

# **Компьютерные технологии в науке и образовании**

# О чем этот курс

Курс будет состоять из 2 частей

- **Введение в Geant4**

(~8 занятий)

Иванов Артём Викторович

- **МЕТОДЫ ОБРАБОТКИ И АНАЛИЗА ДАННЫХ  
ЭКСПЕРИМЕНТАЛЬНЫХ УСТАНОВОК В СРЕДЕ CERN ROOT**

(~8 занятий)

Герценбергер Константин Викторович

**Автомат** – это посещение всех занятий и работа на практике

**Зачет** – 1 задание по ROOT или GEANT4

# **Введение в Geant4**

**Иванов Артем Викторович**

**E-mail: [arivanov@jinr.ru](mailto:arivanov@jinr.ru)**

# Что нужно для курса

## Обязательно

- синтаксис и основные понятия языка C++
- основы объектно-ориентированного программирования (ООП)
- основы работы в ОС Linux
- физику элементарных частиц

## Желательно

- Система сборки cmake, make, знание английского языка, Git

# Введение

- Программы для моделирования процессов производится с учётом как кинематики процесса, так и внутренних симметрий, запретов, квантовых чисел. А при условии, что генератор полного цикла – ещё и адронизации, развития ливня и распадов нестабильных частиц. Один из наиболее популярных это **Pythia8**. Так же **Herwig, MadGraph, CompHEP, MCFCM, VBFNLO ...**
- Для моделирования установки используется в основном пакет **GEANT4**. Который включает в себя модели известных физических законов в сочетании с гибким инструментом создания геометрии и материалов. Так же **FLUKA, MCNPX ...**
- Стандартом для анализа данных в физике высоких энергий является **ROOT**.
- Фреймворки широко распространены для организации ПО эксперимента и общих операций (работа с БД, хранение данных, организация цикла событий и т. д.) Существует семейств фреймворков (**Gaudi, FairRoot ...**)

# Цели моделирования

## При планировании эксперимента

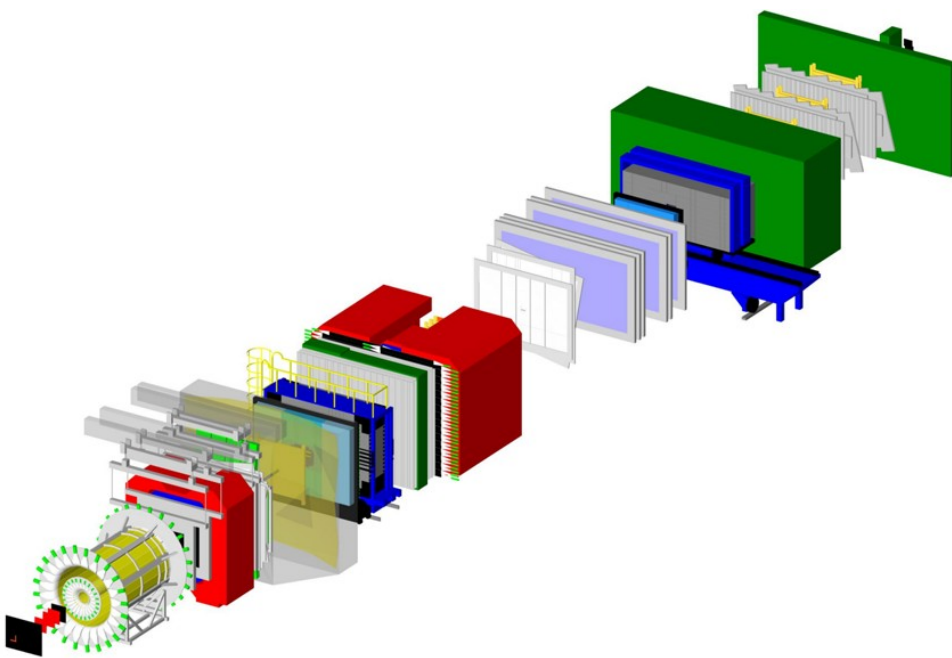
- Оптимизация конструкции детектора
- Отладка алгоритмов реконструкции событий
- Расчет ожидаемых значений сигнала и фоновых процессов.
- Оценка ожидаемой точности измерений

