#include <string.h>

#include <iostream>

#include "package.h"

Package::Package (unsigned int peso, unsigned int custo\_por\_quilo, std::string nome, std::string endereco){

this-> peso = peso;

this-> custo\_por\_quilo = custo\_por\_quilo;

this -> endereco=endereco;

this -> nome=nome;

}

double Package::calculate\_cost(){

return peso\*custo\_por\_quilo;

}

unsigned int Package::get\_peso(){

return peso;

}

unsigned int Package::get\_custo\_por\_quilo(){

return custo\_por\_quilo;

}

Package::~Package(){

}

#ifndef PACKAGE\_H

#define PACKAGE\_H

#include <string.h>

#include <iostream>

class Package {

protected:

std::string nome;

std::string endereco;

unsigned int peso;

unsigned int custo\_por\_quilo;

public:

Package(unsigned int peso, unsigned int custo\_por\_quilo, std::string nome, std::string endereco);

~Package();

virtual double calculate\_cost();

unsigned int get\_peso();

unsigned int get\_custo\_por\_quilo();

};

#endif

#include <string.h>

#include <iostream>

#include "twodaypackage.h"

TwoDayPackage::TwoDayPackage (unsigned int peso, unsigned int custo\_por\_quilo,

unsigned int taxa\_de\_envio, std::string nome, std::string endereco):

Package(peso,custo\_por\_quilo, nome, endereco){

this-> taxa\_de\_envio = taxa\_de\_envio;

}

double TwoDayPackage::calculate\_cost(){

return Package::calculate\_cost()+taxa\_de\_envio;

}

#ifndef TWO\_DAY\_PACKAGE\_H

#define TWO\_DAY\_PACKAGE\_H

#include <string.h>

#include <iostream>

#include "package.h"

class TwoDayPackage: public Package {

private:

unsigned int taxa\_de\_envio;

public:

TwoDayPackage(unsigned int peso, unsigned int custo\_por\_quilo, unsigned int taxa\_de\_envio,std::string nome, std::string endereco);

double virtual calculate\_cost();

};

#endif

#include <string.h>

#include <iostream>

#include "overnightpackage.h"

OverNightPackage::OverNightPackage(unsigned int peso, unsigned int custo\_por\_quilo, unsigned int taxa\_adicional\_por\_quilo, std::string nome, std::string endereco):

Package(peso,custo\_por\_quilo, nome, endereco){

this-> taxa\_adicional\_por\_quilo = taxa\_adicional\_por\_quilo;

}

double OverNightPackage::calculate\_cost(){

return peso\*(custo\_por\_quilo + taxa\_adicional\_por\_quilo);

}

#include <string.h>

#include <iostream>

#include "package.h"

class OverNightPackage: public Package{

private:

unsigned int taxa\_adicional\_por\_quilo;

public:

OverNightPackage(unsigned int peso, unsigned int custo\_por\_quilo, unsigned int taxa\_adicional\_por\_quilo, std::string nome, std::string endereco);

double virtual calculate\_cost();

};

#include <vector>

#include <map>

#include <string>

#include "package.h"

class DestinationPackages {

private:

std::map<std::string,std::vector<Package\*>>\_package\_list;

public:

DestinationPackages();

void add\_package(Package \*g);

double custo\_total();

double custo\_total(std::string nome);

double get\_nome();

};

#include <vector>

#include <map>

#include <string>

#include "destinationpackages.h"

DestinationPackages::DestinationPackages() {

this->\_package\_list = std::map<std::string,std::vector<Package\*>>();

}

void DestinationPackages::add\_package(Package \*g){

if (\_package\_list.count(g->get\_nome()) == 0)

{

\_package\_list[g->get\_nome()] == std::vector<Package\*>();

}

\_package\_list[g->get\_nome()].push\_back(g);

}

double DestinationPackages::custo\_total(){

double total = 0;

for(std::map<std::string,std::vector<Package\*>>::iterator it=\_package\_list.begin(); it!=\_package\_list.end(); ++it)

{

for(std::vector<Package\*>::iterator vec = it->second.begin() ; vec != it->second.end(); ++vec)

{

total = total + (\*vec)->calculate\_cost();

}

}

return total;

}

double DestinationPackages::custo\_total(std::string nome){

double total = 0;

if(\_package\_list.count(nome) > 0)

{

for( std::vector<Package\*>::iterator vec = \_package\_list[nome].begin(); vec != \_package\_list[nome].end(); ++vec)

{

total=total + (\*vec)->calculate\_cost();

}

}

return total;

}