

ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
Факультет вычислительной техники и информатики  
Кафедра прикладной математики и информатики

На конкурс на лучшую научную работу студентов по разделу  
“Методика преподавания физики”

## **Студенческая научная работа**

На тему: **“Определение линейного коэффициента  
поглощения гамма-лучей в веществе”**

Выполнила ст. гр. ПО-01а Краснокутская М.В.

Руководители ст. пр. кафедры физики Товстухи В.С.

к. ф-м. н., доц. кафедры ПМИ Дацун Н.Н.

2002 год

## РЕФЕРАТ

Научно-исследовательская работа 29 стр., 1 рис., 3 прилож., 3 источника.

ВИРТУАЛЬНАЯ ФИЗИЧЕСКАЯ ЛАБОРАТОРИЯ, КОМПЬЮТЕРНАЯ ЛАБОРАТОРНАЯ РАБОТА, РАДИОАКТИВНОСТЬ, ГАММА-ЛУЧИ, КОЭФФИЦИЕНТ ПОГЛОЩЕНИЯ, КОМПЬЮТЕР, ПРОГРАММА

Объект исследования - твердое вещество.

Цель работы – создание модельной компьютерной лабораторной работы для определения линейного коэффициента поглощения гамма-лучей в веществе.

Метод исследования – компьютерное моделирование.

Предлагаемый материал является готовой модельной компьютерной лабораторной работой по ядерной физике для студентов втузов, что являлось целью проведенной работы. В ней программно повторяются классические опыты по определению линейного коэффициента поглощения гамма-лучей в веществе.

Выдаваемое на пересчетное устройство количество зарегистрированных гамма-квантов рассчитывается с учетом погрешности, задаваемой программно в соответствии с распределением Гаусса, что усиливает эффект реальности выполняемого эксперимента.

Программа составлена на языке Си, прошла пробные испытания на лабораторных занятиях и рекомендована методическим семинаром кафедры физики для внедрения в учебный процесс.

## СОДЕРЖАНИЕ

Введение	4
Модельная компьютерная лабораторная работа «Определение линейного коэффициента поглощения гамма-лучей в веществе».	5
Контрольные вопросы	11
Заключение	12
Список использованных источников	13
Приложение 1. Блок-схема алгоритма	14
Приложение 2. Листинг программы	17
Приложение 3. Изображение компьютерной лабораторной установки	28

## ВВЕДЕНИЕ

В лабораториях физического практикума сегодняшнего технического вуза не все экспериментальные идеи могут быть реализованы: одни – по экономическим соображениям, другие по техническим причинам, третьи – по экологическим, другие – из-за требований техники безопасности. В таких случаях эффективным является использование вычислительной техники, на которой можно создавать модельные образы реальных физических явлений. Это кроме всего прочего и экономически оправдано, т.к. на одной ПЭВМ можно воспроизвести неограниченное количество модельных работ.

Целью данной работы является создание модельной компьютерной лабораторной работы по разделу “Ядерная физика”. Прообразом явились известные опыты по определению линейного коэффициента поглощения гамма-лучей в веществе. В работе используется виртуальная лабораторная установка, которая состоит из камеры с источником излучения, счетчика Гейгера и пересчетного устройства. Для создания иллюзии реальности эксперимента количество зарегистрированных гамма-частиц задается с учетом погрешности, задаваемых программой в соответствии с гауссовским распределением.

После проведения компьютерного эксперимента программа может по желанию пользователя обработать результаты измерений, запросив их, и сообщить значение линейного коэффициента поглощения гамма-лучей в веществе с абсолютной и относительной погрешностями, рассчитанными по методу наименьших квадратов.

Текст лабораторной работы приведен в основной части.

Программа составлена на языке Си. Блок-схема алгоритма программы дана в Приложении 1.

Листинг программы приведен в приложении 2.

В приложении 3 дано изображение компьютерной лабораторной установки.

## МОДЕЛЬНАЯ КОМПЬЮТЕРНАЯ ЛАБОРАТОРНАЯ РАБОТА “ОПРЕДЕЛЕНИЕ КОЭФФИЦИЕНТА ПОГЛОЩЕНИЯ ГАММА-ЛУЧЕЙ”

В данной работе предпринята попытка создания программного продукта, лежащего в основе модельной компьютерной лабораторной работы “Определение линейного коэффициента поглощения гамма-лучей в веществе”, которая идеологически повторяет классические опыты приборного определения рассматриваемого коэффициента в физическом практикуме технического вуза. Выполнение лабораторной работы в модельном варианте в компьютерном классе или в режиме дистанционного образования позволяет избежать многих сложностей практического плана и в первую очередь экологических, которые неизбежно возникают при попытке приборной реализации упомянутых опытов с радиоактивными веществами.

Пользователь работает с программой в диалоговом режиме и работа с ней по своей сложности доступна студентам первого курса, даже не знакомым с вычислительной техникой. Программа позволяет в течение лабораторного занятия провести модельный эксперимент и, обработав результаты измерений, получить значение линейного коэффициента поглощения гамма-лучей для выбранного материала при выбранном значении энергии гамма-квантов.

Очень важным является то, что в работу программы введены элементы метода Монте-Карло и подчиняющаяся распределению Гаусса погрешность, делающие процесс измерений и его результаты правдоподобными.

При радиоактивном распаде ядер и при ядерных реакциях возможно образование ядер, находящихся в возбужденном состоянии, которые в дальнейшем, переходя в основное состояние, излучают гамма-кванты. Гамма-излучение представляет собой жесткое электромагнитное излучение с длиной волны менее 1 пм. Гамма-излучение обладает большой проникающей способностью и взаимодействует с веществом значительно слабее альфа- или

бета-излучения и проходит в воздухе сотни метров, а в твердых телах – сантиметры или даже дециметры.

Интенсивность гамма-излучения является величиной, численно равной количеству гамма-квантов, проходящих в единицу времени через поверхность единичной площади, расположенную перпендикулярно потоку.

При прохождении гамма-лучей через вещество основными процессами взаимодействия гамма-квантов с веществом являются фотоэффект, эффект Комптона и образование электрон-позитронных пар, приводящие к уменьшению интенсивности пучка. Вклад того или иного явления в суммарный эффект ослабления интенсивности гамма-лучей зависит как от вида вещества, так и от энергии гамма-квантов. Как правило, при малых энергиях преобладающим является фотоэффект, при больших – образование электрон-позитронных пар. Уменьшение интенсивности гамма-лучей при прохождении через слой вещества пропорционально интенсивности пучка  $I$  и толщине слоя  $dx$ :

$$dI = -\mu I dx \quad (1)$$

Коэффициент пропорциональности  $\mu$ , характеризующий поглощательную способность данного вещества, получил название линейного коэффициента поглощения гамма-лучей. Его величина зависит как от природы поглотителя, так и от энергии излучения. Анализ уравнения раскрывает физический смысл коэффициента: линейный коэффициент поглощения гамма-лучей в веществе показывает, какая часть потока гамма-квантов поглощается на единице длины пути в данном веществе. Разделив переменные и проинтегрировав, получим:

$$I = I_0 \exp(-\mu x) \quad (2)$$

Видно, что интенсивность гамма-лучей убывает с увеличением пути лучей в веществе по экспоненциальному закону, а натуральный логарифм интенсивности гамма-лучей после прохождения слоя вещества является линейной функцией толщины этого слоя. Линейный коэффициент поглощения

гамма-лучей в веществе обратен толщине слоя вещества, при прохождении которого интенсивность гамма-лучей уменьшается в  $e$  раз.

