федеральное ГОСУДАРСТВЕННОЕ Бюджетное ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

**"САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ "ЛЭТИ" им.В.И.Ульянова (Ленина)" (СПбГЭТУ)**

Кафедра МОЭВМ

**Отчет по лабораторной работе №2**

«Реализация программы на языке C++»

Выполнил: Морозов А.В.

Факультет КТИ

Группа № 0304

Преподаватель: Смольянинов А.В.

Санкт-Петербург

2013

## Проектирование системы описаний классов как системы файлов на языке C++

main.cpp

torch.h

cyclic.hpp

intlist.hpp

burner.h

bowl.h

body.h

handle.h

point.h

burner.cpp

bowl.cpp

body.cpp

handle.cpp

point.cpp

rectangle.h

trapezium.h

rectangle.cpp

trapezium.cpp

## Разработка методики тестирования классов и тестовых наборов данных

Тестирование производится по методу «чёрного ящика».

Для тестирования используются следующие методы:

1. перемещение одного или нескольких изображений на сдвиг ;
2. перемещение всех изображений на сдвиг ;
3. перемещение одного или нескольких изображений в указанную точку;
4. перемещение всех изображений в указанную точку.

В процессе тестирования формируется разбиение исходного списка фигур на поднаборы согласно критерию, указанному в проектном решении.

Для тестирования используются наборы данных двух видов:

1. пустой набор изображений => результат: пустые поднаборы изображений;
2. набор изображений, содержащий 1 и более изображений.

Примеры наборов изображений:

* {} – пустой набор;
* – изображение факела высоты 2, располагающегося в точке с координатами ;
* – изображения факелов высотой 14.5, 1 и 10, располагающиеся в точках с координатами , и соответственно.

## Разработка внешней спецификации тестирующей программы

Методы, перечисленные в п. 2, выполняются посредством ввода команд в указанной форме:

|  |  |
| --- | --- |
| moveby {i j…} {dx dy} | Осуществляет сдвиг изображений {i, j,…} на . |
| moveby all {dx dy} | Осуществляет сдвиг всех изображений на . |
| moveto {i j…} | Перемещает изображения {i j…} в указанную точку. |
| moveto all | Перемещает все изображения в указанную точку. |
| detailed i | Выводит на экран подробную информацию об изображении i. |

## Количественные характеристики программы

Сведения о тексте:

|  |  |  |  |
| --- | --- | --- | --- |
| Имя файла | Количество строк-комментариев | Количество строк кода | Количество пустых строк |
| main.cpp | 6 | 339 | 45 |
| cyclic.hpp | 6 | 208 | 40 |
| intlist.hpp | 11 | 243 | 50 |
| point.h | 6 | 34 | 12 |
| point.cpp | 7 | 147 | 30 |
| torch.h | 6 | 40 | 9 |
| torch.cpp | 6 | 115 | 22 |
| burner.h | 6 | 21 | 8 |
| burner.cpp | 6 | 31 | 9 |
| bowl.h | 6 | 21 | 8 |
| bowl.cpp | 6 | 30 | 9 |
| body.h | 6 | 21 | 8 |
| body.cpp | 6 | 30 | 9 |
| handle.h | 6 | 22 | 8 |
| handle.cpp | 6 | 34 | 10 |
| rectangle.h | 6 | 21 | 6 |
| rectangle.cpp | 6 | 21 | 6 |
| trapezium.h | 6 | 21 | 6 |
| trapezium.cpp | 6 | 22 | 6 |

Сведения о классах:

* общее количество классов: 16
* количество новых классов: 7

|  |  |  |
| --- | --- | --- |
| Название класса | Длина текста описания класса | Общее количество функций, описанных вне класса |
| Rectangle | 16 | 6 |
| Trapezium | 18 | 7 |
| Burner | 17 | 6 |
| Bowl | 17 | 6 |
| Body | 17 | 6 |
| Handle | 18 | 7 |
| Torch | 32 | 13 |

Сведения о файлах:

* общее количество заголовочных файлов: 13
* общее количество файлов-кодов: 9

## Вывод

Созданные классы соответствуют заданию и спецификации, и корректно выполняют все поставленные задачи. Тестирующая программа полностью работоспособна и позволяет проверить все возможности разработанных классов.

