KAUNAS UNIVERSITY OF TECHNOLOGY

Faculty of Mathematics and Natural Sciences

Fundamentals of Object Programming (P175B013)

Project Report

Prepared by:

student Stephen Oluwabukunmi Adesina,

group MGMFNA-1

Supervisor:

Liudas Motiejūnas

KAUNAS

CONTENT

1. Data Grouping

1.1. Problem

· Create a class Coin to store the variables for denomination and weight of the coin. The student found the treasure of old Greek coins. What the total monetary value of coins is?

· Add the variables to the class Coin to store the diameter and thickness of the coin. Augment the initial data. Create a class Cylinder to store the variables for the length and diameter of cylinder. How many coins of each denomination separately can accommodate this cylinder?

1.2. Source Code

Coin.h

```
#pragma once
#include <string>

class Coin {

      public:
      std::string denomination;
    double weight;
    double diameter;
    double thickness;
      double no_of_coins;
};
```

Coin.cpp

```
#include "coin.h"
```

Cylinder.cpp

```
#include "cylinder.h"
```

Cylinder.h

```
#pragma once

class Cylinder{
    public:
       double length;
       double diameter;
};
```

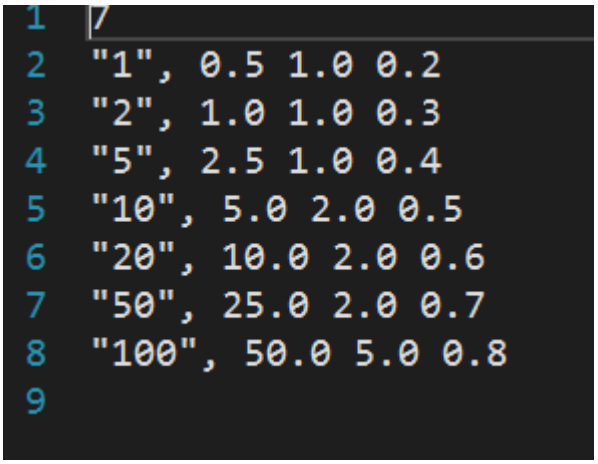
pr-1.cpp

```
#include <map>
#include "coin.h"
#include "cylinder.h"
#include <cmath>
#include <algorithm>
#include <fstream>
#include <stdio.h>
#include <sstream>
#include <iostream>
#include <string>
#include <cmath>
#include <iomanip>


//prototype of functions so the functions can stay below the main but still run

void ReadData(std::string file, Coin coins[], int &nc);
void WriteData(std::string file, Coin coins[], int nc, double value_of_coins);
void  NoOfCoinsInCylinder(Coin coins[], int nc, Cylinder _Cylinder);
double ValueOfCoins(Coin coins[], int nc);
double calcCyVolume(double diameter, double height);

int main()
{
int number_of_coins;
    Coin coins[10];
       Cylinder _Cylinder;
       int nc;
       double value_of_coins;

       std::cout << "Enter the diameter then height of the cylinder:" ;
   std::cin >> _Cylinder.diameter >> _Cylinder.length;



   //creating an empty file
   std::ofstream fd("Result.txt");
       fd.close();

       ReadData("Data.txt", coins, nc);
       value_of_coins = ValueOfCoins(coins, nc);
       NoOfCoinsInCylinder(coins,nc, _Cylinder);
       WriteData("Result.txt", coins, nc, value_of_coins);
       std::cout << "Done" <<std::endl;
       std::cin.get();
    return 0;
}


void ReadData(std::string file, Coin coins[], int &nc){
```

```
	std::ifstream fd(file);
fd >> nc; fd.ignore();
for (int i=0; i<nc; i++){
	std::getline(fd,coins[i].denomination,','); fd >> std::ws;
	coins[i].denomination.erase(
	std::remove(coins[i].denomination.begin(), coins[i].denomination.end(), '\"'),
	coins[i].denomination.end());
	fd >> coins[i].weight;
	fd >> coins[i].diameter;
	fd >> coins[i].thickness;
	fd.ignore();
}
fd.close();
}

void WriteData(std::string file, Coin coins[], int nc, double value_of_coins){
	std::ofstream ft(file, std::ios::app);