Для регистрации гамма-лучей применяется счетчик Гейгера, на выходе которого возникают импульсы тока, каждый из которых с определенной степенью вероятности соответствует прохождению гамма-кванта через рабочий объем счетчика. Количество импульсов, зарегистрированных счетчиком за время  $t$  пропорционально интенсивности потока гамма-излучения:

$$N = kIt, \quad (3)$$

а коэффициент пропорциональности  $k$  определяется параметрами счетчика (его типом, размерами, чувствительностью и т.п.) и энергией гамма-квантов. Это позволяет записать уравнение (2) в следующем виде:

$$\ln N = \ln N_0 - \mu x, \quad (4)$$

и перейти от интенсивности к практически более удобной величине числа импульсов  $N$ , зарегистрированных за произвольное время  $t$ .

При проведении измерений следует учитывать величину фонового гамма-излучения  $N_\phi$ , которое обусловлено существованием других источников природного и техногенного происхождения и космическим излучением.

С учетом фона уравнение примет вид: 
$$\ln \frac{N_0 - N_\phi}{N - N_\phi} = \mu x \quad (5)$$

Измерительная установка состоит из радиометра и источника гамма-излучения. Радиометр состоит из блока питания, счетчика Гейгера и устройства счета импульсов, возникающих при регистрации счетчиком гамма-квантов. Радиометр позволяет регистрировать гамма-излучение и выдает результат в виде количества зарегистрированных за произвольное время импульсов, пропорционального интенсивности потока гамма-лучей. Между счетчиком Гейгера и размещенном в стальном цилиндре источником гамма-излучения устанавливаются пластины исследуемого материала, для которого определяется линейный коэффициент поглощения гамма-излучения.

Все выше описанное было программно реализовано и представлено в диалоговом режиме для дисплейного использования.

После запуска программы появляется запрос о языке, на котором пользователю удобнее вести диалог с компьютером. Затем появляется главная заставка с элементом анимации, информирующая пользователя о названии продукта и его авторах. Нажатие любой клавиши приводит пользователя на следующую страницу с главным меню программы, предлагающим следующие варианты развития сценария:

**1. Порядок выполнения работы.** Здесь студент получает при необходимости инструкции, рекомендации и т.п.

**2. Образец таблицы.** Здесь размещена рекомендуемая таблица для ведения протокола измерений.

**3. Экспериментальная часть.** При выборе этого пункта появляется дополнительное меню со следующими предложениями:

**3.1. Контроль подготовленности к занятию.** Обойти этот пункт нельзя. При попытке обхода программа сообщает об обязательности тестирования. Тест содержит банк вопросов. Каждый вопрос содержит произвольное число вариантов ответов. Количество правильных ответов на вопрос произвольно. При тестировании студенту последовательно задается определенное количество вопросов, берущихся из банка в произвольном порядке. Варианты ответов также располагаются в произвольном порядке. Тестирование завершается выдачей оценки типа зачет-незачет с соответствующим продолжением.

**3.2. Выбор параметров эксперимента.** Предлагается определиться с энергией гамма-квантов, выбрав ее по имеющейся таблице в интервале от 0,1 до 5 МэВ. Предлагается выбрать исследуемый материал по имеющейся таблице со следующим перечнем: свинец, сталь, алюминий, бетон, медь. После сделанного выбора программа сообщает толщину пластинки из данного материала, с которой предстоит работать. Всего их будет 10 шт.



3.3. *Модельный эксперимент.* Появляется заставка с изображением экспериментальной установки. Показаны источник гамма-излучения в закрытой стальной капсуле, набор из 10 пластин исследуемого материала, счетчик Гейгера со счетным устройством. Показаны циферблаты секундомера и счетчика импульсов. В нижней части экрана располагается дополнительное меню, определяющее порядок проведения эксперимента.

3.3.1. Сброс. Эта опция возвращает систему в исходное состояние с обнулением счетчиков и переводом пластин в исходное состояние.

3.3.2. Фон. В этом режиме начинается счет импульсов счетчика Гейгера с закрытым источником излучения. Время измерения определяется пользователем.

3.3.3. Источник. Работа начинается с просьбы указать число пластин, которые будут использованы в данном эксперименте. После ввода ответа требуемое число пластин опускается в зазор между источником излучения и счетчиком Гейгера, открывается отверстие в источнике излучения и поток гамма-квантов направляется через исследуемый материал на счетчик. Автоматически начинается счет импульсов и времени. Время измерения и количество опытов определяется пользователем.

3.3.4. Выход в главное меню.

**4. Обработка результатов измерений.** Программа последовательно запрашивает тип исследуемого материала, толщину одной пластинки, количество используемых в данном измерении пластин и соответствующее количество зарегистрированных импульсов. Обработав полученные данные по методу наименьших квадратов, программа выдает значение линейного коэффициента поглощения гамма-лучей в виде доверительного интервала с указанием относительной погрешности и надежности.

Данная модельная компьютерная лабораторная работа по ядерной физике является законченным программным продуктом, написанном на языке Си, и,

работая в режиме ДОС, может быть использована и на компьютерах предыдущих поколений.

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое радиоактивность?
2. Что такое линейный коэффициент поглощения гамма-лучей?
3. Какие частицы испускаются при радиоактивных превращениях?
4. Что такое гамма-лучи?
5. Как изменяется интенсивность гамма-лучей при прохождении через вещество?
6. Какие процессы происходят при взаимодействии гамма-излучения с веществом?
7. От чего зависит линейный коэффициент поглощения гамма-лучей?
8. Для чего предназначен счетчик Гейгера?
9. Укажите формулу закона поглощения гамма-лучей в веществе.
10. Для чего в счетчик Гейгера добавляют аргон?

## ЗАКЛЮЧЕНИЕ

Созданная модельная компьютерная лабораторная работа: “Определение линейного коэффициента поглощения гамма-лучей в веществе”, была опробована на занятиях в физическом практикуме и продемонстрировала свою работоспособность.

Это позволяет расширить перечень лабораторных работ физпрактикума без привлечения материальных средств на приборное обеспечение.

