МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГТУ», ВГТУ)

Факультет информационных технологий и компьютерной безопасности

Кафедра систем управления и информационных технологий в строительстве

КУРСОВОЙ ПРОЕКТ

Номер варианта 2,285714286

по дисциплине Основы программирования и алгоритмизации

Тема: Разработка калькулятора для расчета количества клея для пола в здании с заданной формой и размером.

**Расчётно-пояснительная записка**

Разработал студент Е. О. Солдатенко

Подпись, дата Инициалы, фамилия

Руководитель О. В. Минакова

Подпись, дата Инициалы, фамилия

Нормоконтролер О. В. Минакова

Подпись, дата Инициалы, фамилия

Защищена \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата

ВОРОНЕЖ

2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГТУ», ВГТУ)

Кафедра систем управления и информационных технологий в строительстве

ЗАДАНИЕ

на курсовой проект

по дисциплине Основы программирования и алгоритмизации

Тема: Разработка калькулятора для расчета количества клея для пола в здании с заданной формой и размером.

Студент группы бИСТ – 235 Солдатенко Екатерина Олеговна

Фамилия, имя, отчество

Технические условия Windows 11, MicrosoftVisualStudio 2022, язык программирования C

Содержание и объём проекта (графические работы, расчёты и прочее):

стр., рисунков., таб., приложение

Сроки выполнения этапов анализ и постановка задачи (10.09.2023 – 5.10.2023); разработка пошаговой детализации программы (06.10.2023 – 11.11. 2023); реализация программы (11.11.2023 – 05.12.2023); тестирование программы (06.12.2023 – 11.12.2023); оформление пояснительной записки (11.12.2023 – 14.12.2023)

Срок защиты курсового проекта

Руководитель О. В. Минакова

Подпись, дата Инициалы, фамилия

Задание принял студент Е. О. Солдатенко

Подпись, дата Инициалы, фамилия

Замечания руководителя

**Содержание**

[ВВЕДЕНИЕ 4](#_Toc156043173)

[1. Постановка задачи 5](#_Toc156043174)

[2. Алгоритм программы 8](#_Toc156043175)

[3. Тестирование программы 13](#_Toc156043176)

[ЗАКЛЮЧЕНИЕ 17](#_Toc156043177)

[ИСТОЧНИКИ 18](#_Toc156043178)

[ПРИЛОЖЕНИЕ 19](#_Toc156043179)

**ВВЕДЕНИЕ**

В современном мире всё больше стран страдают от перенаселения. С каждым годом людей становится всё больше и возникает вопрос: где все они будут жить? Именно поэтому в современном мире строительство крепких и надёжных зданий стало одной из важнейших задач.  Важной составляющей этого процесса является расчет требуемого количества клея для пола. Калькулятор расчета необходимого количества клея может стать незаменимым помощником для специалистов, занимающихся проектированием, строительством и ремонтом зданий.

Целью данной курсовой работы является разработка программы, которая будет выполнять расчет необходимого количества клея на основе данных размера пола и его формы.

Для достижения этой цели необходимо разработать алгоритм расчета требуемого количества клея.

Задачи курсового проекта:

* Разработать интерфейс калькулятора;
* Составить алгоритм расчета необходимого количества клея;
* Реализовать ввод данных и проверку их корректности;
* Реализовать запись введенной информации и полученных результатов в файл;
* Реализовать чтение информации из файла;
* Протестировать калькулятор.

Результаты данной курсовой работы могут быть использованы как специалистами в области строительства и ремонта зданий, так и обычными пользователями, желающими сделать ремонт или небольшой дом самостоятельно. Для людей, хотящих всё сделать самостоятельно наверняка появится необходимость рассчитать требуемое количество клея для пола.

1. **Постановка задачи**

Цель курсового проекта: создать программу-калькулятор для расчета количества клея для пола, размер и форма которого заданы.

В программе должны присутствовать следующие операции:

* Ввод пользователем данных:
* Форма пола;
* Длина и ширина пола в метрах, основываясь на предыдущем пункте;
* Вывод данных:
* Площадь пола;
* Виды клея и их необходимое количество в килограммах для заливки пола;
* Запись данных в файл;
* Чтение данных из файла;
* Проверка на корректность и выход из программы.

