Übungen zu EDV für Physikerinnen und Physiker (physik131) WS 2011/2012

Jörg Pretz und Daniel Elsner

10. Übung Woche: 19.12.-23.12.2011

Lernziele

Einführung in ROOT

Präsenzübungen

Auf der Internetseite root.cern.ch finden Sie sowohl das ROOT Softwarepaket selbst als auch detailierte Informationen darüber. Als Einstieg ist hier der User's Guide zu empfehlen. Für weitergehende Programmierarbeiten empfiehlt sich der Reference Guide. Hier können Sie auch dediziert Informationen über die einzelnen Klassen finden.
Auf den Rechnern im CIP-Pool finden Sie im Verzeichnis `$ROOTSYS/tutorials` zahlreiche Beispiele.

Histogramme

1. Öffnen Sie eine ROOT-Session:
   `root` (oder `root -l`)
   Sie können eine Root-Session jederzeit durch die Eingabe von .q beenden.

2. Öffnen Sie einen "Canvas":
   `TCanvas* c1 = new TCanvas("c1","mein canvas",800,500);`
   Was passiert, wenn Sie die Parameter 800 und 500 ändern?
   (Zur Information: Die Dokumentation der Klasse TCanvas finden Sie unter http://root.cern.ch/root/html532/TCanvas.html.)

3. Erzeugen Sie ein Histogramm und füllen Sie es mit einigen Werten zwischen 0 und 10:
   ```
   TH1F* h1 = new TH1F(''h1'',''mein histo'',10,0,10)
   h1->Fill(3.2)
   h1->Fill(7)
   h1->Fill(3.6)
   h1->Fill(4.2)
   ...
   ```

4. Beschriften Sie x und y-Achse:
   ```
   h1->SetXTitle(''x'')
   h1->SetYTitle(''Anzahl'')
   ```

5. Plotten Sie das Histogramm:
   `h1->Draw()`
   (Zur Information: Die Dokumentation der Klasse TH1F finden Sie unter http://root.cern.ch/root/html532/TH1F.html. Die Klasse TH1F leitet sich von der Klasse TH1 für eindimensionale Histogramme ab, d.h. wesentliche Methoden sind dort beschrieben, z.B. die hier verwendete `Fill()` Funktion.)

6. Schreiben Sie die obigen Befehle in ein Makro `histo.C`. Vergessen Sie nicht ein Semikolon am Ende jedes Befehls zu setzen. Fügen Sie in der ersten Zeile ein ‘‘{‘‘ und in die lezte Zeile ein ‘‘}’’ ein. Die Datei `histo.C` sieht also etwa so aus:

   ```
   {
   TCanvas* c1 = new TCanvas("c1","␣mein␣canvas␣2",800,500);
   TH1F* h1 = new  TH1F("h1","mein␣histo",10,0,10);
   .
   .
   .
   h1->Draw();
   }
   ```

7. Dieses Makro können Sie jetzt entweder von der LINUX shell mit
   `root -l histo.C`
   oder innerhalb einer ROOT session mit
   `.x histo.C`
   ausführen.

8. Geben Sie
   `c1->SetLogx(1)`
   ein. Wie könnte der Befehl lauten um wieder zu einer linearen x-Achse zu kommen?

9. Probieren Sie auch durch Operationen mit der Maus das Aussehen des Histogramms zu verändern. Hierfür öffnet sich ein Auswahlmenü bei Betätigung der rechten Maustaste in Abhängigkeit von der Position des Mauszeigers. Zum Beispiel können Sie die Darstellung der x-Achse modifizieren, wenn der Mauszeiger kurz unterhalb der x Achse steht.

10. Tipp: Durch Verwendung der Tabulator-Taster können Sie auflisten welche Befehle (Methoden) bei dem gerade verwendeten Objekt möglich sind, probieren Sie z. B.

    ```
    TH1F* h1 = new  TH1F("h1","mein␣histo",10,0,10);
    h1->Set [TAB]
    ```

Schöner plotten

1. Laden Sie von **eCampus** die Datei `.rootlogon.C` herunter und speichern Sie sie in Ihrem aktuellen Arbeitsverzeichnis.

2. Öffnen Sie jetzt eine ROOT session. Sie erhalten die Meldung:

   ```
   Welcome to my rootlogon.C
   For approved plots use: gROOT->SetStyle("BABAR");
   ```

3. Führen Sie jetzt in ROOT den Befehl
`gROOT->SetStyle("BABAR")`
aus.

4. Führen Sie jetzt nochmals das Makro `histo.C` aus:
`.x histo.C`
Wie Sie nun nochmals ein ROOT-Makro ausführen, sehen Sie, dass Ihr Historamm jetzt größere, besser lesbare Achsenbeschriftungen hat. Diese Parameter wurden mit den Befehlen in der Datei `.rootlogon.C` geändert und können nach eigenem Geschmck angepasst werden indem man die Werte und Befehle in .rootlogon.C ändert.

Berichtsaufgaben

Graphen

1. Auf **eCampus** finden Sie die Datei `data_fehler_bericht.dat` welche nochmals die Daten aus Übung 7 enthält, jedoch um zwei Spalten für die Messfehler erweitert. Erstellen Sie jetzt einen Graphen mithilfe der ROOT-Klasse `TGraphErrors`. Mit dieser Klasse kann man y-Werte gegen die x-Werte mit Fehlerbalken plotten. Ein möglicher Konstruktoraufruf der `TGraphErrors`-Klasse lautet:

```
TGraphErrors* g1 = new TGraphErrors(n,x,y,ex,ey);
```

wobei `n` für die Anzahl der Punkte steht. Die Felder `x,y,ex,ey` enthalten die Werte der entsprechenden Variable. Um die Daten aus der Datei hiermit darzustellen, müssen die Werte aus der Datei erst eingelesen werden und dann an den Konstruktor übergeben werden.
Eine weitere Möglichkeit ist die direkte Angabe der einzulesenden Datei im Konstruktor, wenn die Datei ausreichend formatiert ist:

```
TGraphErrors *tge = new TGraphErrors("data_fehler_bericht.dat")
```

2. Mit

```
g1->Draw("ap");
```

wird der Graph geplottet. Die Optionen "ap" stehen für:

- a = Zeichne Koordinatenachsen und
- p = Zeichne Punkte

Weiter Optionen Sind in der Beschreibung der Klasse TGraphPainter zu finden.

3. Mit dem Befehl `g1->Fit(''pol1'')` wird eine Gerade an die Daten angepasst. Steigung und Schnittpunkt mit der y-Achse werden auf dem Bildschirm ausgegeben. Zusätzlich lassen sich die Fitparamter im Canvas anzeigen wenn Sie den Menüpunkt `Options->Fit Parameters` anwählen.

Erzeugen Sie einen Plot der Daten mit der angepaten Fitfunktion und den Parametern für Steigung und y-Achsenabschnitt.