Object-Oriented Application to Compute Weighted Average

Shahram Rahatlou

SAPIENZA
UNIVERSITÀ DI ROMA

Corso di Programmazione++

Roma, 30-31 March 2009

Possible Implementation of Our Exercise with Classes

```
// app1.cc
#include <vector>

class Datum; // basic data object
class InputService; // class dedicated to handle input of data
class Calculator; // impelments various algorithms
class Result; // how is Result different from Datum ?

int main() {

  InputService input;
  std::vector<Datum> dati = input.readDataFromUser();

  Calculator calc;
  calc.setData( dati );

  Result r1 = calc.weightedAverage();
  Result r2 = calc.arithmeticAverage();
  Result r3 = calc.geometricAverage();
  Result r3 = calc.fancyAverage();

  r1.display();

  return 0;
}
```

This code does not compile.

What is missing? ☺

Interface of Classes for Weighted Average

```
#ifndef Calculator_h
#define Calculator_h

#include <vector>
#include "Datum.h"
#include "Result.h"

class Calculator {
 public:
  Calculator();
  void setData(std::vector<Datum>& data);

  Result weightedAverage();
  Result arithmeticAverage();
  Result geometricAverage();
  Result fancyAverage();

private:
  std::vector<Datum> data_;
};
#endif
```

```
#ifndef Result_h
#define Result_h
class Result {
  public:
    Result();
    Result(double x, double y);
    Result(const Result& Result);
    double mean() { return mean_; }
    double stdDev() { return stdDev_; }
    double significance();
    void display();
  private:
    double mean_;
    double stdDev_;
};
#endif
```

You see the interface but don't know how the methods are implemented!

```
#ifndef InputService_h
#define InputService_h
#include <vector>
#include "Datum.h"

class InputService {
 public:
   InputService();
   std::vector<Datum> readDataFromUser();
 private:
};
#endif
```

```
#ifndef Datum_h
#define Datum_h
// Datum.h
#include <iostream>
using namespace std;

class Datum {
  public:
    Datum();
    Datum(double x, double y);
    Datum(const Datum& datum);
    double value() { return value_; }
    double error() { return error_; }
    double significance();
  private:
    double value_;
    double error_;
};
#endi
```

Application for Weighted Average

```
// wgtavg.cpp
#include <vector>

#include “Datum.h” // data objects
#include “InputService.h” // class dedicated to handle input of data
#inlcude “Calculator.h” // implements various algorithms
#include “Result.h” // how is Result different from Datum ?

int main() {

  InputService input;
  std::vector<Datum> dati = input.readDataFromUser();

  Calculator calc;
  calc.setData( dati );

  Result r1 = calc.weightedAverage();
  Result r2 = calc.arithmeticAverage();
  Result r3 = calc.geometricAverage();
  Result r3 = calc.fancyAverage();

  r1.display();

  return 0;
}
```

```
$ g++ -c InputService.cc
$ g++ -c Datum.cc
$ g++ -c InputService.cc
$ g++ -c Calculator.cc
$ g++ -c Result.cc
$ g++ -o wgtavg wgtavg.cpp InputService.o Datum.o Result.o Calculator.o
```

Today's Lab Session

- Use these 4 classes to write your application
- Take the interface and implement the functions
- Encapsulate your algorithms into methods of Calculator
- Adapt the exchange of data between functions to use Datum and Result classes