После ввода данных программа предложит продолжить ввод либо выйти в главное меню.

Для расчёта количества килограмм клея, необходимых для заливки будет использоваться следующая формула:

(1)

где К - количество килограмм клея, S - площадь пола с учётом его формы, h - высота заливки, вводимая пользователем, e - расход конкретного типа клея в кг/.

Так как у пола могут быть разные формы, то для подсчёта его площади будет использоваться одна из приведённых ниже формул.

Если форма пола - треугольник, то будет использоваться формула:

,  (2)

где S – площадь треугольного пола, а и b – длины катетов (какие это будут катеты, зависит от вида треугольника).

Если форма пола – прямоугольник, то будет использоваться формула:

(3)

В программе будут использованы: двумерный массив, поля ввода для сторон пола, запись данных.

При вводе данных должны верно происходить вычисления, показывая правильные результаты. При вводе иных значений, не предусмотренных программой будет показываться ошибка. Всё это должно быть показано при тестировании программы.

Таблица 1 - Ожидаемые результаты при тестировании программы

|  |  |  |
| --- | --- | --- |
| № | Вводные данные | Результат |
| 1 | Форма: Треугольник  Сторона 1: 4  Сторона 2: 6  Толщина слоя: 2  Расход: 3.6 | 87 |
| 2 | Форма: Прямоугольник  Сторона 1: 4  Сторона 2: 6  Толщина слоя: 2  Расход: 3.6 | 173 |
| 3 | Форма: Треугольник  Сторона 1: 4  Сторона 2: -6  Толщина слоя: 2  Расход: 3.6 | Ошибка |

1. **Алгоритм программы**

Программа начинается с функции «main». Выбор пунктов меню осуществляется с помощью оператора многоходового перехода (switch). Выйти из него можно с помощью выбора и нажатия специального пункта, позволяющего покинуть программу. Работа функции продемонстрирована на рисунке 1.

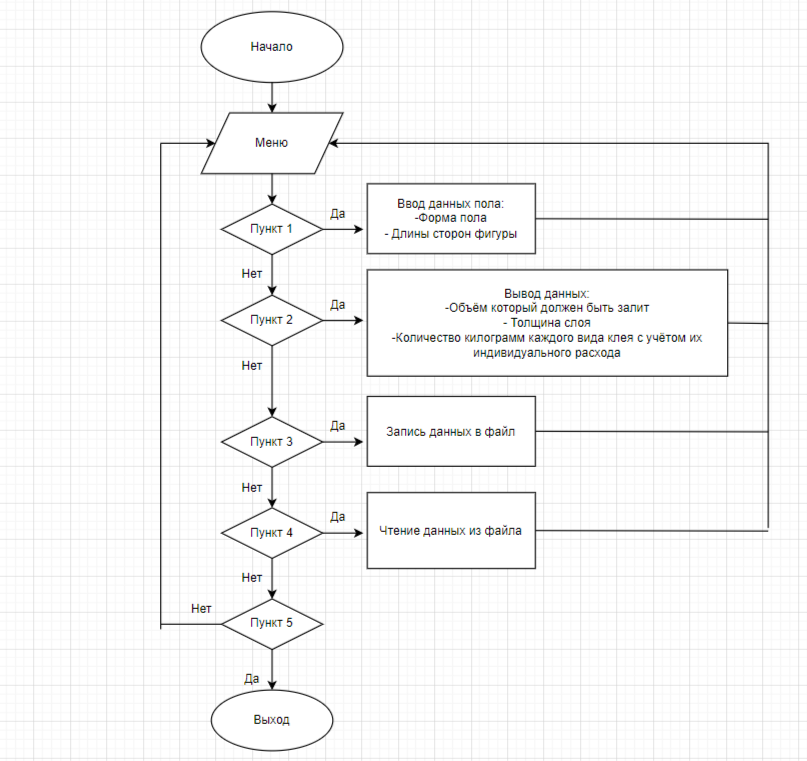


Рисунок 1 – Блок-схема функции main

В функцию int input( double ptr\_array[200][2], int aaa) записываются данные пола, которые продемонстрированы на рисунке 1; Функция принимает в качестве параметров двумерный массив ptr\_array и индекс записи aaa в массиве. Она приглашает к выбору фигуры, которую имеет пол, а также к вводу длин сторон фигуры в метрах и высоте толщины слоя в сантиметрах. Корректность введённых данных проверяется с помощью функции corr(). (Функция возвращает «++ааа»).