## При анализе данных

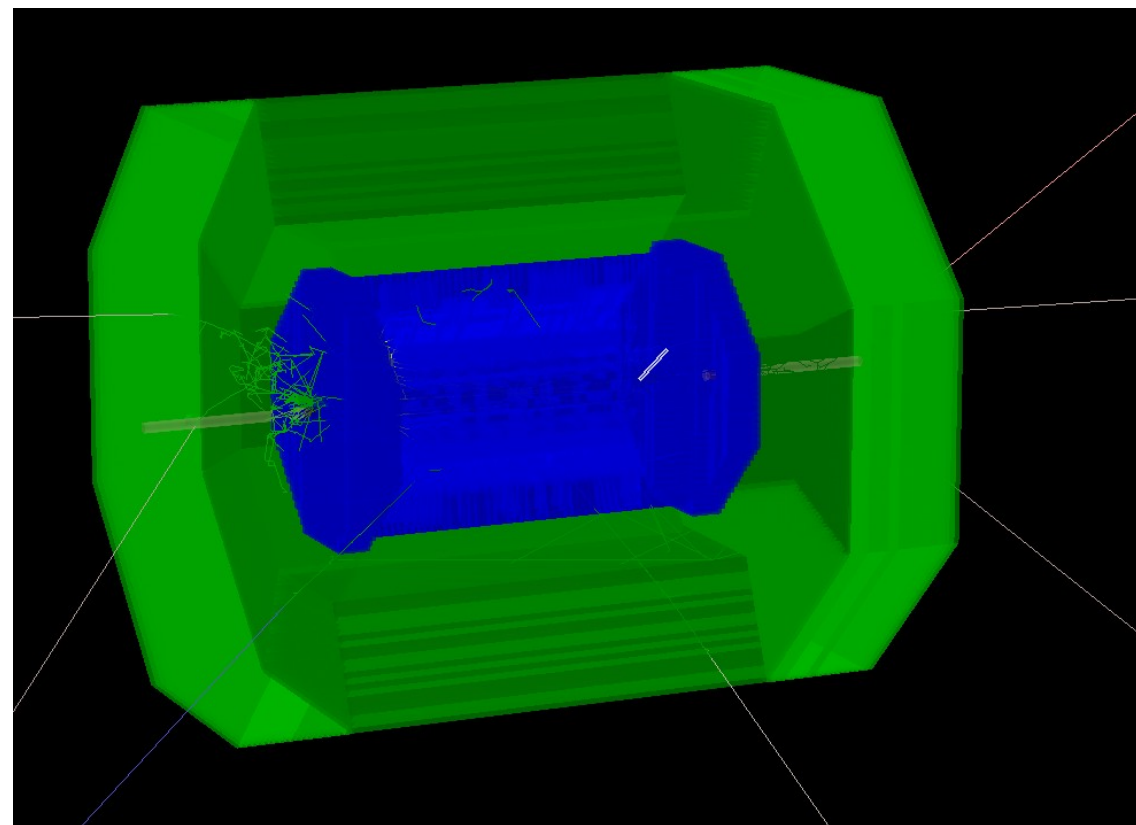
- оптимизация процедуры анализа данных
- аксептанса установки
- определение вклада фоновых процессов
- оценка систематических погрешностей
- сравнение результатов анализа с теоретическими предсказаниями