Достоинство программы состоит в том, что она построена на классах и разбита на модули, с разделением описания и реализации. Разделение классов – составных частей изображения (горелка, чаша, корпус, рукоятка), и классов – представлений данных частей (прямоугольник, трапеция) делает их подходящими для использования в других программах.

## Список литературы

1. Herb Sutter “Exceptional C++: 47 Engineering Puzzles, Programming Problems, and Solutions”, Addison-Wesley Professional, 1999 г. – 240 с.

## Приложение 1

#ifndef RECTANGLE\_H

#define RECTANGLE\_H

/\*

\* File: rectangle.h

\* Declaration of rectangle.

\* Author: Morozov Alexey

\* Created on: May 06, 2013; Version: 2

\*/

#include <iostream.h>

#include "bccomp.h"

#include "point.h"

class Rectangle

**{**

private**:**

Point topLeft**;**

float width**;**

float height**;**

public**:**

Rectangle**(**const Point **&**bottomCenter**,** float width**,** float height**);**

Rectangle**(**const Rectangle **&**r**);**

**~**Rectangle**();**

Point **\*** TopLeft**()** **{** **return** **&**topLeft**;** **}**

float GetWidth**()** const **{** **return** width**;** **}**

float GetHeight**()** const **{** **return** height**;** **}**

**};**

ostream **&** **operator** **<<(**ostream **&**os**,** const Rectangle **&**r**);**

#endif

/\*

\* File: rectangle.cpp

\* Definition of rectangle.

\* Author: Morozov Alexey

\* Created on: May 07, 2013; Version: 2

\*/

#include "rectan~1.h"

#include <assert.h>

#include <iostream.h>

Rectangle**::**Rectangle**(**const Point **&**bottomCenter**,** float width**,** float height**)**

**:** topLeft**(**bottomCenter**.**MoveBy**(-**width **\*** 0.5**,** height**)),**

width**(**width**),** height**(**height**)**

**{**

assert**(**width **>** 0 **&&** height **>** 0**);**

**}**

Rectangle**::**Rectangle**(**const Rectangle **&**r**)**

**:** topLeft**(**r**.**topLeft**),** width**(**r**.**width**),** height**(**r**.**height**)**

**{**

**}**

Rectangle**::~**Rectangle**()**

**{**

**}**

ostream **&** **operator** **<<(**ostream **&**os**,** const Rectangle **&**r**)**

**{**

**return** os **<<** "Rect[Point" **<<** **\***r**.**TopLeft**()** **<<** ", "

**<<** r**.**GetWidth**()** **<<** ", " **<<** r**.**GetHeight**()** **<<** "]"**;**

**}**

#ifndef TRAPEZIUM\_H

#define TRAPEZIUM\_H

/\*

\* File: trapezium.h

\* Declaration of trapezium.

\* Author: Morozov Alexey

\* Created on: May 06, 2013; Version: 2

\*/

#include <iostream.h>

#include "bccomp.h"

#include "point.h""

class Trapezium

**{**

private**:**

Point topLeft**;**

float height**;**

float side1**,** side2**;**

public**:**

Trapezium**(**const Point **&**bottomCenter**,** float side1**,** float side2**,** float height**);**

Trapezium**(**const Trapezium **&**t**);**

**~**Trapezium**();**

Point **\*** TopLeft**()** **{** **return** **&**topLeft**;** **}**

float GetHeight**()** const **{** **return** height**;** **}**

float GetSide1**()** const **{** **return** side1**;** **}**

float GetSide2**()** const **{** **return** side2**;** **}**

**};**

ostream **&** **operator** **<<(**ostream **&**os**,** const Trapezium **&**t**);**

#endif

/\*

\* File: trapezium.cpp

\* Definition of trapezium.