Для вывода информации используется функция  double output(double ptr\_array[200][2], int aaa). Она принимает в качестве параметров двумерный массив ptr\_array и индекс записи aaa в массиве. На экран выводится информация в виде таблицы, где записаны стороны фигуры и толщина заливки (информация, которая записывалась до этого). Далее последует приглашение к вводу номера выбранной фигуры. После ввода, в зависимости от номера фигуры, произойдёт подсчёт объёма, который необходимо залить. Информация запишется в переменную «colclay». Далее при выводе информации будет учитываться марка клея. Поэтому переменная, хранящая информацию будет умножена на «расход (кг/м2)». В итоге выведется информация: количество клея в кг для каждой марки. Функция возвращает переменную «colclay». Блок-схема данной функции продемонстрирована на рисунке 2.

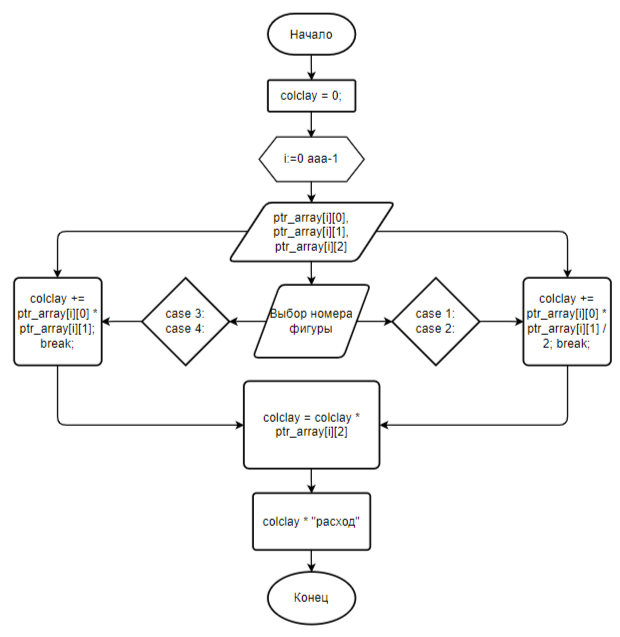


Рисунок 2 – Блок-схема вывода данных

int write(double ptr\_array[200][2], int index, char\* fname) – записывает вводимые данные в созданный ею же текстовый файл clay.txt. Она принимает в качестве параметров двумерный массив ptr\_array и индекс записи index в массиве. Если файл не открывается, то появится сообщение об ошибке. Возвращается 1.

int read(char\* fname) считывает файл с параметром fname и открывает его для чтения. При невозможности открыть файл появляется сообщение об ошибке и возвращается 0. При успешном открытии файла, на экран выводятся вводимые данные. Далее вводим номер выбранной фигуры и на экран выведется нужная информация. После всех действий внизу появится строчка приглашающая нажать Enter для возвращения в главное меню. Возвращается 1.

Функция double corr() – проверяет корректность вводимых значений. Сохраняет вводимые значения в переменную saving и проверяет их корректность. При введении неверных (или некорректных) значений функция выведет сообщение об ошибке и попросит ввести другие данные. Возвращается переменная «saving».