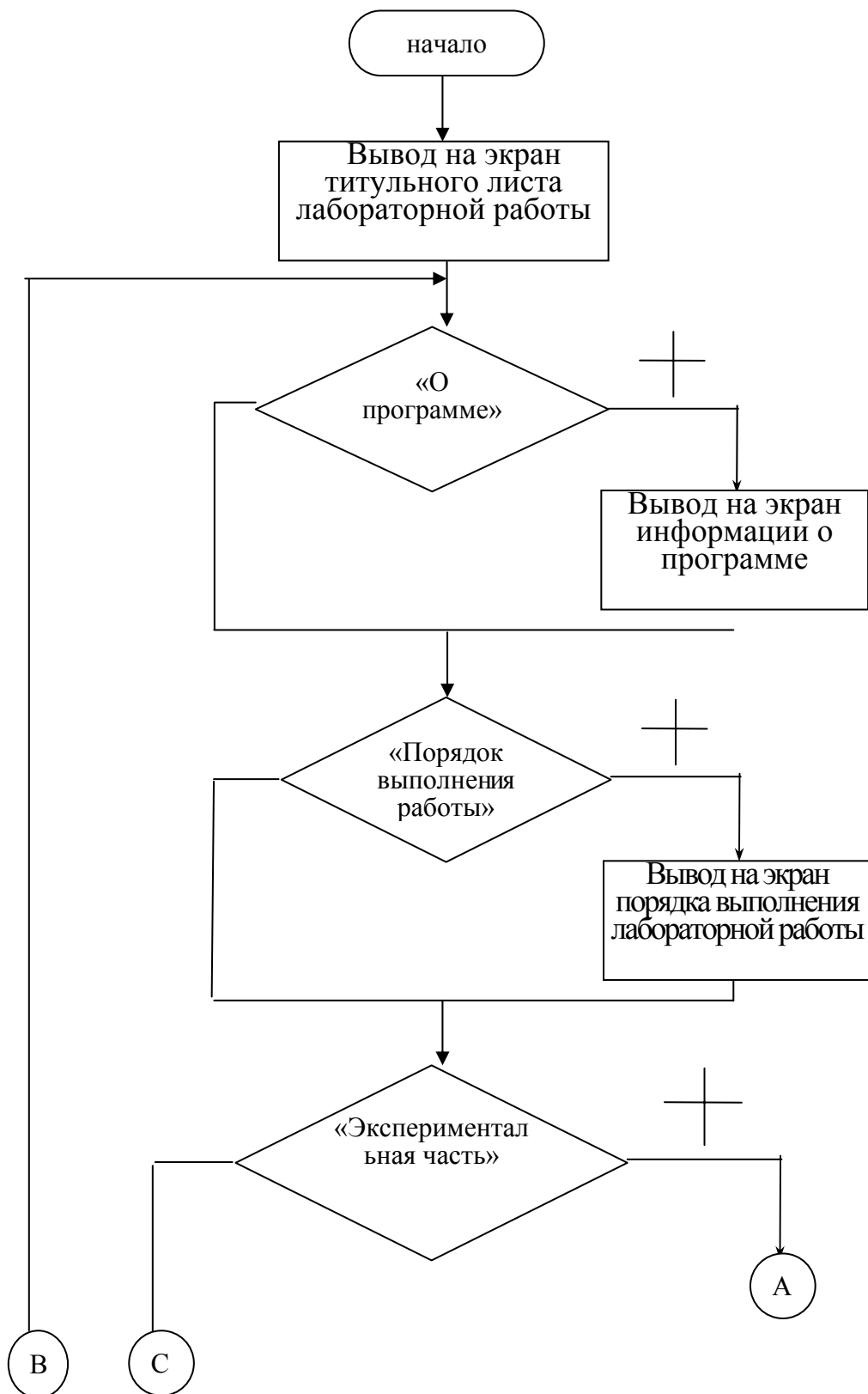
Созданный программный продукт можно использовать для описания всех аспектов распределения молекул по потенциальным энергиям.

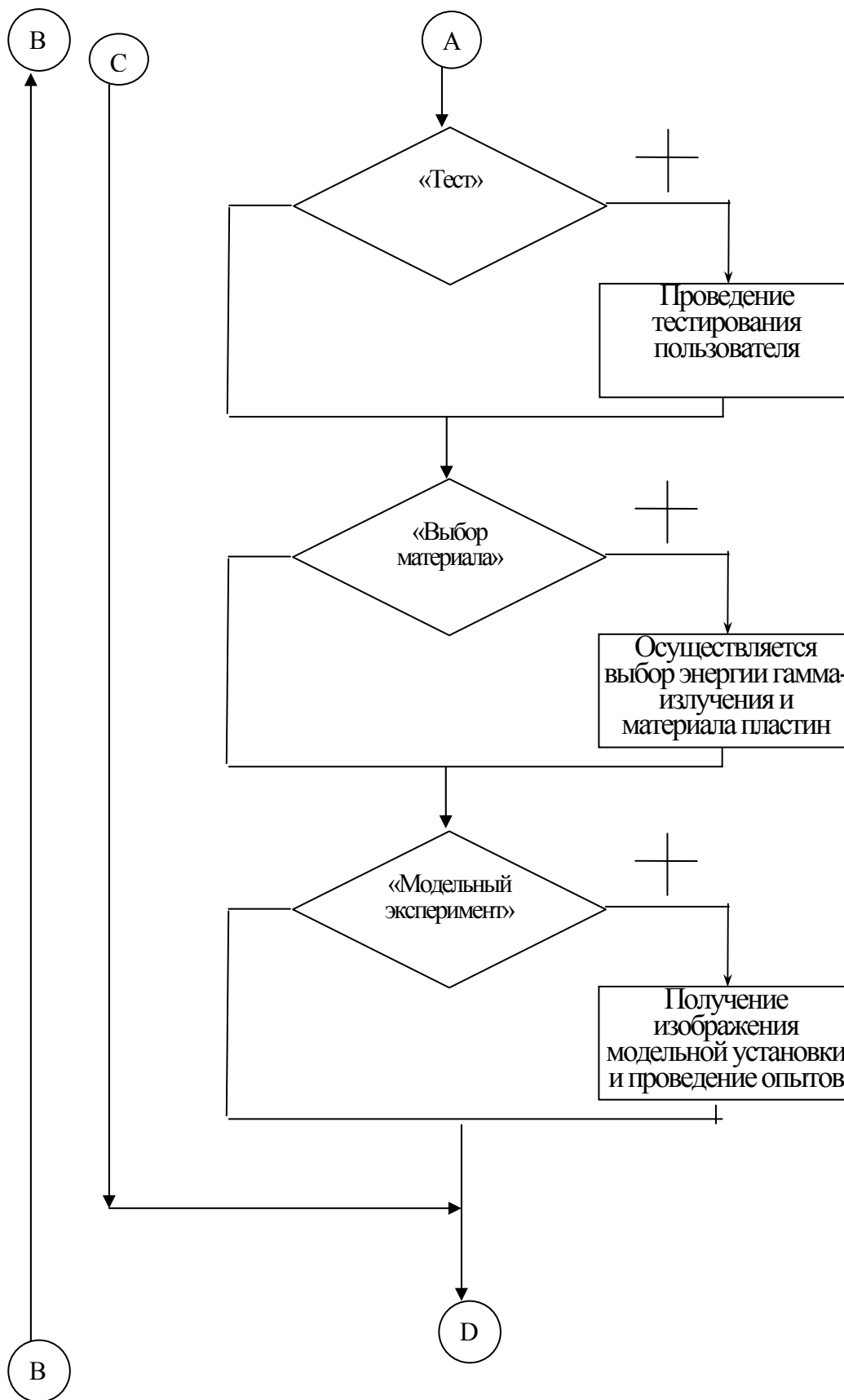
## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