ft.setf(std::ios::fixed); ft.setf(std::ios::left);
ft << "Number of coins: " << nc << std::endl;
ft << "Value of coins: " << value_of_coins << std::endl;
ft << "List of Coins: \n";
ft <<
"-------------------------------------------------------------------------------\n";
ft << "|     Denomination |   Weight  |   Diameter  |   Thickness | Number of coins
|\n";
ft <<
"-------------------------------------------------------------------------------\n";
	for (int i=0; i<nc; i++){
ft << "|   " << std::setw(16) << coins[i].denomination << " |   " <<
std::setprecision(1) << std::setw(5) << coins[i].weight << "   |   " << std::setw(11) <<
coins[i].diameter << " |   " << std::setw(7) << coins[i].thickness << "  |" <<
coins[i].no_of_coins << "  | " << std::endl;
	}
ft <<
"-------------------------------------------------------------------------------\n"
;
	ft.close();
}

double calcCyVolume(double diameter, double height){
 double r = diameter/2;
 return M_PI * r* r* height;
}

double ValueOfCoins(Coin coins[],int nc){
	double coin_sum;
	int c;

		for(int i=0; i<nc; i++){
	c = std::stoi(coins[i].denomination);
	coin_sum += c;

		}
		coin_sum /= 100;
```

```
            std::cout << "The total value of the coins are: " << std::setprecision(3)<<
coin_sum << " Euros" <<std::endl;
            return coin_sum;
}

//void  NoOfCoinsInCylinder(Coin coins[],int nc, Cylinder _Cylinder){
//
//   double vol_of_cyl = calcCyVolume(_Cylinder.diameter, _Cylinder.length);
//   for (int i=0; i<nc; i++){
//     if (coins[i].diameter <= _Cylinder.diameter && coins[i].thickness <=
_Cylinder.length){
//   double vol_of_coin = calcCyVolume(coins[i].diameter, coins[i].thickness);
//     int no_of_coins = (int)vol_of_cyl/vol_of_coin;
//     coins[i].no_of_coins = no_of_coins;
//   std::cout << "The cylinder can contain " << no_of_coins << " coins of denomination
" << coins[i].denomination << std::endl;
//   }
//
//     else{
//     std::cout << "The diameter of " << coins[i].denomination << " is too big to enter
the cylinder.\n";
//
//     }
//   }
//}

void  NoOfCoinsInCylinder(Coin coins[],int nc, Cylinder _Cylinder){


   for (int i=0; i<nc; i++){
      if (coins[i].diameter <= _Cylinder.diameter && coins[i].thickness <=
_Cylinder.length){
            int no_of_coins = (int)_Cylinder.length/coins[i].thickness;
            coins[i].no_of_coins = no_of_coins;
            std::cout << "The cylinder can contain " << no_of_coins << " coins of
denomination " << coins[i].denomination << std::endl;
   }

      else{
            std::cout << "The diameter of " << coins[i].denomination << " is too big to
enter the cylinder.\n";

      }
   }
}
```

1.3. Initial Data and Results



7
"1", 0.5 1.0 0.2
"2", 1.0 1.0 0.3
"5", 2.5 1.0 0.4
"10", 5.0 2.0 0.5
"20", 10.0 2.0 0.6
"50", 25.0 2.0 0.7
"100", 50.0 5.0 0.8

```
Number
of
coins: 7
```

List of Coins:

Denomination	Weight	Diameter	Thickness
1	0.5	1.0	0.2
2	1.0	1.0	0.3
5	2.5	1.0	0.4
10	5.0	2.0	0.5
20	10.0	2.0	0.6
50	25.0	2.0	0.7
100	50.0	5.0	0.8