Учебный курс: Технологии программирования

Кафедра: ИС Факультет: ИТиП

Лабораторная работа №5-6. Наследование.

Выполнил: Трофимов В.А. Группа: 1511

Преподаватель: Повышев В.В.

# Задание

|  |  |  |
| --- | --- | --- |
| 8 | Edit – графическая текстовая форма ввода. Функционал:  Получить\изменить координаты  Получить на вход текст  Сохранять\не сохранять историю ввода  Включить\выключить автозаполнение  Произвести автозаполнение на основе истории и части введенного текста | NumericEdit – форма ввода, ограниченная числовыми значениями на вход. Функционал:  Получить\установить границы вводимого числа  Получить\установить реакцию на ввод числа вне диапазона (округлить \игнорировать\выдать ошибку)  Получить\установить запоминаемое кол-во знаков после запятой  Получить численное\строковое представление введенного числа |

# Входные/Выходные данные

Отсутствуют

# Исходный текст

## Файл main.cpp

#include <string>

using namespace std;

enum Reaction {

Round,

Ignore,

ShowError

};

class Range {

private:

long long left;

long long right;

public:

Range() {

this -> left = 0;

this -> right = 0;

}

void setLeft(const long long value) {

left = value;

}

void setRight(const long long value) {

right = value;

}

long long getLeft() {

return left;

}

long long getRight() {

return right;

}

};

class TextEdit {

private:

int x;

int y;

bool historySaving;

bool autoFill;

protected:

string content;

public:

TextEdit() {

x = y = 0;

content = "";

historySaving = autoFill = false;

}

~TextEdit() {

content.clear();

}

void setX(const int x) {

this -> x = x;

}

void setY(const int y) {

this -> y = y;

}

int getX() {

return x;

}

int getY() {

return y;

}

void addString(const string input) {

content.append(input);

}

void saveHistory(const bool value) {

historySaving = value;

}

void setAutoFill(const bool value) {

autoFill = value;

}

void AutoFill() {

//Тут должно быть автозаполнение

}

};

class NumericEdit : public TextEdit {

private:

Range range;

Reaction reaction;

unsigned char count;

public:

NumericEdit() {

reaction = Ignore;

count = 5;

Range r;

range = Range();

}

void setRange(const Range r) {

this -> range = r;

}

void setRange(const long long left, const long long right) {

this -> range.setLeft(left);

this -> range.setRight(right);

}

Range getRange() {

return range;

}

void setReaction(const Reaction r) {

this -> reaction = r;

}

Reaction getReaction() {

return reaction;

}

void setCount(const unsigned char c) {

this -> count = c;

}

unsigned char getCount() {

return count;

}

long long getIntRepresentation() {

return atol(content.c\_str());

}

string getStringRepresentation() {

return content;

}

};

int main() {

NumericEdit e = NumericEdit();

return 0;

}