\* Author: Morozov Alexey

\* Created on: May 07, 2013; Version: 2

\*/

#include "trapez~1.h"

#include <assert.h>

#include <iostream.h>

Trapezium**::**Trapezium**(**const Point **&**bottomCenter**,** float side1**,** float side2**,** float height**)**

**:** topLeft**(**bottomCenter**.**MoveBy**(-**side1 **\*** 0.5**,** height**)),**

height**(**height**),** side1**(**side1**),** side2**(**side2**)**

**{**

assert**(**side1 **>** 0 **&&** side2 **>** 0 **&&** height **>** 0**);**

**}**

Trapezium**::**Trapezium**(**const Trapezium **&**t**)**

**:** topLeft**(**t**.**topLeft**),** height**(**t**.**height**),** side1**(**t**.**side1**),** side2**(**t**.**side2**)**

**{**

**}**

Trapezium**::~**Trapezium**()**

**{**

**}**

ostream **&** **operator** **<<(**ostream **&**os**,** const Trapezium **&**t**)**

**{**

**return** os **<<** "Trapezium[Point" **<<** **\***t**.**TopLeft**()** **<<** ", sides: ("

**<<** t**.**GetSide1**()** **<<** ", " **<<** t**.**GetSide2**()** **<<** "), height: "

**<<** t**.**GetHeight**()** **<<** "]"**;**

**}**

#ifndef BURNER\_H

#define BURNER\_H

/\*

\* File: burner.h

\* Declaration of burner.

\* Author: Morozov Alexey

\* Created on: May 06, 2013; Version: 2

\*/

#include <iostream.h>

#include "bccomp.h"

#include "point.h"

#include "rectan~1.h"

#include "cyclic.hpp"

class Burner

**{**

private**:**

Rectangle box**;**

public**:**

Burner**(**const Point **&**position**,** float length**);**

Burner**(**const Burner **&**b**);**

**~**Burner**();**

void GetPoints**(**CyclicList**<**Point **\*>** **&**list**);**

float GetLength**()** const**;**

ostream **&** PrintTo**(**ostream **&**os**)** const**;**

**};**

ostream **&** **operator** **<<(**ostream **&**os**,** const Burner **&**b**);**

#endif

/\*

\* File: burner.cpp

\* Definition of burner.

\* Author: Morozov Alexey

\* Created on: May 07, 2013; Version: 2

\*/

#include "burner.h"

#include <assert.h>

#include <iostream.h>

Burner**::**Burner**(**const Point **&**position**,** float length**)**

**:** box**(**position**,** length**,** length**)**

**{**

assert**(**length **>** 0**);**

**}**

Burner**::**Burner**(**const Burner **&**b**)**

**:** box**(**b**.**box**)**

**{**

**}**

Burner**::~**Burner**()**

**{**

**}**

void Burner**::**GetPoints**(**CyclicList**<**Point **\*>** **&**list**)**

**{**

list**.**AddLast**(**box**.**TopLeft**());**

**}**

float Burner**::**GetLength**()** const

**{**

**return** box**.**GetWidth**();**

**}**

ostream **&** Burner**::**PrintTo**(**ostream **&**os**)** const

**{**

**return** os **<<** "Burner[" **<<** box **<<** "]"**;**

**}**

ostream **&** **operator** **<<(**ostream **&**os**,** const Burner **&**b**)**

**{**

**return** b**.**PrintTo**(**os**);**

**}**

#ifndef BOWL\_H

#define BOWL\_H

/\*

\* File: bowl.h

\* Declaration of bowl.

\* Author: Morozov Alexey

\* Created on: May 06, 2013; Version: 2

\*/

#include <iostream.h>

#include "bccomp.h"

#include "point.h"

#include "trapez~1.h"

#include "cyclic.hpp"

class Bowl

**{**

private**:**

Trapezium trap**;**

public**:**

Bowl**(**const Point **&**position**,** float height**);**

Bowl**(**const Bowl **&**b**);**

**~**Bowl**();**

void GetPoints**(**CyclicList**<**Point **\*>** **&**list**);**

float GetHeight**()** const**;**

ostream **&** PrintTo**(**ostream **&**os**)** const**;**

**};**

ostream **&** **operator** **<<(**ostream **&**os**,** const Bowl **&**b**);**

#endif

/\*

\* File: bowl.cpp

\* Definition of bowl.

