 2

- Создадим элемент **G4Element** из созданных ранее изотопов. Конструктор элемента:

```
G4Element(const G4String& name,      //имя
          const G4String& symbol,    //символьное обозначение элемента
          G4int nbIsotopes)          //количество изотопов
```

Используя данный конструктор:

```
G4Element *enrichedU = new G4Element("enrichedU", "U", 2);
enrichedU->AddIsotope(u235, 5.0 * perCent);
enrichedU->AddIsotope(u238, 95.0 * perCent);
```

Так же существует возможность создавать элементы не задавая изотопы напрямую:

```
G4Element(const G4String& name,      //имя
          const G4String& symbol,    //символьное обозначение элемента
          G4double  Z,               //атомный номер
          G4double  A)               //молярная масса
```

Тогда для фтора:

```
G4Element *elF = new G4Element("Fluorine", "F", 9., 18.998 * g / mole);
```

Создание материалов «с нуля»: Шаг 3

- Создадим материал используя следующий конструктор:

```
G4Material(const G4String& name,                    //имя
           G4double  density,                       //плотность
           G4int     nComponents,                   //количество компонентов
           G4State   state     = kStateUndefined,   //состояние
           G4double  temp      = NTP_Temperature,   //температура
           G4double  pressure = CLHEP::STP_Pressure)//давление
```

Состояние может принимать следующие значения, описанные энумерацией:

kStateUndefined	**0**	**не определено**
kStateSolid	1	твердое
kStateLiquid	2	жидкое
kStateGas	3	газообразное

Создание материалов «с нуля»: Шаг 3

Тогда в результате, используя данный конструктор, мы получим:

```
G4Material *fuel = new G4Material("NuclearFuel", 5.09 * g / cm3, 2, kStateSolid,
                                  640 * kelvin, 1.5e7 * pascal);
fuel->AddElement(elF, 6);
fuel->AddElement(enrichedU, 1);
```

Примечание: если не указывать температуру и давление, то будут заданы нормальные условия

Создание материалов «с нуля»: Финал

В результате должно получиться:

```
33      G4Isotope *u235 = new G4Isotope("U235", 92, 235, 235.044 * g / mole);
34      G4Isotope *u238 = new G4Isotope("U238", 92, 238, 238.051 * g / mole);
35
36      G4Element *enrichedU = new G4Element("enrichedU", "U", 2);
37      enrichedU->AddIsotope(u235, 5.0 * perCent);
38      enrichedU->AddIsotope(u238, 95.0 * perCent);
39
40      G4Element *elF = new G4Element("Fluorine", "F", 9., 18.998 * g / mole);
41
42      G4Material *fuel = new G4Material("NuclearFuel", 5.09 * g / cm3, 2, kStateSolid,
43                                        640 * kelvin, 1.5e7 * pascal);
44      fuel->AddElement(elF, 6);
45      fuel->AddElement(enrichedU, 1);
```