

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Кафедра: 806 «Вычислительная математика и программирование»
Факультет: «Прикладная математика и физика»
Дисциплина: «Объектно-ориентированное программирование»

Лабораторная работа №1.
Тема: «Классы»

Группа: 8О-408Б
Студент: Забарин Никита Игоревич
Преподаватель: Поповкин Александр Викторович
Вариант: №26

Москва
2017

Цель работы

Целью лабораторной работы является:

- Программирование классов на языке C++
- Управление памятью в языке C++
- Изучение базовых понятий ООП.
- Знакомство с классами в C++.
- Знакомство с перегрузкой операторов.
- Знакомство с дружественными функциями.
- Знакомство с операциями ввода-вывода из стандартных библиотек.

Задание

- Необходимо спроектировать и запрограммировать на языке C++ классы фигур, согласно варианту задания(квадрат, прямоугольник и трапеция).
- Классы должны удовлетворять следующим правилам:
- Должны иметь общий родительский класс Figure.
- Должны иметь общий виртуальный метод Print, печатающий параметры фигуры и ее тип в стандартный поток вывода cout.
- Должны иметь общий виртуальный метод расчета площади фигуры – Square.
- Должны иметь конструктор, считывающий значения основных параметров фигуры из стандартного потока cin.
- Должны быть расположены в отдельных файлах: отдельно заголовки (.h), отдельно описание методов (.cpp).
- Программа должна позволять вводить фигуру каждого типа с клавиатуры, выводить параметры фигур на экран и их площадь.

Выводы

Листинг

```
trapezoid.h:
#ifndef TRAPEZOID_H
#define TRAPEZOID_H

#include <iostream>
#include "figure.h"

class Trapezoid : public Figure
{
public:
    Trapezoid();
    Trapezoid(std::istream& is);

    void print() const override;
    double area() const override;

private:
    double m_sideA;
    double m_sideB;
    double m_height;
};

#endif
```

```

square.h:
#ifndef SQUARE_H
#define SQUARE_H

#include <iostream>
#include "figure.h"

class Square : public Figure
{
public:
    Square();
    Square(std::istream& is);

    void print() const override;
    double area() const override;

private:
    double m_side;
};

#endif

trapezoid.cpp:
#include "trapezoid.h"

Trapezoid::Trapezoid()
{
    m_sideA = 0.0;
    m_sideB = 0.0;
    m_height = 0.0;
}

Trapezoid::Trapezoid(std::istream& is)
{
    std::cout << "=====" << std::endl;
    std::cout << "Enter side A: ";
    is >> m_sideA;
    std::cout << "Enter side B: ";
    is >> m_sideB;
    std::cout << "Enter height: ";
    is >> m_height;
}

void Trapezoid::print() const
{
    std::cout << "=====" << std::endl;
    std::cout << "Figure type: trapezoid" << std::endl;
    std::cout << "Side A size: " << m_sideA << std::endl;
    std::cout << "Side B size: " << m_sideB << std::endl;
    std::cout << "Height: " << m_height << std::endl;
}

double Trapezoid::area() const
{
    return m_height * (m_sideA + m_sideB) / 2.0;
}

rectangle.h:
#ifndef RECTANGLE_H
#define RECTANGLE_H

#include <iostream>
#include "figure.h"

```

```

class Rectangle : public Figure
{
public:
    Rectangle();
    Rectangle(std::istream& is);

    void print() const override;
    double area() const override;

private:
    double m_sideA;
    double m_sideB;
};

#endif

```

```

figure.h:
#ifndef FIGURE_H
#define FIGURE_H

```

```

class Figure
{
public:
    virtual ~Figure() {}
    virtual void print() const = 0;
    virtual double area() const = 0;
};

#endif

```

```

square.cpp:
#include "square.h"

```

```

Square::Square()
{
    m_side = 0.0;
}

Square::Square(std::istream& is)
{
    std::cout << "=====" << std::endl;
    std::cout << "Enter side: ";
    is >> m_side;
}

```

```

void Square::print() const
{
    std::cout << "=====" << std::endl;
    std::cout << "Figure type: square" << std::endl;
    std::cout << "Side size: " << m_side << std::endl;
}

```

```

double Square::area() const
{
    return m_side * m_side;
}

```

```

makefile:
CC = g++
CFLAGS = -std=c++11 -Wall -Werror -Wno-sign-compare -Wno-unused-result
FILES = *.cpp
PROG = lab1

```

```

all:
    $(CC) $(CFLAGS) -o $(PROG) $(FILES)

clean:
    rm $(PROG)

main.cpp:
#include "square.h"
#include "rectangle.h"
#include "trapezoid.h"

void testFigure(Figure* figure);

int main()
{
    testFigure(new Square(std::cin));
    testFigure(new Rectangle(std::cin));
    testFigure(new Trapezoid(std::cin));

    return 0;
}

void testFigure(Figure* figure)
{
    figure->print();

    std::cout << "Area: " << figure->area() << std::endl;

    delete figure;
}

rectangle.cpp:
#include "rectangle.h"

Rectangle::Rectangle()
{
    m_sideA = 0.0;
    m_sideB = 0.0;
}

Rectangle::Rectangle(std::istream& is)
{
    std::cout << "=====" << std::endl;
    std::cout << "Enter side A: ";
    is >> m_sideA;
    std::cout << "Enter side B: ";
    is >> m_sideB;
}

void Rectangle::print() const
{
    std::cout << "=====" << std::endl;
    std::cout << "Figure type: rectangle" << std::endl;
    std::cout << "Side A size: " << m_sideA << std::endl;
    std::cout << "Side B size: " << m_sideB << std::endl;
}

double Rectangle::area() const
{
    return m_sideA * m_sideB;
}

```