\* Author: Morozov Alexey

\* Created on: May 07, 2013; Version: 2

\*/

#include "bowl.h"

#include <assert.h>

#include <iostream.h>

Bowl**::**Bowl**(**const Point **&**position**,** float height**)**

**:** trap**(**position**,** height **\*** 2**,** height **\*** 1.5f**,** height**)**

**{**

**}**

Bowl**::**Bowl**(**const Bowl **&**b**)**

**:** trap**(**b**.**trap**)**

**{**

**}**

Bowl**::~**Bowl**()**

**{**

**}**

void Bowl**::**GetPoints**(**CyclicList**<**Point **\*>** **&**list**)**

**{**

list**.**AddLast**(**trap**.**TopLeft**());**

**}**

float Bowl**::**GetHeight**()** const

**{**

**return** trap**.**GetHeight**();**

**}**

ostream **&** Bowl**::**PrintTo**(**ostream **&**os**)** const

**{**

**return** os **<<** "Bowl[" **<<** trap **<<** "]"**;**

**}**

ostream **&** **operator** **<<(**ostream **&**os**,** const Bowl **&**b**)**

**{**

**return** b**.**PrintTo**(**os**);**

**}**

#ifndef BODY\_H

#define BODY\_H

/\*

\* File: body.h

\* Declaration of body.

\* Author: Morozov Alexey

\* Created on: May 06, 2013; Version: 2

\*/

#include <iostream.h>

#include "bccomp.h"

#include "point.h"

#include "trapez~1.h"

#include "cyclic.hpp"

class Body

**{**

private**:**

Trapezium trap**;**

public**:**

Body**(**const Point **&**position**,** float height**);**

Body**(**const Body **&**b**);**

**~**Body**();**

void GetPoints**(**CyclicList**<**Point **\*>** **&**list**);**

float GetHeight**()** const**;**

ostream **&** PrintTo**(**ostream **&**os**)** const**;**

**};**

ostream **&** **operator** **<<(**ostream **&**os**,** const Body **&**b**);**

#endif

/\*

\* File: body.cpp

\* Definition of body.

\* Author: Morozov Alexey

\* Created on: May 07, 2013; Version: 2

\*/

#include "body.h"

#include <assert.h>

#include <iostream.h>

Body**::**Body**(**const Point **&**position**,** float height**)**

**:** trap**(**position**,** height **\*** 0.2f**,** height **\*** 0.1f**,** height**)**

**{**

**}**

Body**::**Body**(**const Body **&**b**)**

**:** trap**(**b**.**trap**)**

**{**

**}**

Body**::~**Body**()**

**{**

**}**

void Body**::**GetPoints**(**CyclicList**<**Point **\*>** **&**list**)**

**{**

list**.**AddLast**(**trap**.**TopLeft**());**

**}**

float Body**::**GetHeight**()** const

**{**

**return** trap**.**GetHeight**();**

**}**

ostream **&** Body**::**PrintTo**(**ostream **&**os**)** const

**{**

**return** os **<<** "Body[" **<<** trap **<<** "]"**;**

**}**

ostream **&** **operator** **<<(**ostream **&**os**,** const Body **&**b**)**

**{**

**return** b**.**PrintTo**(**os**);**

**}**

#ifndef HANDLE\_H

#define HANDLE\_H

/\*

\* File: handle.h

\* Declaration of handle.

\* Author: Morozov Alexey

\* Created on: May 06, 2013; Version: 2

\*/

#include <iostream.h>

#include "bccomp.h"

#include "point.h"

#include "rectan~1.h"

#include "cyclic.hpp"

class Handle

**{**

private**:**

Rectangle box**;**

public**:**

Handle**(**const Point **&**position**,** float width**);**

Handle**(**const Handle **&**h**);**

**~**Handle**();**

void GetPoints**(**CyclicList**<**Point **\*>** **&**list**);**

float GetWidth**()** const**;**

const Point GetPosition**()** const**;**

ostream **&** PrintTo**(**ostream **&**os**)** const**;**

**};**

ostream **&** **operator** **<<(**ostream **&**os**,** const Handle **&**h**);**

#endif

/\*

\* File: handle.cpp

\* Definition of handle.

