# Министерство науки и высшего образования Российской Федерации

## Федеральное государственное бюджетное образовательное учреждение высшего образования

## «Новосибирский государственный технический университет»

## Кафедра теоретической и прикладной информатики

### Расчётно-Графическое задание по дисциплине «Объектно-ориентированное программирование»

**Создание диаграммы классов и прототипа кода на языке c++ в системе «Enterprise Architect»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Факультет: | ПМИ |  |  |
| Группа: | ПМИ-81 |  |  |
| Студенты: | Демидович Е.Ю.,  Муравьев М.И. |  |  |
| Вариант: | 7;8а |  |  |
| Преподаватель: | Еланцева И.Л., Лисицин Д.В. |  |  |

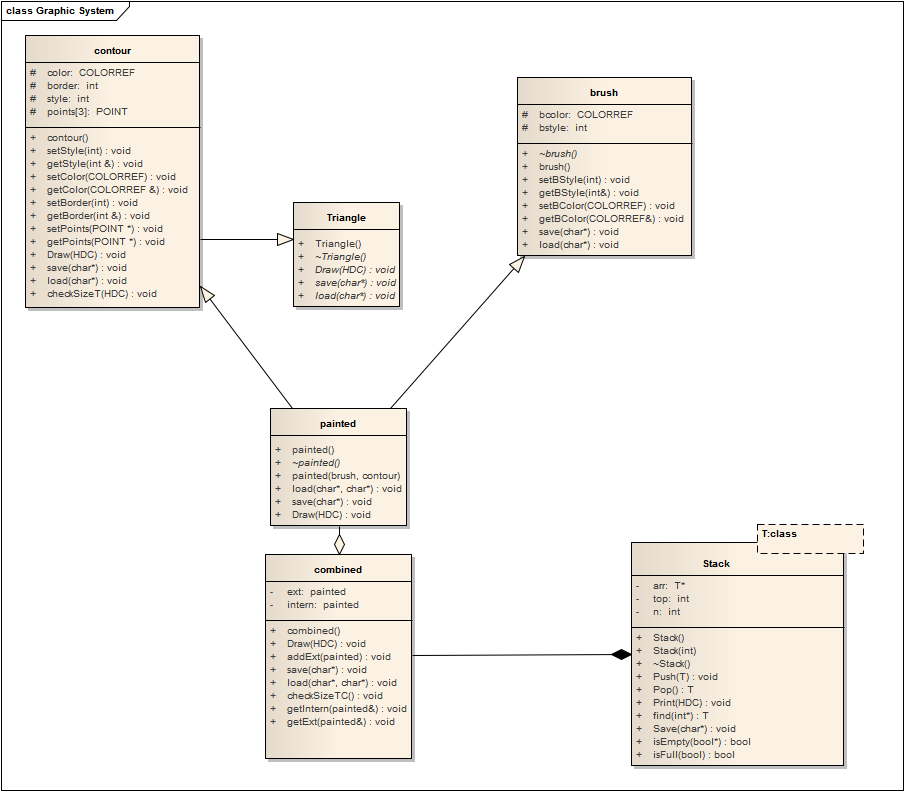
Новосибирск

2019

1. **Постановка задачи**

Составить диаграмму классов графической системы, разработанной в лабораторных работах № 4, 5, и произвести автоматическую генерацию прототипа кода разработанных классов.

1. **Разработанная диаграмма класса**



1. **Текст прототипа кода**

**Triangle.h**

class Triangle

{

public:

Triangle();

virtual ~Triangle();

virtual void Draw(HDC hdc) =0;

virtual void save(const char\* name) =0;

virtual void load(const char\* name) =0;

};

**Stack.h**

template<class T>

class Stack

{

public:

Stack(){

}

Stack(int N){

}

~Stack(){

}

void Push(T d){

}

T Pop(){

return NULL;

}

void Print(HDC hdc){

}

T find(int\* key){

return NULL;

}

void Save(const char\* name){

}

bool isEmpty(bool false){

return false;

}

bool isFull(bool false){

return false;

}

private:

T\* arr;

int top;

int n;

};

**brush.h**

class brush

{

public:

brush();

virtual ~brush();

void setBStyle(int new\_bstyle);

void getBStyle(int& curr\_bstyle);

void setBColor(COLORREF new\_bcolor);

void getBColor(COLORREF& curr\_bcolor);

void save(const char\* name);

void load(const char\* name);

protected:

COLORREF bcolor;

int bstyle;

};

**contour.h**

class contour : public Triangle

{

public:

virtual ~contour();

contour();

void setStyle(int new\_style);

void getStyle(int & curr\_style);

void setColor(COLORREF new\_color);

void getColor(COLORREF & curr\_color);

void setBorder(int new\_border);

void getBorder(int & curr\_border);

void setPoints(POINT \* new\_points);

void getPoints(POINT \* curr\_points);

void Draw(HDC hdc);

void save(const char\* name);

void load(const char\* name);

void checkSizeT(HDC hdc);

protected:

COLORREF color;

int border;

int style;

POINT points[3];

};

**painted.h**

class painted : public brush, public contour

{

public:

painted();

painted(brush Brush, contour Contour);

virtual ~painted();

void load(const char\* name1, const char\* name2);

void save(const char\* name);

void Draw(HDC hdc);

};

**combined.h**

class combined

{

public:

virtual ~combined();

combined();

void Draw(HDC hdc);

void addExt(painted Ext);

void save(const char\* name);

void load(const char\* name1, const char\* name2);

void checkSizeTC();

void getIntern(painted& t);

void getExt(painted& t);

private:

painted ext;

painted intern;

};