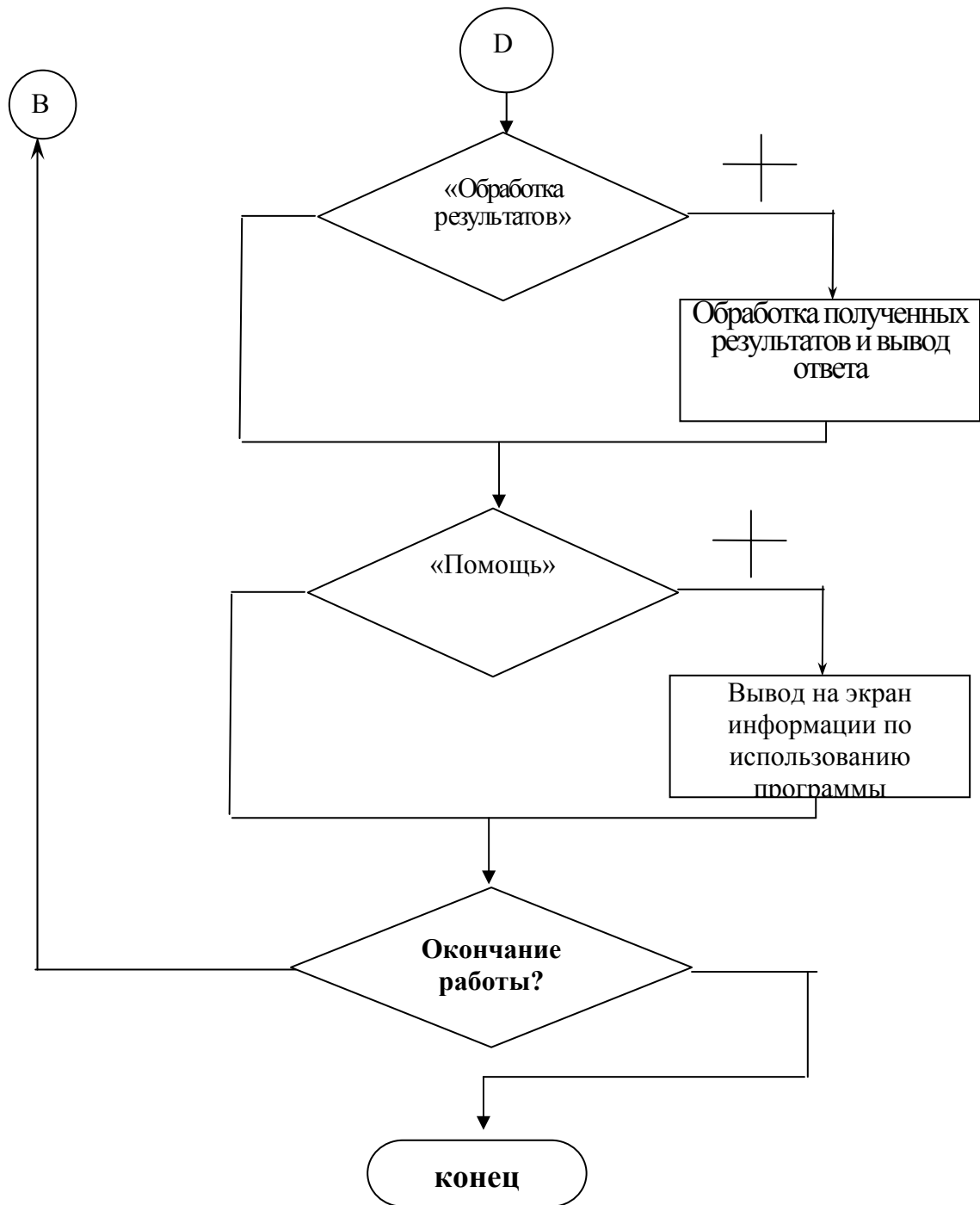
1. Савельев И. В. Курс общей физики. – М: Наука, 1982. Т.3.
2. Физический энциклопедический словарь / Под ред. Б. А. Введенского. – М.: Советская энциклопедия, 1960. Т.1. – 11, 94.
3. Керниган Б., Ритчи Д., Фьюэр А. Язык программирования Си. Задачи по языку Си – М.: Финансы и статистика, 1985.

## ПРИЛОЖЕНИЕ 1

## Блок-схема алгоритма









## ПРИЛОЖЕНИЕ 2

### Листинг программы

Приведен текст основной функции и текст функции модельного эксперимента.

```

/* программа иерархического меню
   для графического режима */

#include "dos.h"
#include "stdlib.h"
#include "string.h"
#include "stdio.h"
#include "graphics.h"
#include "conio.h"
#include "bios.h"
#include "ctype.h"
#include "math.h"
#include "alloc.h"

#define BORDER 1
#define ESC 27
#define MAX_FRAME 6
#define FONT_DEFAULT_FONT
#define CH_SIZE 1

void func_1();
void save_video();
void goto_xy(),write_video();
void display_menu(int);
void pd_driver(void),draw_border(int);
void write_string(int,int,int);
int is_in(char *, char);
int
make_menu(int,char**,char*,int,int,int,int);
int pulldown(int);
int get_resp(int);
void zastavka();
void scrolling(char*);
void test();
void soob(char *);
void set_language();
void quit(char*);
void quit1(char*);
extern char* okruglenie(char*);

int Flag1 = 0; /* Тест пройден? */
int Flag2 = 0; /* Материал выбран? */
int Flag3 = 0; /* Главное меню? */
char s[100];
FILE* in;
char far *vid_mem;
int DX; /* высота форточки меню */
void *ptr1;
int imas, jmas;
float b,a,da;
int language=0;
int nf_nach;
enum leng { RUSSIAN, ENGLISH };

float mas[9][5]=
{
{ 60, 2.82, 3.94, 0.444, 0.378 },
{ 11.8, 1.13, 1.40, 0.323, 0.275 },
{ 4.76, 0.85, 0.95, 0.278, 0.236 },
{ 2.51, 0.73, 0.824, 0.251, 0.214 },
{ 1.72, 0.66, 0.732, 0.228, 0.194 },
{ 0.79, 0.47, 0.522, 0.166, 0.141 },
{ 0.58, 0.38, 0.426, 0.137, 0.116 },
{ 0.51, 0.33, 0.371, 0.117, 0.100 },
{ 0.49, 0.25, 0.282, 0.075, 0.064 }
};

struct menu_frame
{
int startx,endx,starty,endy;
/* указатель на строки меню */
char **menu;
/* рамка включено/выключено */
int border;
/* горячие клавиши */
char *keys;
/* число альтернатив */
int count;
/* активно ли меню сейчас */
int active;
} frame[MAX_FRAME];

char *menu_3[2] =
{
" РУССКИЙ ",
" ENGLISH "
};

char *menu_1[2][6] =
{
{
"О программе",
"Порядок выполнения работы",
"Экспериментальная часть",
"Обработка результатов измерений",
"Помощь",
"Выход"
},
{
>About a program",
"The order of performance of work",
"Experimental part",
"Working of results of measuring",
"Help",
"Exit"
}
};

char *menu_2[2][3] =
{
{
"Тест",
"Выбор материала",
"Модельный эксперимент"
},
{
"Test",
"Choice of a material",
"Model experiment"
}
};

char *menu_4[2][9] =
{

