

Aufgaben Blatt 9

10.1.2025

Bitte bearbeiten Sie die Aufgaben bis zum 17.1.2025, 8:00 Uhr und schicken Sie den source Code von Aufgabe 2 als tar File an marks@physi.uni-heidelberg.de (mit Angabe Ihres Names).

Aufgabe 1: I/O in ROOT Files

Das File [rootIO_commands.txt](#) enthält Anweisungen zum Input/Output Verhalten von ROOT. Bitte probieren Sie die Kommandos in einer interaktiven ROOT session aus.

Aufgabe 2: I/O von beliebigen Objekten in ROOT Files

Um I/O Funktionen wie das Schreiben von beliebigen Objekten in ROOT verwenden zu können, müssen wir

- das `ClassDef` macro in unsere Klasse hinzufügen, dazu wird der header `#include TDirectory.h` benötigt (siehe Seite 65 Root.pdf)
- ein ROOT Dictionary mit `rootcling` erzeugen (siehe Seite 66 Root.pdf)
- eine shared library bauen (siehe Seite 67 Root.pdf)
- eine Anwendung erzeugen

a) Verwenden Sie die Klasse der komplexen Zahlen aus Aufgabe 2 Blatt 3, um das Schreiben von Objekten der komplexen Zahlen in ROOT Files zu testen. Separieren Sie dazu das File `aufgabe3.1.cc` in ein header File `myComplex.h`, das nur die Klassenbeschreibung enthält, und ein File `myComplex.cc`, das nur die Methoden der Klasse enthält. `myComplex.h` benötigt einen `ClassDef` Eintrag und es muss `LinkDef.h` geschrieben werden. Dann läßt sich mit `rootcling` das Dictionary erzeugen und eine shared library `libmyComplex.so` bauen, die dann zu Ihren Anwendungen gelinkt werden kann. Siehe Anweisungen in `HowTo_Aufgabe9.2`

b) Schreiben Sie ein Programm, das mehrere Objekte der komplexen Zahlen in einen ROOT Tree schreibt. Framework: `writeComplexObject.C`