# Примеры экспериментов

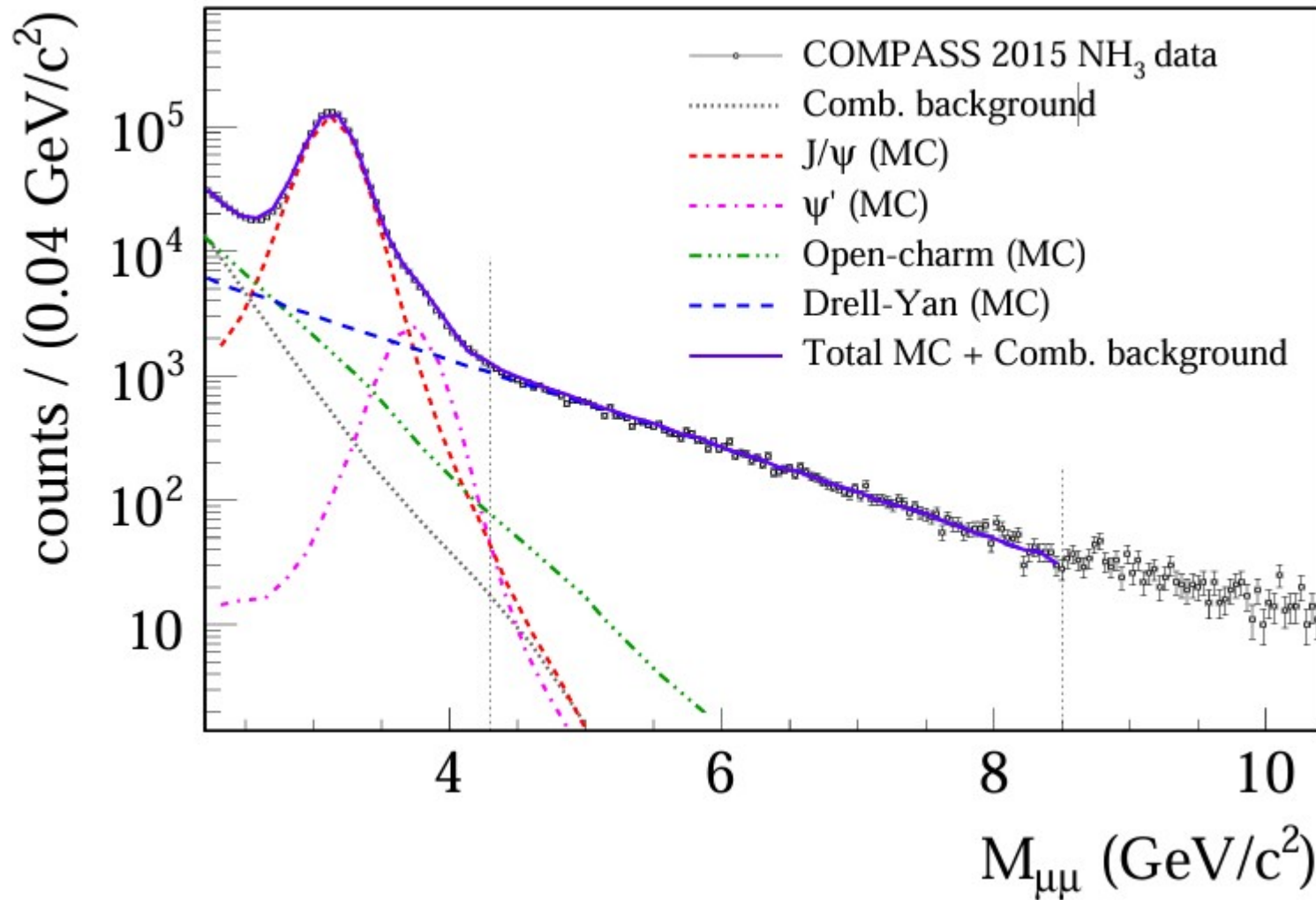
COMPASS



SPD



# Примеры из физики

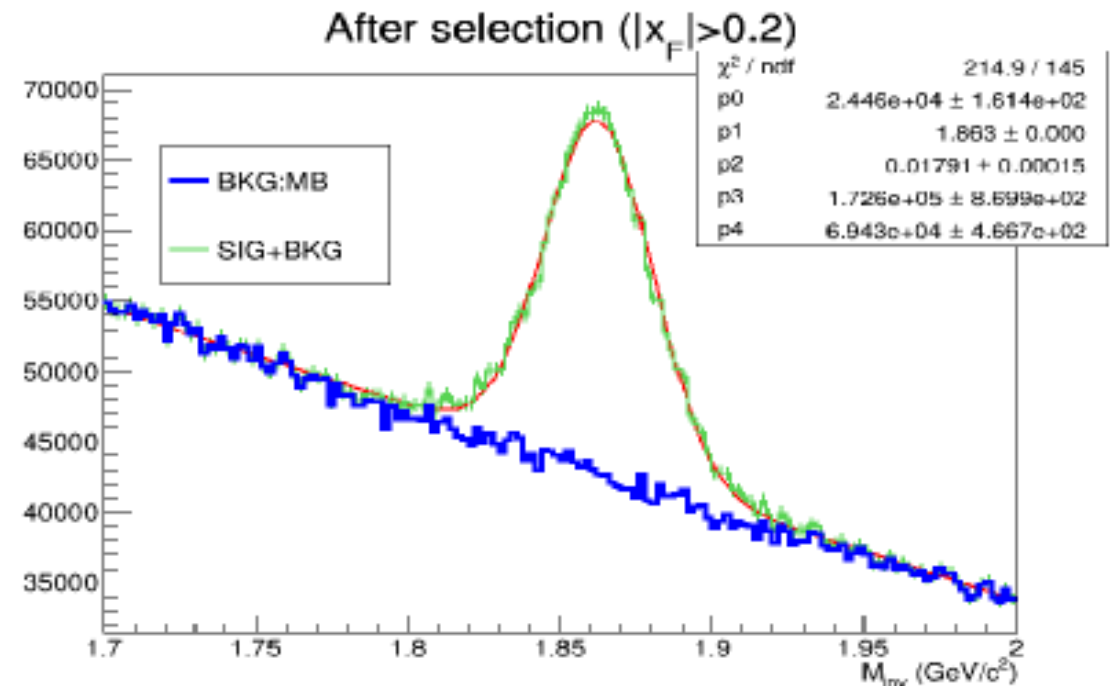
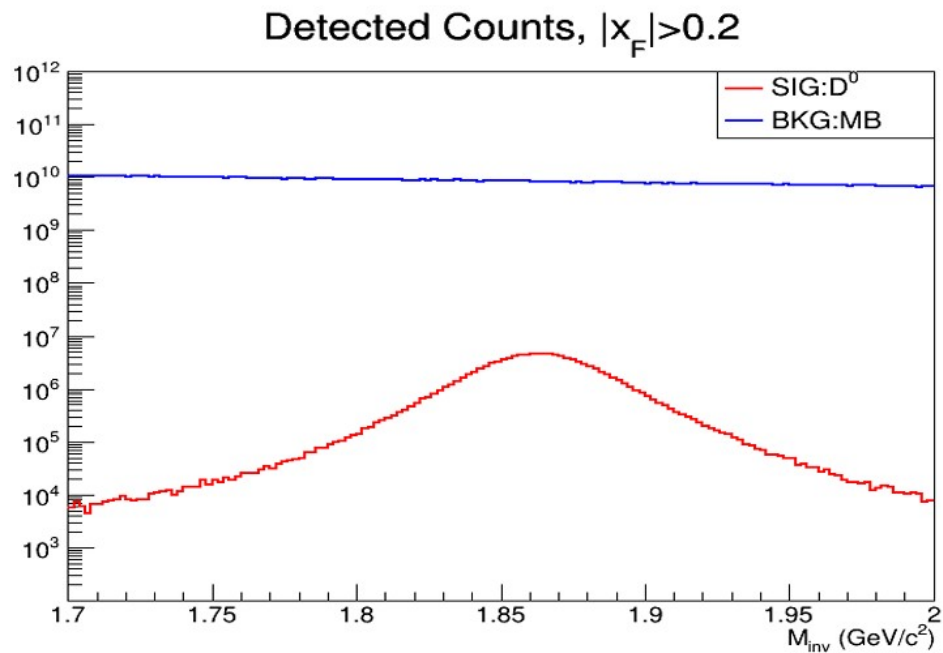


In this range, the sum of all background contributions is estimated to be below 4%.



# Примеры из физики

$$D^0 \rightarrow \pi^+ K^-$$



# Введение

**Geant4** – это пакет для моделирования прохождения элементарных частиц через вещество с использованием методов Монте-Карло.

В пакет включены:

- средства описания геометрии
- описание материалов
- проведение частиц через вещество и электромагнитные поля
- модели физические процессов, описывающие взаимодействие частиц с веществом с энергиями от эВ до ТэВ.
- визуализация детектора и траекторий частиц
- сохранение и анализ моделированных данных

# История

1974 – GEANT

1982 – GEANT3

- Написан на языке Фортран

1995 – GEANT4

- Объектно-ориентированная программа с функциональностью GEANT3
- Написана на языке C++

# Область применения

- **High Energy Physics**

- LHC experiments ( [ALICE](#), [ATLAS](#), [CMS](#), [LHCb](#) )
- future experiments ( [CLIC](#), [ILC](#), [FCC](#) )
- and many others ( [Fermilab experiments](#), [HARP](#), [BDSIM](#), [SLAC](#), [BaBar](#), ... )

- **Space and Radiation Science**

- European Space Agency ( [Geant4 Space Users](#), [SPENVIS](#), [GRAS](#), [MULASSIS](#) )
- Gamma Ray Large Area Space Telescope ( [GLAST](#) )
- Medium Energy Gamma-ray Astronomy library ( [MEGALib](#) )