```

```

{
    " 0,1 МэВ  ",
    " 0,2 МэВ  ",
    " 0,3 МэВ  ",
    " 0,4 МэВ  ",
    " 0,5 МэВ  ",
    " 1,0 МэВ  ",
    " 1,5 МэВ  ",
    " 2,0 МэВ  ",
    " 5,0 МэВ  "
},
{
    " 0,1 MeV  ",
    " 0,2 MeV  ",
    " 0,3 MeV  ",
    " 0,4 MeV  ",
    " 0,5 MeV  ",
    " 1,0 MeV  ",
    " 1,5 MeV  ",
    " 2,0 MeV  ",
    " 5,0 MeV  "
}
};
char *menu_5[2][5] =
{
    {
        " Свинец ",
        " Железо ",
        " Медь ",
        " Алюминий ",
        " Бетон "
    },
    {
        "Lead",
        "Iron",
        "Copper",
        "Aluminium",
        "Concrete"
    }
};
char *menu_6[2][4]=
{
    {
        " Фон      ",
        " Источник ",
        " Сброс    ",
        " Выход    "
    },
    {
        " Background ",
        " Sourse     ",
        " Fault      ",
        " Exit       "
    }
};
char *mes1 [] =
{
    "Enter - выбор опции меню, Esc - выход в
главное меню",
    "Enter - choose option of menu, Esc - escape
to main menu"
};
char *mes2 [] =
{
    "Выберите нужную опцию:",
    "Choose needed option:"
};
char *mes3 [] =
{
    "Enter - выбор опции меню, Esc - выход из
программы",
    "Enter - choose option of menu, Esc - exit"
};
char*m[2] =
{
    "Завершение работы? ",
    "Do you really want to quit? "
};
char* m_1[2] =
{
    "Вы должны выбрать материал!",
    "You must choose a material!"
};
char* m_2[2] =
{
    "Вы должны пройти тест!",
    "You must pass a test!"
};
/*-----*/
/*          ОСНОВНАЯ ПРОГРАММА          */
/*-----*/
main()
{
    int i,maxx,maxy;
    int graphdriver=ДЕТЕКТ; /* автоопределение */
    int gmode, errorcode;

    /* инициализация графического режима */
    initgraph(&graphdriver,&gmode,"");
    errorcode=graphresult();
    if(errorcode != grOk) /* если ошибка */
    {
        printf("Ошибка: %s\n",
grapherrormsg(errorcode));
        printf("Для останова нажмите любую
клавишу\n");
        getch();
        exit(1); /* завершение с кодом ошибки */
    }
    cleardevice();
    maxx=getmaxx();
    maxy=getmaxy();

    setbkcolor(BLUE);
    setfillstyle(SOLID_FILL,BLUE);
    bar(0,0,maxx,maxy-15);

    settextstyle(installuserfont("simp.chr"),HORIZ
_DIR,4);
    setcolor(LIGHTCYAN);
    setfillstyle(SOLID_FILL,LIGHTCYAN);
    bar(0,maxy-30,maxx,maxy);
    settextjustify(CENTER_TEXT,CENTER_TEXT);
    outtextxy(270,30,"Выберите язык:");
    outtextxy(270,60,"Choose the language:");
    settextstyle(0,0,1);
    setcolor(BLUE);
    outtextxy(270,maxy-7,"Выберите нужную опцию
(выделена голубым):");
    outtextxy(270,maxy-23,"Choose needed option
(blue choice):");
    make_menu(2,menu_3,"",2,130,190,BORDER);
    /* выбор языка */
    set_language();
    zastavka();
    soob(mes3[language]);
    /*"Enter - выбор опции меню, Esc - выход из
программы");*/

    /* Создадим фреймы меню */
    make_menu(0,menu_1,"zfuhvr",6,70,120,BORDER);
    make_menu(1,menu_2,"r;jp",3,110,180,BORDER);
    make_menu(3,menu_4,"",9,80,100,BORDER);
    make_menu(4,menu_5,"",5,80,300,BORDER);
    make_menu(5,menu_6,"",4,380,10,BORDER);

```

```

pd_driver();          /* запуск системы меню */
closegraph();
return(0);
}

/*-----*/
/*      программа запуска меню      */
/*-----*/

void pd_driver()
{
    void func_1(void);
    int choice1, choice2, selection;

/* активизация окон по мере надобности */
    nf_nach=5+(int)(16*random(1000)/1000.0);

    while((choice1=pulldown(0)) != -1)
    { Flag3 = 0;
      switch ( choice1 )
      {
        case 0: if(language)
                 scrolling("o_prog_e.txt");
                 else
                 scrolling("o_prog_r.txt");
                 soob(mes3[language]); /*
"Enter - выбор опции меню, Esc - выход из
программы" */
                 break;
        case 4: if(language)
                 scrolling("help_e.txt");
                 else
                 scrolling("help_r.txt");
                 soob(mes3[language]);
                 break;
        case 1: cleardevice();
                 if(language)
                 scrolling("text_k_e.txt");
                 else
                 scrolling("text_k_r.txt");
                 soob(mes3[language]); /*
"Enter - выбор опции меню, Esc - выход из
программы" */
                 break;
        case 3: if(Flag2)
                 {
                   cleardevice();

result(menu_5[language][jmas],menu_4[language]
[imas], &a, &da);
                 soob(mes3[language]); /*
"Enter - выбор опции меню, Esc - выход из
программы" */
                 }
                 else quit1(m_1[language]);
                 break;
        case 2: soob(mes1[language]);
                 Flag3 = 1;
                 while((choice2=pulldown(1)) != -1)
                 if ( choice2 ==1)
                 {
                   soob(mes1[language]);
                   func_1();
                   cleardevice();
                   soob(mes1[language]);
                 }
                 else if ( choice2 ==2)
                 {
                   if((Flag1)&&(Flag2))
                   {
                     cleardevice();
                     opyt();
                     soob(mes1[language]
                     }
                     else
                     if(!Flag2)
                     quit1(m_1[language]);
                     else quit1(m_2[language]);
                   }
                   else if(choice2==0)
                   {
                     test();
                     soob(mes1[language]);
                   }
                   soob(mes3[language]);
                   Flag3 = 1;
                   break;
                   case 5: quit(m[language]);
                   break;
                   /*exit(1); */
                   } /* switch */
                   Flag3=0;
                   } /* while */
                   } /* pd_driver */
                   /*|-----|*/
                   /*|      Функция выводит сообщение      |*/
                   /*|-----|*/
                   void soob(char * mess)
                   {
                     int maxx,maxy;
                     maxx=getmaxx();
                     maxy=getmaxy();
                     cleardevice();
                     setbkcolor(BLUE);
                     setcolor(LIGHTCYAN);

settextstyle(installuserfont("simp.chr"),HORIZ
_DIR,4);
settextjustify(CENTER_TEXT,CENTER_TEXT);
outtextxy(250,30,mes2[language]);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
setfillstyle(SOLID_FILL, LIGHTCYAN);
bar(0,maxy-15,maxx,maxy);
setcolor(BLUE);
outtextxy(275,maxy-7,mess);
                   }
                   /*-----*/
                   /*      Высветить меню и получить выбор      */
                   /*-----*/
                   int pulldown(int num)
                       /* номер фрейма */
                   {
                     int choice;
                     /* узнать активность окна */
                     if (!frame[num].active) /* не активно */
                     {
                       /* save_video(num); */ /*
запомнить часть окна */
                       frame[num].active=1; /* установка
признака активности */
                     }
                     if (frame[num].border) draw_border(num);
                     display_menu(num);
                     return get_resp(num); /*
возвратить выбор */
                   } /* конец pulldown */
                   /*-----*/
                   /*      Создание фрейма меню      */
                   /*-----*/
                       /* 1 если фрейм может быть создан
0 в противном случае */