\* Author: Morozov Alexey

\* Created on: May 07, 2013; Version: 2

\*/

#include "handle.h"

#include <assert.h>

#include <iostream.h>

Handle**::**Handle**(**const Point **&**position**,** float width**)**

**:** box**(**position**,** width**,** width **\*** 1.5f**)**

**{**

**}**

Handle**::**Handle**(**const Handle **&**h**)**

**:** box**(**h**.**box**)**

**{**

**}**

Handle**::~**Handle**()**

**{**

**}**

void Handle**::**GetPoints**(**CyclicList**<**Point **\*>** **&**list**)**

**{**

list**.**AddLast**(**box**.**TopLeft**());**

**}**

float Handle**::**GetWidth**()** const

**{**

**return** box**.**GetWidth**();**

**}**

const Point Handle**::**GetPosition**()** const

**{**

**return** **(\***box**.**TopLeft**()).**MoveBy**(**box**.**GetWidth**()** **\*** 0.5f**,** **-**box**.**GetHeight**());**

**}**

ostream **&** Handle**::**PrintTo**(**ostream **&**os**)** const

**{**

**return** os **<<** "Handle[" **<<** box **<<** "]"**;**

**}**

ostream **&** **operator** **<<(**ostream **&**os**,** const Handle **&**h**)**

**{**

**return** h**.**PrintTo**(**os**);**

**}**

#ifndef TORCH\_H

#define TORCH\_H

/\*

\* File: body.h

\* Declaration of torch.

\* Author: Morozov Alexey

\* Created on: May 06, 2013; Version: 2

\*/

#include <iostream.h>

#include "bccomp.h"

#include "point.h"

#include "cyclic.hpp"

#include "burner.h"

#include "bowl.h"

#include "body.h"

#include "handle.h"

class Torch

**{**

private**:**

static int nextID**;**

static int currentCount**;**

int id**;**

CyclicList**<**Point **\*>** points**;**

Burner burner**;**

Bowl bowl**;**

Body body**;**

Handle handle**;**

public**:**

Torch**();**

Torch**(**const Point **&**position**,** float height**);**

Torch**(**const Torch **&**t**);**

**~**Torch**();**

const Burner **&** GetBurner**()** const**;**

const Bowl **&** GetBowl**()** const**;**

const Body **&** GetBody**()** const**;**

const Handle **&** GetHandle**()** const**;**

const Point GetPosition**()** const**;**

void MoveTo**(**const Point **&**p**);**

void MoveBy**(**float dx**,** float dy**);**

ostream **&** PrintDetailedTo**(**ostream **&**os**)** const**;**

ostream **&** PrintPointsTo**(**ostream **&**os**)** const**;**

**};**

bool **operator** **==(**const Torch **&**t1**,** const Torch **&**t2**);**

bool **operator** **!=(**const Torch **&**t1**,** const Torch **&**t2**);**

ostream **&** **operator** **<<(**ostream **&**os**,** const Torch **&**t**);**

#endif

/\*

\* File: torch.cpp

\* Definition of torch.

\* Author: Morozov Alexey

\* Created on: May 07, 2013; Version: 2

\*/

#include "torch.h"

#include <assert.h>

#include <iostream.h>

int Torch**::**nextID **=** 0**;**

int Torch**::**currentCount **=** 0**;**

Torch**::**Torch**()**

**:** burner**(**Point**(),** 1**),** bowl**(**Point**(),** 1**),**

body**(**Point**(),** 1**),** handle**(**Point**(),** 1**),**

id**(**nextID**++)**

**{**

currentCount**++;**

cout **<<** " -- temp Torch[ID: " **<<** id **<<** "], total = " **<<** currentCount **<<** endl**;**