- **Medical Physics**

- [Geant4-DNA](#) - Geant4-DNA project
- [G4MED](#) (in Japanese) - Geant4 Medical Physics in Japan
- [G4NAMU](#) - Geant4 North American Medical User Organization
- [GAMOS](#) - Geant4-based Architecture for Medicine-Oriented Simulations
- [GATE](#) - Geant4 Application for Tomographic Emission
- [GHOST](#) - Human Oncology Simulation Tool
- [TOPAS](#) - Geant4 Monte Carlo Platform for Medical Applications


# Важно

Geant4 рассматривает классические частицы с четко определенным положением и импульсом

Частицы рассматриваются классически, но их взаимодействия (сечение и конечные состояния) учитывают результаты квантово-механических эффектов. Берутся из экспериментальных или теоретических данных  
Реализованы в Geant4 либо в виде таблиц, либо эмпирических формул

# Официальный сайт

<https://geant4.web.cern.ch/>



Geant4

About

Download

Documentation

User Forum

Bug Reports

Events

Contact Us

Geant4

Toolkit for the simulation of the passage of particles through matter. Its areas of application include high energy, nuclear and accelerator physics, as well as studies in medical and space science.

Getting started

### Get started

Everything you need to get started with Geant4.

[I'm ready to start!](#)

### Download

Geant4 source code and installers are available for download, with source code under an open source license.

Latest: [11.3.2](#)

### Docs

Documentation for Geant4, along with tutorials and guides, are available online.

[Read documentation](#)

### News

26 Jun 2025

[Release 11.4.beta](#)

25 Apr 2025

[Release 11.3.2](#)

21 Mar 2025

[Release 11.3.1](#)

11 Mar 2025

[2025 Planned Features](#)

06 Dec 2024

[Release 11.3](#)

[» More](#)

### About us

What is Geant4, where it's used, details on Collaboration.

[Learn More](#)

### Collaboration

Geant4 team and documents

[Learn More](#)

### Contribute

How external users can contribute to Geant4.

[Learn More](#)

### Events

[» More](#)

- 8/18/2025

-

8/21/2025

Geant4 Tutorial Course at Jefferson Lab, Jefferson Lab, Newport News, Virginia (USA)
- 9/22/2025

-

9/25/2025

Introductory lessons and hands on Geant4 at the 2nd Intensive School on Laser, Plasma and Fusion, Budva (Montenegro)
- 10/7/2025

-

10/10/2025

13th International Geant4 School, Laboratori Nazionali del Sud (INFN/LNS), Catania (Italy)
- 10/20/2025

-

10/24/2025

30th Geant4 Collaboration Meeting, Stony Brook University, New York (USA)

Geant4 Homepage

Geant4 Documentation



11.3 (doc Rev9.2)

Search docs

- Introduction to Geant4
- Installation Guide
- Book For Application Developers
- Physics Reference Manual
- Guide for Physics Lists
- Frequently Asked Questions (FAQ)
- User's Guide for Toolkit Developers

Docs » Geant4 Documentation

### Geant4 Documentation

- Introduction to Geant4
  - Geant4 Scope of Application
  - History of Geant4
  - Overview of Geant4 Functionality
  - Geant4 User Support
  - Software Knowledge Required to Use the Geant4 Toolkit
  - Computing Environment Required by the Geant4 Toolkit
- Installation Guide
  - Build and Install Geant4 from Source
  - Install Geant4 via a Package Manager
- Book For Application Developers
  - Introduction
  - Getting Started with Geant4 - Running a Simple Example
  - Toolkit Fundamentals
  - Detector Definition and Response
  - Tracking and Physics
  - User Actions
  - Control
  - Visualization
  - Analysis
  - Language Bindings
  - Examples
  - Appendix
  - Bibliography
- Physics Reference Manual
  - General Information
  - Decay
  - Electromagnetic Interactions
  - Hadronic Physics in Geant4
  - Gamma- and Lepto-Nuclear Interactions
  - Neutrino Interactions
  - Solid State Physics
- Guide for Physics Lists
  - Physics List Guide
  - Reference Physics Lists
  - Electromagnetic physics constructors
  - Hadronic Physics
- Frequently Asked Questions (FAQ)
  - Installation
  - Run Time Problems
  - Geometry
  - Tracks and steps
  - Physics and cuts
  - Visualization
  - User Support Policy
- User's Guide for Toolkit Developers
  - Introduction
  - Design and Function of Geant4 Categories
  - Extending Toolkit Functionality
  - Bibliography

### Status of this Document

Guide for Application Developers using the Geant4 toolkit.