```

```

/* #define DX 15      высота форточки меню */
#define DY 8        /* отступ от края рамки текста */
*/
int make_menu(num,menu,keys,count,x,y,border)
    int num;        /* номер меню */
    char **menu;    /* текст меню */
    char *keys;     /* горячие клавиши */
    int count;     /* число альтернатив */
    int x,y;       /* левый верхний угол */
    int border;    /* рамка */

{
    register int i,len;
    int endx,endy,choice;
    unsigned int *p;
    unsigned int size;
    int maxx, maxy;
    maxx=getmaxx();
    maxy=getmaxy();
    if (num>=MAX_FRAME)
    {
        printf(" слишком много меню ");
        return(0);
    }
    /* вычисление размеров строки меню */
    settextstyle(FONT,HORIZ_DIR,CH_SIZE);
    DX=textheight(menu[0])+8;
    /* вычисление высоты форточки */
    len=0;
    for (i=0; i<count; i++)
        if (textwidth(menu[i]) > len)
            len=textwidth(menu[i]);
    endx=x+count*DX;
    endy=y+len+2*DY;
    if ((endx+3 > maxx) || (endy+3 > maxy))
    {
        printf(" выход за пределы экрана ");
        return(0);
    }
    /* размещение памяти для видеобуфера */
    /* size=imagesize(y-4,x-4,endy+4,endx+4);
    p=( unsigned int*) malloc(size);
    if(!p) exit(1); */ /* нет памяти */
    /* создание фрейма */
    frame[num].startx=x;
    frame[num].endx=endx;
    frame[num].starty=y;
    frame[num].endy=endy;
    frame[num].menu=menu;
    frame[num].border=border;
    frame[num].keys=keys;
    frame[num].count=count;
    frame[num].active=0;
    return 1;
} /* конец make_menu */
/*-----*/
/*      Высветивание меню на своем месте      */
/*-----*/
void display_menu(num)
    int num;
{
    char **m;
    register int i,x,y;

    x=frame[num].startx+DX/2;
    switch(num)
    {
        case 0:  m=menu_1[0+language];
                 break;
        case 1:  m=menu_2[0+language];
                 break;
        case 2:  m=menu_3;
                 break;
        case 3:  m=menu_4[0+language];
                 break;
        case 4:  m=menu_5[0+language];
                 break;
        case 5:  m=menu_6[0+language];
                 break;
    };
    y=frame[num].starty+DY;
    setlinestyle(SOLID_LINE,0,THICK_WIDTH);
    setcolor(LIGHTBLUE);
    setfillstyle(SOLID_FILL,LIGHTGRAY);
    bar(frame[num].starty,frame[num].startx,
        frame[num].endy,frame[num].endx);
    setttextjustify(LEFT_TEXT,CENTER_TEXT);
    setttextstyle(FONT,HORIZ_DIR,CH_SIZE);
    setcolor(BLACK);
    for (i=0; i < frame[num].count; i++)
    {
        outtextxy(y,x+i*DX,m[i]);
    }
} /* конец display_menu */

/*-----*/
/*      Высветить рамку      */
/*-----*/
void draw_border(num)
    int num;
{
    setlinestyle(SOLID_LINE,0,THICK_WIDTH);
    setcolor(LIGHTBLUE);
    rectangle(frame[num].starty-
2,frame[num].startx-2,
frame[num].endy+2,frame[num].endx+2);
}
/*-----*/
/*      Выбор пользователя      */
/*-----*/
get_resp(num)
    int num;
{
    union inkey
    {
        char ch[2];
        int i;
    } c;
    int arrow_choice=0, key_choice;
    /* осветить первый выбор */
    write_string(arrow_choice,1,num);
    for(;;)
    {
        while(!bioskey(1)); /* ждать нажатия */
        c.i=bioskey(0);
        /* вернуть выбор в нормальный режим */
        write_string(arrow_choice,0,num);
        if(c.ch[0]) /* обычная клавиша */
        {
            key_choice=
is_in(frame[num].keys,tolower(c.ch[0]));
            if(key_choice) return (key_choice-
1);
            switch(c.ch[0])
            {
                case '\r' : return
arrow_choice;
                case ' ' : arrow_choice++;
                           break;
                case ESC  :
                           if(Flag3) return -
1;
                           else
                           break;
            } /* выйти */
        } /* switch */
    } /* if then */
    else /* специальная клавиша */

```

```

{
    switch(c.ch[1])
    {
        case 72 : arrow_choice--;
/* стрелка вниз */
            break;
        case 80 : arrow_choice++;
/* стрелка вверх */
            break;
    } /* switch */
} /* else */
if(arrow_choice==frame[num].count)
arrow_choice=0;
if(arrow_choice<0)
arrow_choice=frame[num].count-1;
/* подсветить выбранную функцию */
write_string(arrow_choice,1,num);
} /* for */
} /* конец процедуры get_resp */
/*-----*/
/* вывод строки в определенном цвете */
/*-----*/
/* attrib=1 - высвечивается
выбор;
attrib=0 - нормальное
свечение. */
void write_string(arrow_choice,attrib,num)
int arrow_choice, attrib, num;
{
int color_fon;
char** m;
switch(num)
{
case 0: m=menu_1[0+language];
break;
case 1: m=menu_2[0+language];
break;
case 2: m=menu_3;
break;

case 3: m=menu_4[0+language];
break;
case 4: m=menu_5[0+language];
break;
case 5: m=menu_6[0+language];
break;
};
setcolor(BLUE);
if(attrib) color_fon=LIGHTCYAN;
else color_fon=LIGHTGRAY;
setfillstyle(SOLID_FILL,color_fon);

bar(frame[num].starty,frame[num].startx+arrow_
choice*DX,

frame[num].endy,frame[num].startx+(arrow_choic
e+1)*DX);

outtextxy(frame[num].starty+DY,frame[num].star
tx+

arrow_choice*DX+DX/2,m[arrow_choice]);
}
/* конец программы write_string */
/*-----*/
/* is_in */
/*-----*/
int is_in(s,c)
char *s,c;
{
register int i;
for(i=0; *s; i++)
if(*s++ == c) return (i+1);
return 0;
}
/*-----*/
/* Функция выбора материала */
/*-----*/
void func_1()
{ int choice3, choice4, selection ,userfont ;
float m;
char str[15];
char* m1[2] =
{
"Выберите энергию излучения",
"Choose energy of radiation"
};
char *m2[2] =
{
"Выберите материал пластинки",
"Choose a material of a plate"
};
char* m3[2] =
{
"Толщина пластинки:",
"Thickness of a plate"
};
char* m4[2] =
{
"см", "sm"
};
userfont=installuserfont("simp.chr");
setfillstyle(SOLID_FILL,BLUE);
bar(frame[0].starty-5,frame[0].startx-
5,frame[0].endy+5,frame[0].endx+5);
setcolor(BLUE);
settextstyle(userfont,HORIZ_DIR,4);
settextjustify(CENTER_TEXT,CENTER_TEXT);
outtextxy(250,30,mes2[language]);
setcolor(LIGHTCYAN);
settextstyle(0,HORIZ_DIR,1);
settextjustify(CENTER_TEXT,CENTER_TEXT);
outtextxy(155,50,m1[language]);
while((choice3=pulldown(3)) != -1)
{
switch(choice3)
{
case 0:imas=0;
break;
case 1:imas=1;
break;
case 2:imas=2;
break;
case 3:imas=3;
break;
case 4:imas=4;
break;
case 5:imas=5;
break;
case 6:imas=6;
break;
case 7:imas=7;
break;
case 8:imas=8;
break;
}
setcolor(LIGHTCYAN);
settextstyle(0,HORIZ_DIR,1);
settextjustify(CENTER_TEXT,CENTER_TEXT);
outtextxy(385,50,m2[language]);
while((choice4=pulldown(4)) != -1)
{
switch(choice4)
{
case 0:jmas=0;
break;
case 1:jmas=1;
break;
case 2:jmas=2;
break;
}
}
}
}
}