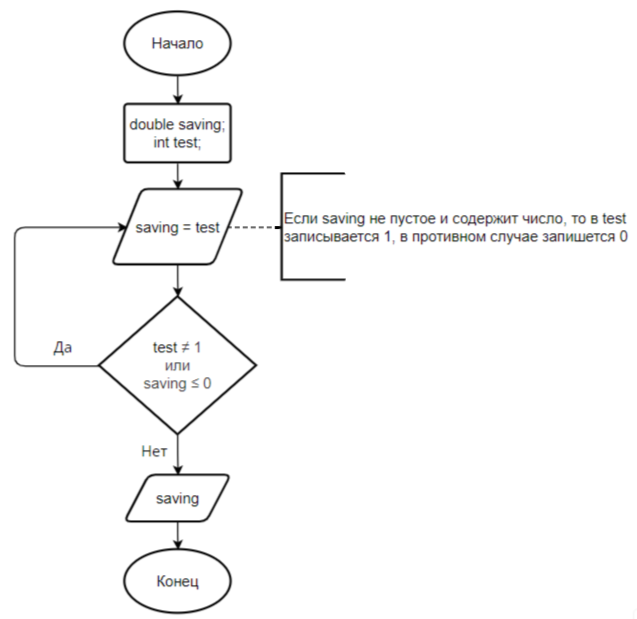


Рисунок 3 – Блок-схема функции корректности

1. **Тестирование программы**

При запуске программы появляется главное меню калькулятора, которое продемонстрировано на рисунке 4.

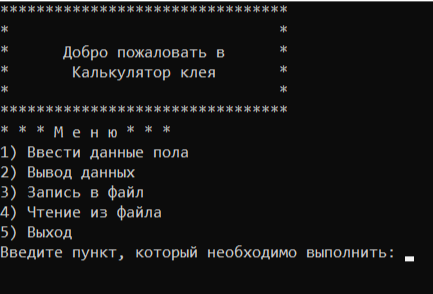


Рисунок 4 – Главное меню калькулятора

После выбора первого пункта появится список для выбора формы пола. После выбора программа предложит ввести соответствующие значения (если 1 пункт, то длину двух катетов, если 2 пункт, то длину высоты и основания треугольника и т.д.). После потребуется ввести толщину слоя. И в конце будет запрос на повтор ввода данных. Всё представлено на рисунке 5.

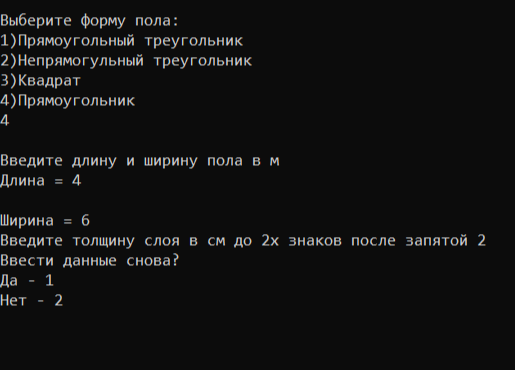


Рисунок 5 – Результат пункт главного меню

При вводе некорректных данных произойдёт то, что было описано ранее и теперь будет продемонстрировано на рисунке 6.

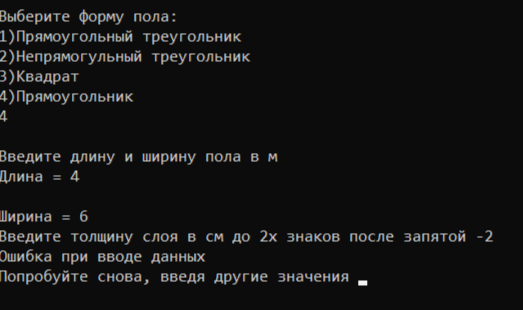


Рисунок 6 – Некорректные данные

Второй пункт – вывод данных. Всё, что выводится в этом пункте продемонстрированно на рисунке 7.

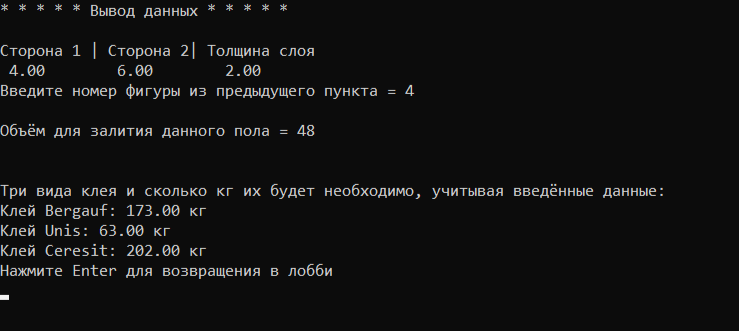


Рисунок 7 – Вывод данных

Третий пункт записывает данные в файл. Рисунок 8.

