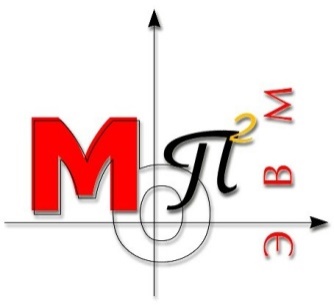
Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждения высшего образования   
«Южный федеральный университет»

Инженерно-технологическая академия

Институт компьютерных технологий и информационной безопасности

Кафедра математического обеспечения и применения ЭВМ

**

Лабораторная работа №2

по дисциплине

"ООП"

на тему

"Наследование в С++"

**Выполнил:**

студент группы КТбо2-7

Борисов Я. П

**Проверил:**

Преподаватель  
Тарасов С.А.

**ВАРИАНТ №5**

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

Создать класс Figure, объект которого задается координатами точек, c вычислением площади, центра тяжести и пери-метра. На его основе реализовать классы Rectangle и Pentagon.

**СПЕЦИФИКАЦИЯ КЛАССОВ**

**БАЗОВЫЙ КЛАСС**

class Figure {

public:

Points\* points;

double\* weight = new double[2];

Figure(int count);

virtual double getArea();

virtual double getPerimeter();

virtual void getWeight();

virtual void onRotate(const double angle);

virtual void onMove(const double x, double y);

bool asSidesExist(const int count);

~Figure();

protected:

double\* side;

private:

Points\* getPoints(const int count);

};

**ПРОИЗВОДНЫЙ КЛАСС RACTANGLE**

class Ractangle: public Figure {

public:

Ractangle(const int count);

double getArea() override;

double getPerimeter() override;

void onRotate(const double angle) override;

void onMove(const double x, const double y) override;

void getWeight() override;

~Ractangle();

private:

static const int \_R = 4;

};

**ПРОИЗВОДНЫЙ КЛАСС PENTAGON**

class Pentagon : public Figure {

public:

Pentagon(const int count);

double getArea() override;

double getPerimeter() override;

void getWeight() override;

void onRotate(const double angle) override;

void onMove(const double x, const double y) override;

~Pentagon();

private:

static const int \_P = 5;

};

**ВСПОМОГАТЕЛЬНЫЙ КЛАСС POINT**

class Points {

public:

Points(int n);

double\* x;

double\* y;

~Points();

};

**КЛАСС ФАБРИКИ**

class ShapeFactory {

public:

static Figure\* createFigure(const int value);

private:

static const int \_MINIMUM\_FIGURE\_SIDE = 3;

};

**КЛАСС ИТЕРФЕЙСА**

class Console {

public:

Console();

void onConsoleCall();

~Console();

private:

static const int \_COMMANDS\_LENGS = 9;

Figure\* figure;

int length;

std::string strings[\_COMMANDS\_LENGS] = {"Create shape", "What area?", "What weight?", "What perimeter?",

"Move Figure" , "Rotate Figure" , "Show coordinates", "Delete shape", "Exit"};

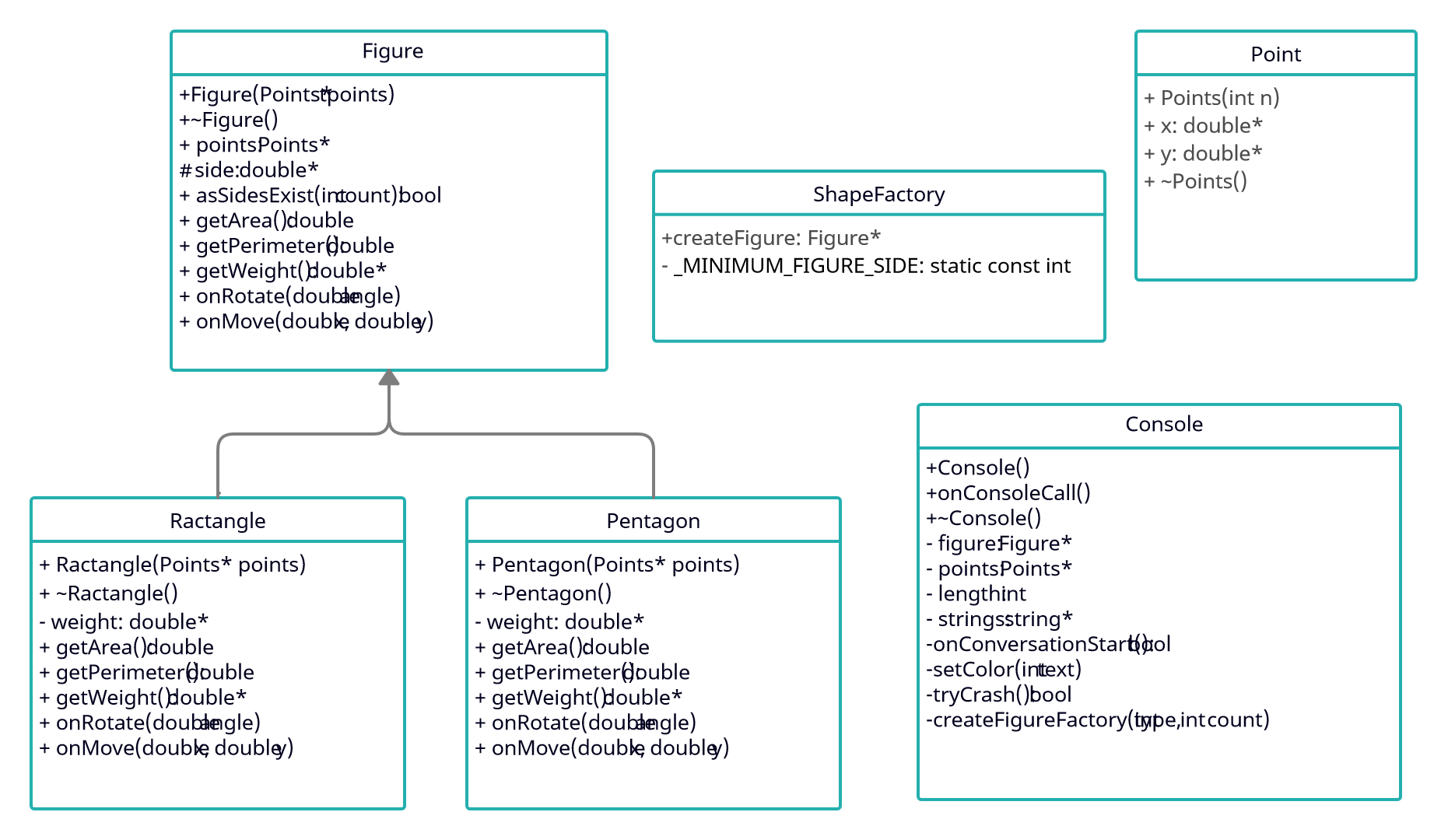
bool onConversationStart();

static void setColor(int text);

bool tryCrash();

void createFigureFactory();

};

**Диаграмма классов**