```

```

        case 3:jmas=3;
            break;
        case 4:jmas=4;
            break;
    }
    m=log(2);
    if (mas[imas][jmas]<m)
        b=m/mas[imas][jmas];
    else if(
mas[imas][jmas]<m*10)

b=((int)(10*m/mas[imas][jmas]))/10.0;
        else
b=((int)(100*m/mas[imas][jmas]))/100.0;

        gcvt((double)b,6,str);
        okruglenie(str);

setfillstyle(SOLID_FILL,LIGHTGRAY);

bar(300,frame[3].endx+50,450,frame[3].endx+100);

setlinestyle(SOLID_LINE,0,THICK_WIDTH);
        setcolor(LIGHTBLUE);
        rectangle(300-
2,frame[3].endx+50-2,
        450+2,frame[3].endx+100+2);
        setcolor(LIGHTCYAN);

        settextjustify(CENTER_TEXT,CENTER_TEXT)
;

        outtextxy(375,frame[3].endx+20,m3[language]);

        setcolor(BLUE);
        outtextxy(350,
frame[3].endx+75, strcat(str,m4[language]));
        flag2=1;
        }
        return;
    }
        setcolor(BLUE);
settextjustify(CENTER_TEXT,CENTER_TEXT);
outtextxy(155,50,"Выберите энергию
излучения");
setcolor(LIGHTCYAN);
settextjustify(CENTER_TEXT,CENTER_TEXT);
}
/*|-----|*/
/*|          Функция выбора языка          |*/
/*|-----|*/
void set_language()
{
    int choice;
    choice = pulldown(2);
    switch(choice)
    {
        case 0: language = RUSSIAN;
            break;
        case 1: language = ENGLISH;
            break;
    }
}
/*|-----|*/
/*|          Окно с сообщением          |*/
/*|-----|*/
void quit(char*m)
{
    char c;
    setfillstyle(SOLID_FILL,LIGHTCYAN);
    bar(200,200,430,250);
    setlinestyle(SOLID_LINE,1,3);
    setcolor(LIGHTBLUE);
    rectangle(200,200,430,250);

        setcolor(BLUE);
        outtextxy(215,215,m);
        outtextxy(290,230,"Y/N");
        for(;;)
        {
            c=getch();
            if( (c=='y') || (c=='Y') || (c=='h') ||
(c=='H'))
            {
                closegraph();
                exit(0);
            };
            if( (c=='n') || (c=='N') || (c=='т') ||
(c=='Т') )
            {
                setfillstyle(SOLID_FILL,BLUE);
                bar(195,195,435,255);
                return;
            }
        }
    }
}
void quit1(char*m)
{
    int width;
    char c;
    settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
    setfillstyle(SOLID_FILL,LIGHTCYAN);
    width=220+textwidth(m);
    bar(200,200,width,250); /*430*/
    setlinestyle(SOLID_LINE,1,3);
    setcolor(LIGHTBLUE);
    rectangle(200,200,width,250);
    setcolor(BLUE);
    outtextxy(210,225,m);
    for(;;)
    {
        if (getch())
        {
            setfillstyle(SOLID_FILL,BLUE);
            bar(195,195,width+5,255);
            return;
        }
    }
}
/* Программа модельного эксперимента*/

#include <stdio.h>
#include <time.h>
#include <string.h>
#include <graphics.h>
#include <stdlib.h>
#include <conio.h>
#include <bios.h>
#include <math.h>
#include "stack.h"
#include <alloc.h>

#define IM 100;

extern int language;
int nf,period,flag,flag2;
void *imaget;
char* m1[2] =
{
    " Нажмите любую клавишу для остановки ",
    " Press any key to stop "
};
void opyt()
{
    extern int nf_nach;
    extern int pulldown(int);
    extern float vvod_chisla(int,int,int*);
    extern unsigned long counts;