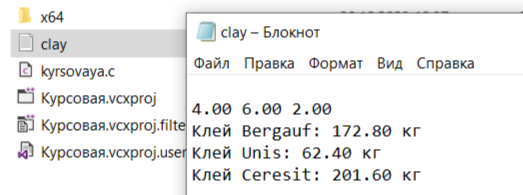


Рисунок 8 – Запись вводимых данных и результатов вычислений

О том, что данные успешно записаны свидетельствует надпись над всем калькулятором; рисунок 9.



Рисунок 9 – Оповещение о успешном сохранении данных

Четвёртый пункт – чтение данных из файла. Рисунок 10.

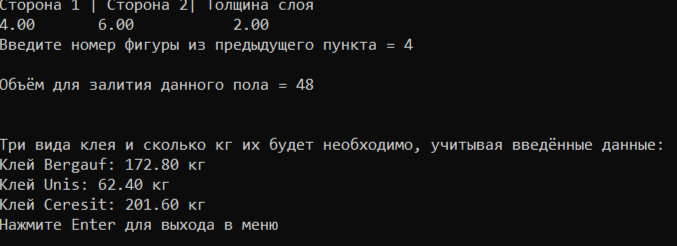


Рисунок 10 – Чтение данных из файла

Пятый пункт – выход из программы. После нажатия цифры «5», появится окно как на рисунке 11.

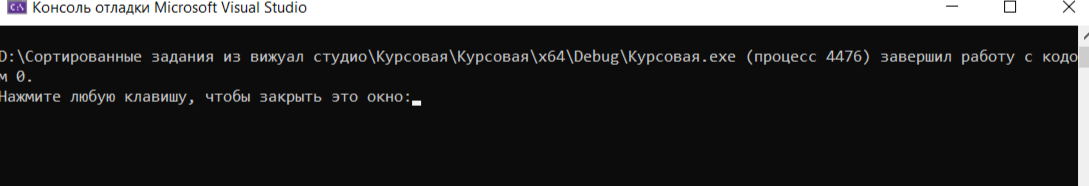


Рисунок 11 – Завершение работы программы

**ЗАКЛЮЧЕНИЕ**

Был разработан калькулятор для расчёта количества клея, необходимого для заливки пола под плитку. Ключевыми пунктами программы являются:

1. Ввод формы пола;
2. Ввод длин сторон и толщины слоя;
3. Вывод всей введённой информации вместе с подсчётами на основе введённых данных;
4. Запись в файл;
5. Считывание с файла;
6. Выход из программы.

В процессе работы были использованы различные алгоритмы, циклы, функции, массивы. Программа соответствует требованиям из постановки задачи, что стало ясно после тестирования.

Калькулятор решил все поставленные задачи и может быть в дальнейшем использован.

**ИСТОЧНИКИ**

1. Курипта О.В. Основы программирования и алгоритмизации: практикум / О.В.Курипта, О.В. Минакова, Д.К. Проскурин; Воронежский ГАСУ. – Воронеж, 2015.
2. Уинер Р. Язык Turbo Си. Пер. с англ. - М.: Мир, 1991.- 384 с. ( Библиотека НГТУ - 51У931 ).
3. В.В. Подбельский, С.С.Фомин.Программирование на языке Си. М.: ФиС, 1999, 600 с.
4. Романов E.Л.Язык Си. Типы данных и управление памятью.
5. Дуэйн Гриффитс, Доун Гриффитс.  «Изучаем программирование на C».
6. M. Уэйт, С. Прата, Д. Мартин - Язык Си руководство для начинающих.
7. Рацеев С.М. Программирование на языке Си: учебное пособие / Ульяновский государственный университет. – Ульяновск, 2015.
8. Егорова Е.В. Программирование на языке Си: учебное пособие / Алтайский государственный технический университет. – Барнаул, 2012.
9. Касюк С.Т. Курс программирования на языке Си: учебное пособие / Южно-Уральский государственный университет. – Челябинск, 2010.
10. Шишкин А.Д. Программирование на языке Си: учебное пособие / Российский государственный гидрометеорологический университет. – СПб, 2023.