- Rev 1.0: First Sphinx version implemented for Geant4 Release 10.4, 8th Dec 2017
- Rev 2.0: Updates and fixes in documentation for Geant4 Release 10.4, 15th May 2018
- Rev 3.0: Geant4 Release 10.5, 11th December 2018
- Rev 3.1: Geant4 Updates and fixes - especially to search functionality, 5th March 2019
- Rev 4.0: Geant4 Release 10.6, 6th December 2019
- Rev 5.0: Geant4 Release 10.7, 4th December 2020
- Rev 6.0: Geant4 Release 11.0, 10th December 2021
- Rev 7.0: Geant4 Release 11.1, 9th December 2022

Next

# Системные требования

## Язык

- C++
- Geant4Py — реализация Geant4 на Python

## Основные платформы

- Linux + gcc
- Windows + VisualC++
- MacOS + gcc4

Требует около 2.3 Гб дискового пространства (полная установка)



# Установка и настройка Geant4

## Source code

- Скачать исходный код и скомпилировать его самостоятельно.

Source code is freely available from [CERN GitLab](#) or through [GitHub](#).

Source code can also be browsed through the [LXR source code browser](#).

Download zip

Download tar.gz

Download tar.bz2

Download tar

## Binary releases

Download tar.gz

MacOS Sequoia, clang-17.0.0

Download tar.gz

Linux Alma9, g++-11.5.0

Download exe

Windows 11, Visual Studio Code-17.10.3

Download zip

Windows 11, Visual Studio Code-17.10.3

- Скачать скомпилированный GEANT4 (бинарную сборку)

- Скачать виртуальную машину (Virtual Box) или docker image с предустановленным GEANT4
- А также через различные пакетные менеджеры Spack/Homebrew/Conda/Guix



# Кратко по командам LINUX

- команда, которая выводит полный путь к текущей директории, в которой находится пользователь.

*pwd*

- отображать информацию о директориях (каталогах) и файлах, находящихся в папках.

*ls (-alh)*

- команда для создания новых каталогов.

*mkdir directory\_name*

- для смены текущего рабочего каталога

*cd path\_name*      .    текущая директория  
                         ..    родительская директория  
                         ~    домашняя директория

- для удаления файлов и каталогов из файловой системы

*rm file\_name , rm -r directory\_name*

# Кратко по командам LINUX

- используется для копирования файлов и каталогов:

*cp old\_file\_name new\_file\_name, cp -r old\_dir\_name new\_dir\_name*

- используется для переименования файлов и каталогов или для их перемещения в другое местоположение:

*mv old\_name new\_name*

- Простой графический редактор в Linux:

*gedit filename*

# Кратко что важно знать из C++

```
#include <iostream>
#include <string>

using namespace std;

int main(int argc, char **argv) {

    double Mproton = 938.272 //MeV

    cout<< argv[1] <<" "
         << Mproton
         << endl;

    return 0;
}
```

## Запуск программы

./prog proton

***int argc** - количество аргументов, переданных при запуске программы.*

**char \*\*argv** - массив строк (char\*), где каждый элемент — один аргумент командной строки.

**argv[0]**: Имя исполняемого файла (может быть полным путем или коротким именем, зависит от ОС).

**argv[1] ... argv[argc-1]**: Остальные аргументы.

# Кратко что важно знать из C++

*// Класс detector - модель реального объекта*

**class Detector {**

**private:** *// Секция приватных членов*

int size;

string name

**public:** *// Секция публичных методов*

*// Конструктор (инициализирует объект)*

**Detector**(double \_size, string \_name){ size = \_size; \_name = name;}

*// Метод для вывода информации о детекторе*

**void print ()**{

cout<<" name="<<name<<" size="<<size<<endl;

}

*// Метод для вычисления чего-то важного*

double **doing\_something\_important()**{

*// do do do do do*

return nobelprize;

}

};

int main() {

*// Создание объекта класса Detector*

**Detector** ecal(20.2, "ECAL");

ecal.print();

double prize=ecal.doing\_something\_important();

return 0;