**}**

Torch**::**Torch**(**const Point **&**position**,** float height**)**

**:** burner**(**position**.**MoveBy**(**0**,** **(**height **/** 14.5f**)** **\*** 13.5f**),** height **/** 14.5f**),**

bowl**(**position**.**MoveBy**(**0**,** **(**height **/** 14.5f**)** **\*** 11.5f**),** **(**height **/** 14.5f**)** **\*** 2**),**

body**(**position**.**MoveBy**(**0**,** **(**height **/** 14.5f**)** **\*** 1.5f**),** **(**height **/** 14.5f**)** **\*** 10**),**

handle**(**position**,** height **/** 14.5f**),**

id**(**nextID**++)**

**{**

currentCount**++;**

cout **<<** " -- new Torch[ID: " **<<** id **<<** "], total = " **<<** currentCount **<<** endl**;**

burner**.**GetPoints**(**points**);**

bowl**.**GetPoints**(**points**);**

body**.**GetPoints**(**points**);**

handle**.**GetPoints**(**points**);**

**}**

Torch**::**Torch**(**const Torch **&**t**)**

**:** burner**(**t**.**burner**),** bowl**(**t**.**bowl**),**

body**(**t**.**body**),** handle**(**t**.**handle**),**

id**(**nextID**++)**

**{**

currentCount**++;**

cout **<<** " -- copy Torch[ID: " **<<** id **<<** "], total = " **<<** currentCount **<<** endl**;**

burner**.**GetPoints**(**points**);**

bowl**.**GetPoints**(**points**);**

body**.**GetPoints**(**points**);**

handle**.**GetPoints**(**points**);**

**}**

Torch**::~**Torch**()**

**{**

currentCount**--;**

cout **<<** " -- delete Torch[ID: " **<<** id **<<** "], total = " **<<** currentCount **<<** endl**;**

**}**

const Burner **&** Torch**::**GetBurner**()** const

**{**

**return** burner**;**

**}**

const Bowl **&** Torch**::**GetBowl**()** const

**{**

**return** bowl**;**

**}**

const Body **&** Torch**::**GetBody**()** const

**{**

**return** body**;**

**}**

const Handle **&** Torch**::**GetHandle**()** const

**{**

**return** handle**;**

**}**

const Point Torch**::**GetPosition**()** const

**{**

**return** handle**.**GetPosition**();**

**}**

void Torch**::**MoveTo**(**const Point **&**p**)**

**{**

Point diff **=** p **-** GetPosition**();**

CyclicList**<**Point **\*>::**Iterator it **=** points**.**begin**();**

**for** **(;** it **!=** points**.**end**();** it**++)**

**{**

Point **\***p **=** **\***it**;**

**\***p **=** **\***p **+** diff**;**

**}**

**}**

void Torch**::**MoveBy**(**float dx**,** float dy**)**

**{**

CyclicList**<**Point **\*>::**Iterator it **=** points**.**begin**();**

**for** **(;** it **!=** points**.**end**();** it**++)**

**{**

Point **\***p **=** **\***it**;**

**\***p **=** **(\***p**).**MoveBy**(**dx**,** dy**);**

**}**

**}**

bool **operator** **==(**const Torch **&**t1**,** const Torch **&**t2**)**

**{**

**return** t1**.**GetPosition**()** **==** t2**.**GetPosition**()**

**&&** t1**.**GetBurner**().**GetLength**()** **==** t2**.**GetBurner**().**GetLength**();**

**}**

bool **operator** **!=(**const Torch **&**t1**,** const Torch **&**t2**)**

**{**

**return** **!(**t1 **==** t2**);**

**}**

ostream **&** Torch**::**PrintPointsTo**(**ostream **&**os**)** const

**{**

CyclicList**<**Point **\*>::**Iterator it **=** points**.**begin**();**

**for** **(;** it **!=** points**.**end**();** it**++)**

**{**

os **<<** **\*(\***it**)** **<<** "\n"**;**

**}**

**return** os**;**

**}**

ostream **&** Torch**::**PrintDetailedTo**(**ostream **&**os**)** const

**{**

**return** os **<<** "Torch[\n "

**<<** "Point" **<<** GetPosition**()** **<<** ",\n "

**<<** GetBurner**()** **<<** ",\n "

**<<** GetBowl**()** **<<** ",\n "

**<<** GetBody**()** **<<** ",\n "

**<<** GetHandle**()** **<<** "\n"

**<<** "]"**;**

**}**

ostream **&** **operator** **<<(**ostream **&**os**,** const Torch **&**t**)**

**{**

**return** os **<<** "Torch[Point" **<<** t**.**GetPosition**()** **<<** "]"**;**

**}**