**ПРИЛОЖЕНИЕ**

Код программы

#define \_CRT\_SECURE\_NO\_DEPRECATE

#include <stdio.h>

#include <locale.h>

#include <stdlib.h>

#define \_USE\_MATH\_DEFINES

#include <math.h>

#include <string.h>

int input(double ptr\_array[200][2], int); // ввод данных

double output(double ptr\_array[200][2], int); // вывод данных

int write(double ptr\_array[200][2], int, char\*); // запись в файл

int read(char\*); // чтение из файла

double corr(); // проверка корректности ввода

int main()

{

setlocale(LC\_CTYPE, "RUS");

int store = 0; // сохранность

int c; // выбор пункта

int flag;

int test; //проверка

char fname[200] = "clay.txt"; //хранение названия файла для записи и чтения

double clay[200][2];

int aaa = 0;

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\* \*\n");

printf("\* Добро пожаловать в \*\n");

printf("\* Калькулятор клея \*\n");

printf("\* \*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

while (1)

{

printf("\* \* \* М е н ю \* \* \* \n1) Ввести данные пола \n2) Вывод данных \n3) Запись в файл \n4) Чтение из файла \n5) Выход \nВведите пункт, который необходимо выполнить: ");

flag = scanf("%d", &c);

system("cls");

if (flag == 1) {

switch (c) {

case 1:

store = 1;

while (1) {

aaa = input(clay, aaa);

printf("Ввести данные снова? \nДа - 1\nНет - 2\n");

scanf("%d", &c);

system("cls");

if (c == 2) break;

}

break;

case 2:

if (aaa) {

double colclay;

colclay = output(clay, aaa);

printf("\nОбъём для залития данного пола = %2.f \n", colclay);

printf("\n");

printf("\nТри вида клея и сколько кг их будет необходимо, учитывая введённые данные:");

printf("\nКлей Bergauf: %.2f кг", ceil(colclay \* 3.6));

printf("\nКлей Unis: %.2f кг", ceil(colclay \* 1.3));

printf("\nКлей Ceresit: %.2f кг", ceil(colclay \* 4.2));

printf("\nНажмите Enter для возвращения в лобби\n");

getchar();

getchar();

system("cls");

}

else printf("Данные отсутствуют\n\n");

break;

case 3:

if (aaa) {

test = write(clay, aaa, fname);

if (test) {

printf("Информация успешно записана\n\n");

store = 0;

}

else {

printf("Ошибка\n\n");

}

}

else printf("Нет информации\n\n");

break;

case 4:

read(fname);

break;

case 5:

return 0; // выход из программы

default:

printf("Упс\n\n");

getchar();

break;

}

}

else {

system("cls");

printf("Введены некорректные данные\n\n");

getchar();

}

}

}

int input(double ptr\_array[200][2], int aaa) { //ВВОД ДАННЫХ

int ch;

printf("\nВыберите форму пола:\n1)Прямоугольный треугольник\n2)Непрямогульный треугольник\n3)Квадрат\n4)Прямоугольник\n");

scanf("%d", &ch);

switch (ch) {

case 1:

printf("\nВведите длины катетов в м:\n");

printf("\nКатет 1 = ");

ptr\_array[aaa][0] = corr();

printf("\nКатет 2 = ");

ptr\_array[aaa][1] = corr();

break;

case 2:

printf("\nВведите высоту и основание треугольного пола в м");

printf("\nВысота = ");

ptr\_array[aaa][0] = corr();

printf("\nОснование = ");

ptr\_array[aaa][1] = corr();

break;

case 3:

printf("\nВведите стороны квадрата в м");

printf("\nСторона 1 = ");

ptr\_array[aaa][0] = corr();

printf("\nСторона 2 = ");

ptr\_array[aaa][1] = corr();

break;

case 4:

printf("\nВведите длину и ширину пола в м");

printf("\nДлина = ");

ptr\_array[aaa][0] = corr();

printf("\nШирина = ");

ptr\_array[aaa][1] = corr();

break;