}

# Кратко что важно знать из C++

**Наследование** — это механизм, при котором один класс (производный) наследует свойства и методы другого класса (базового). Это позволяет повторно использовать код

```
class MyDetector : public Detector {
public:
    MyDetector(double _size, sting _name) : Detector(double _size, sting _name)
    {}

    void measure() {
        // measure number of hits
    }
};
```

```
int main() {
    // Создание объекта класса Detector
    MyDetector super_ecal(20.2, "ECAL");
    super_ecal.doing_something_important();
    super_ecal.measure();
    super_ecal.print();

    return 0;
}
```

# Кратко что важно знать из C++

**Пространства имён (namespace) в C++** — это механизм для организации кода в логические группы, позволяющий избежать конфликтов имён. Они инкапсулируют идентификаторы (переменные, функции, классы и т.д.) в определённой области видимости

**#ifndef/#endif и #define** — это директивы препроцессора C++, используемые для защиты от множественного включения заголовочных файлов.

```
#ifndef Geant4_h  
#define Geant4_h 1  
  
#endif
```

- Препроцессор проверяет, не был ли ранее определён макрос с именем Geant4\_h
- Если макрос не определён -> переходим к следующей строке
- Если макрос уже определён -> пропускаем весь код до #endif

# Что еще важно знать

**CMake** — это кроссплатформенная система для управления процессом сборки. Она не собирает проект напрямую, а генерирует файлы для нативных систем сборки (например, Makefile для Make, проекты для Visual Studio, Xcode и т.д.).

**Make** — это утилита, которая автоматизирует процесс сборки программ. Она читает инструкции из файла с названием Makefile (или makefile). В этом файле описываются зависимости между файлами и команды для компиляции и linking.

# Базова сборка

Процесс сборки описан для версии **Geant4-11.3.2** для **Ubuntu-24.04.3**

- Устанавливаем необходимые утилиты для сборки: компилятор C++, cmake, а также необходимые библиотеки

```
sudo apt-get install g++ git cmake libexpat1 libexpat1-dev
```

- Для GUI будем использовать qt5

```
sudo apt-get install qtbase5-dev
```

- Скачиваем архив с официального сайта.
- Создаем директорию geant4 в своем домашнем каталоге

```
mkdir geant4
```

- Распаковываем в неё скачанный архив
- В директории geant4 создаем папки *build* и *install*.

```
mkdir build install
```

В папке *build* мы будем собирать GEANT4, а в папку *install* устанавливать.

- Переходим в директорию *build*:

```
cd ~/geant4/build`
```



# Установка и настройка Geant4

```
cmake -DCMAKE_INSTALL_PREFIX=../install -DGEANT4_INSTALL_DATA=ON  
-DGEANT4_USE_QT=ON -DGEANT4_BUILD_MULTITHREADED=ON ../geant4
```

**-DCMAKE\_INSTALL\_PREFIX=/path/to/directory**

вы указываете путь до директории, в которую нужно установить Geant4

**-DGEANT4\_INSTALL\_DATA=ON**

устанавливается в значение ON, если надо скачать data-файлы. Если у вас уже есть эти файлы, укажите к ним путь. Пункт обязательный.

**-DGEANT4\_USE\_QT=ON**

после установки визуализация будет по умолчанию осуществляться средствами QT

**-DGEANT4\_BUILD\_MULTITHREADED=ON**

geant4 будет установлен с поддержкой многопоточности, что позволяет проводить расчеты более чем на одном ядре

# Базовая сборка

- Собираем GEANT4

*make -jN* , где N количество ядер

- Устанавливаем GEANT4:

*make install*

- Устанавливаем переменные окружения:

*source ~/geant4/install/bin/geant4.sh*

*Данную команду можно прописать в ~/.bashrc для автоматической установки переменных среды.*

# Сборка проекта

geant4/geant4-11.3-release/install/share/Geant4/examples

*basic*

*advanced*

*extended*

- Создаем директорию

```
mkdir practice
```

```
cd practice
```

```
cp geant4-11.3-release/install/share/Geant4/examples/basic/B1 ./
```

- Создаем директорию

```
mkdir build
```

- Собираем проект

```
cd build
```

```
cmake ../B1
```

```
make
```

# Режимы работы

В Geant4, по умолчанию, предполагается 2 режима работы: *интерактивный* или *пакетный*.