}

printf("Введите толщину слоя в см до 2х знаков после запятой ");

ptr\_array[aaa][2] = corr();

return ++aaa;

}

double output(double ptr\_array[200][2], int aaa) { //ВЫВОД ДАННЫХ

double colclay = 0;

int n;

printf("\* \* \* \* \* Вывод данных \* \* \* \* \*\n\nСторона 1 | Сторона 2| Толщина слоя\n");

for (int i = 0; i < aaa; i++)

{

printf("%5.2lf %.2lf %.2lf\n", ptr\_array[i][0], ptr\_array[i][1], ptr\_array[i][2]);

printf("Введите номер фигуры из предыдущего пункта = ");

scanf("%d", &n);

switch (n) {

case 1:

case 2:

colclay += ptr\_array[i][0] \* ptr\_array[i][1] / 2; break;

case 3:

case 4:

colclay += ptr\_array[i][0] \* ptr\_array[i][1]; break;

}

colclay = colclay \* ptr\_array[i][2];

}

return colclay;

}

int write(double ptr\_array[200][2], int index, char\* fname) { //ЗАПИСЬ В ФАЙЛ

FILE\* in = fopen(fname, "wt");

if (!in) {

printf("Ошибка");

return 0;

}

for (int i = 0; i < index; i++) {

fprintf(in, "\n%.2f %.2f %.2f", ptr\_array[i][0], ptr\_array[i][1], ptr\_array[i][2]);

fprintf(in, "\nКлей Bergauf: %.2f кг", ptr\_array[i][0]\* ptr\_array[i][1]\*ptr\_array[i][2] \* 3.6);

fprintf(in, "\nКлей Unis: %.2f кг", ptr\_array[i][0] \* ptr\_array[i][1] \* ptr\_array[i][2] \* 1.3);

fprintf(in, "\nКлей Ceresit: %.2f кг", ptr\_array[i][0] \* ptr\_array[i][1] \* ptr\_array[i][2] \* 4.2);

}

fclose(in);

return 1;

}

int read(char\* fname) { //ЧТЕНИЕ ИЗ ФАЙЛА

double colclay = 0;

double st1, st2, ex;

int n;

FILE\* out = fopen(fname, "rt");

if (out == NULL) {

printf("Ошибка\n\n");

return 0;

}

printf("\* \* \* \* \* Вывод данных \* \* \* \* \*\n\nСторона 1 | Сторона 2| Толщина слоя\n");

while (1) {

fscanf(out, "%lf", &st1);

fscanf(out, "%lf", &st2);

fscanf(out, "%lf", &ex);

printf("%.2lf %10.2lf %14.2lf\n", st1, st2, ex);

printf("Введите номер фигуры из предыдущего пункта = ");

scanf("%d", &n);

switch (n) {

case 1:

case 2:

colclay += st1 \* st2 \* ex / 2; break;

case 3:

case 4:

colclay += st1 \* st2 \* ex; break;

if (feof(out)) break;

}

fclose(out);

printf("\nОбъём для залития данного пола = %2.f \n", colclay);

printf("\n");

printf("\nТри вида клея и сколько кг их будет необходимо, учитывая введённые данные:");

printf("\nКлей Bergauf: %.2f кг", colclay \* 3.6);

printf("\nКлей Unis: %.2f кг", colclay \* 1.3);

printf("\nКлей Ceresit: %.2f кг", colclay \* 4.2);

printf("\nНажмите Enter для выхода в меню\n");

getchar();

getchar();

system("cls");

return 1;

}

}

double corr() { //КОРРЕКТНОСТЬ

double saving;

while (1) {

int test = scanf("%lf", &saving);

getchar();

if ((test != 1) || (saving <= 0)) printf("Ошибка при вводе данных \nПопробуйте снова, введя другие значения ");

else break;

}

return saving;

}