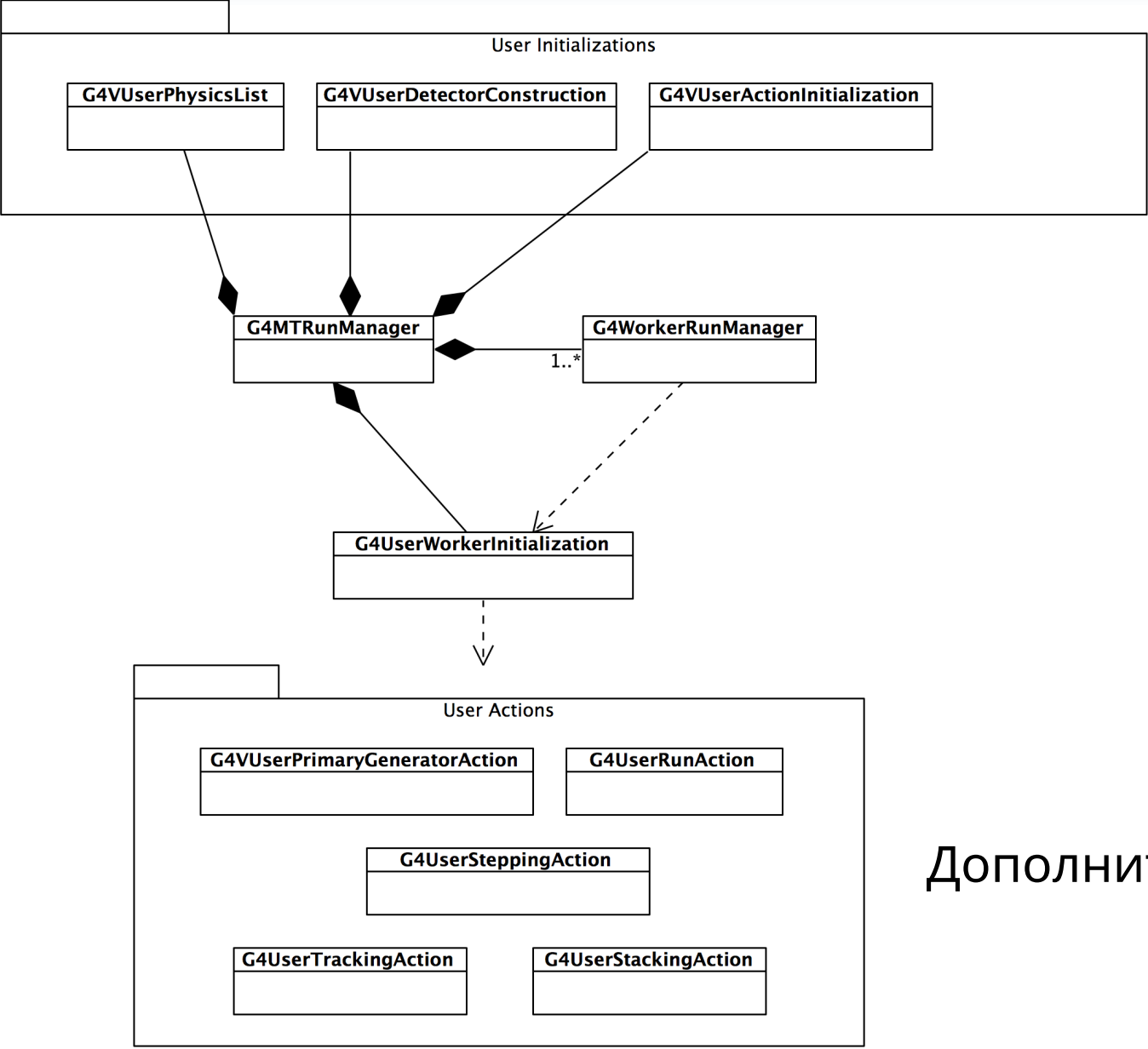
- Под пакетным режимом работы следует понимать использование заранее созданного макрос файла представляющего собой список команд в планируемом порядке их запуска.

*./exampleB1 run1.mac*

- Под интерактивным режимом подразумевается вызов визуальной оболочки Geant4, а так же интерфейса командной строки для последующего ввода команд:

*./exampleB1*

# Схема Geant4



Базовые класс

Дополнительные классы

# Базовые классы

Обязательными для наследования являются **3 класса**:

*G4VUserDetectorConstruction*

задается геометрия детектора и используемые материалы

*G4VUserPhysicsList*

задается используемые частицы, и взаимодействия в которых они участвуют;

*G4VUserPrimaryGeneratorAction*

создаются первичные частицы – задается их тип, направление движения, энергия и т.д.