```

```

extern void set_time();
extern void close_time();
extern char * s;
extern FILE * in;
void run_menu(int,int,int,int);
extern float b, mas[9][5];
extern int imas, jmas;
char* filename="";
int
x=20,y=20,x1,rand_num,i,y1,j,num,n,userfont,ty
peline,typeline1;
int Gd=DETECT,Gm,err, maxx, maxy,
fon=BLUE,
linec=YELLOW,
text=LIGHTCYAN,
plast=LIGHTCYAN,
impuls=LIGHTGREEN;
char str[15];
float eps;
double r;
unsigned size;
char* m[2] =
{ "см","см"
};
char str_o[5][12]=
{"Свинцовые",
"Железные",
"Медные",
"Алюминиевые",
"Бетонные"};
maxx=getmaxx();
maxy=getmaxy();
setcolor(text);
setbkcolor(fon);
if(lenguage)
filename="opyt1_e.txt";
else filename="opyt1.txt";
if ((in=fopen(filename,"r"))==NULL)
{
printf("Файл opyt1 не открыт!!!\n");
exit(-1);
};
userfont=installuserfont("simp.chr");
settextstyle(userfont,HORIZ_DIR,4);
settextjustify(LEFT_TEXT,CENTER_TEXT);
fgets(s,80,in);
outtextxy(x,y,s); /* Модельный эксперимент */
y=y+70;
setfillstyle(SOLID_FILL,linec);
bar(x,y+100,x+80,y+180);
settextstyle(userfont,HORIZ_DIR,1);
fgets(s,80,in);
outtextxy(x,y+195,s); /* Камера */
fgets(s,80,in);
outtextxy(x,y+215,s); /* с источником */
fgets(s,80,in);
outtextxy(x,y+235,s); /* излучения */
setcolor(fon);
setfillstyle(SOLID_FILL,fon);
pieslice(x+60,y+140,0,360,10);
setcolor(LIGHTRED);
setfillstyle(SOLID_FILL,LIGHTRED);
sector(x+60,y+140,0,360,5,3);
setfillstyle(SOLID_FILL,plast);
bar(x+150,y,x+349,y+90);
setcolor(text);
settextstyle(userfont,HORIZ_DIR,1);
gcvt((double)b,6,str);
strcat(str,m[lenguage]);/*" см"*/
fgets(s,80,in);
outtextxy(x+85,y-25,str_o[imas]);
strcat(s," ");
strcat(s,str);
outtextxy(x+85+10+textwidth(str_o[imas]),y-
25,s);
/* Пластины исследуемого материала, толщиной */
setcolor(fon);
x1=x+150;
setlinestyle(SOLID_LINE,1,NORM_WIDTH);
for(i=1;i<10;i++)
{
x1=x1+20;
line(x1,y,x1,y+90);
}
size=imagesize(x+151,y-2,x+170,y+90);
imaget=(void*)malloc(size);
getimage(x+151,y-2,x+170,y+90,imaget);
setcolor(linec);
x1=x+50;
setlinestyle(SOLID_LINE,1,THICK_WIDTH);
line(x1+400,y+80,x1+400,y+200);
line(x1+450,y+80,x1+450,y+200);
rectangle(x1+390,y+200,x1+460,y+210);
setfillstyle(SOLID_FILL,fon);
sector(x1+425,y+80,0,180,25,50);
settextstyle(userfont,HORIZ_DIR,1);
setcolor(text);
fgets(s,80,in);
outtextxy(x1+460,y+80,s); /* Счетчик */
fgets(s,80,in);
outtextxy(x1+460,y+100,s); /* Гейгера */
setcolor(fon);
line(x1+402,y+80,x1+448,y+80);
setcolor(linec);
line(x1+425,y+80,x1+425,y+200);
line(x1+425,y+210,x1+425,y+240);
line(x1+415,y+230,x1+425,y+240);
line(x1+435,y+230,x1+425,y+240);
setlinestyle(SOLID_LINE,1,NORM_WIDTH);
y1=y+2;
rectangle(x1+380,y1+240,x1+475,y1+270);
rectangle(x1+382,y1+242,x1+473,y1+268);
rectangle(x1+260,y1+240,x1+355,y1+270);
rectangle(x1+262,y1+242,x1+353,y1+268);
line(x1+355,y1+254,x1+380,y1+254);
line(x1+355,y1+256,x1+380,y1+256);
settextstyle(userfont,HORIZ_DIR,2);
setcolor(text);
fgets(s,80,in);
outtextxy(x1+385,y1+283,s); /* Импульсы */
fgets(s,80,in);
outtextxy(x1+270,y1+280,s); /* Время */
setfillstyle(SOLID_FILL,LIGHTCYAN);
bar(0,maxy-30,maxx,maxy);
setcolor(BLUE);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
fgets(s,80,in);
s[strlen(s)-1]='\0';
outtextxy(10,maxy-20,s); /* Enter - выбор
вида работы,
"Фон" - измерение фонового
излучения,"Источник"-*/
fgets(s,80,in);
s[strlen(s)-1]='\0';
outtextxy(10,maxy-7,s); /* измерение
излучения с источником,
"Сброс"-сброс результатов,"Выход",Esc - выход
*/
settextstyle(userfont,HORIZ_DIR,2);
fclose(in);
run_menu(x1,y1,x,y);
free(imaget);
return;
}
/*|-----|*/
/*|          функция запуска меню          |*/
/*|-----|*/
void run_menu(int x1, int y1,int x,int y)
{

```

```

void mess();
void fon_measuring(int,int);
void radiant_measuring(int,int,int,int);
int i,choice,xt,maxx,maxy,userfont;
maxx=getmaxx();
maxy=getmaxy();
userfont=installuserfont("simp.chr");
while ((choice=pulldown(5))!=-1)
{
    setfillstyle(SOLID_FILL,BLUE);
    bar(x1+383,y1+243,x1+472,y1+267);
    bar(x1+263,y1+243,x1+352,y1+267);
    bar(x+150,y+93,x+349,y+90+95);
    setfillstyle(SOLID_FILL,LIGHTCYAN);
    bar(x+150,y,x+349,y+90);
    setfillstyle(SOLID_FILL,BLUE);
    setcolor(BLUE);
    xt=x+150;
    setlinestyle(SOLID_LINE,1,NORM_WIDTH);
    for(i=1;i<10;i++)
    {
        xt=xt+20;
        line(xt,y,xt,y+90);
    }
    setfillstyle(SOLID_FILL,YELLOW);
    bar(x+71,y+135,x+80,y+145);
    setfillstyle(SOLID_FILL,BLUE);
    mess();
    switch(choice)
    {
        case 0: fon_measuring(x1, y1);
            mess();
            break;
        case 1: radiant_measuring(x1, y1, x, y);
            mess();
            break;
        case 2: setfillstyle(SOLID_FILL,BLUE);
            bar(x1+383,y1+243,x1+472,y1+267);
            bar(x1+263,y1+243,x1+352,y1+267);
            bar(x+150,y+93,x+349,y+90+95);

            setfillstyle(SOLID_FILL,LIGHTCYAN);
            bar(x+150,y,x+349,y+90);
            setfillstyle(SOLID_FILL,BLUE);
            setcolor(BLUE);
            xt=x+150;

            setlinestyle(SOLID_LINE,1,NORM_WIDTH);
            for(i=1;i<10;i++)
            {
                xt=xt+20;
                line(xt,y,xt,y+90);
            }
            setfillstyle(SOLID_FILL,YELLOW);
            bar(x+71,y+135,x+80,y+145);
            setfillstyle(SOLID_FILL,BLUE);
            /*---Рисует строку сообщения-----*/
            mess();
            break;
        case 3: return ;
    }
}
}
/*|-----|*/
/*|          Строка сообщения          |*/
/*|-----|*/
void mess()
{
    extern FILE*in;
    extern char*s;
    int maxx,maxy;
    maxx=getmaxx();
    maxy=getmaxy();
    if(language) in=fopen("opyt_e.txt","r");
    else in=fopen("opyt.txt","r");
    fclose(in);
}
}
/*|-----|*/
/*|          Функция измерения фона          |*/
/*|-----|*/
void fon_measuring(int x,int y)
{
    extern int nf_nach;
    extern void set_time();

    extern void close_time();
    extern unsigned long counts;
    int nt,rand_num,maxx,maxy;
    char str[8],str1[8];
    float eps;
    maxx=getmaxx();
    maxy=getmaxy();
    settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
    setcolor(WHITE);
    setbkcolor(BLUE);
    setfillstyle(SOLID_FILL,BLUE);
    nt=0;
    str[0]=0;
    str1[0]=0;
    flag2=0;
    flag=0;
    bar(0,maxy-30,maxx,maxy);
    setfillstyle(SOLID_FILL,LIGHTCYAN);
    bar(0,maxy-15,maxx,maxy);
    settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
    setcolor(BLUE);
    outtextxy(50,maxy-7,m1[language]);/*" Нажмите
любую клавишу для останковки "*/
    settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
    setcolor(WHITE);
    setbkcolor(BLUE);
    setfillstyle(SOLID_FILL,BLUE);

    randomize();
    rand_num = random(1000);
    if(rand_num < 184)
        eps=0;
    else if(rand_num < 363)
        eps=0.01;
    else if(rand_num < 525)
        eps=0.02;
    else if(rand_num < 664)
        eps=0.03;
    else if(rand_num < 776)
        eps=0.04;
    else if(rand_num < 860)
        eps=0.05;
    else if(rand_num < 920)
        eps=0.06;
    else if(rand_num < 960)
        eps=0.07;
    else if(rand_num < 985)
        eps=0.08;
    else eps=0.09;
    if( random(1000) <500) eps=-1*eps;
    nf=nf_nach*(1+eps);
    period=1000/nf;
    setttextjustify(2,1);

    set_time();
    do
    {
        if(flag2)
        {
            itoa(counts,str1,10);
            bar(x+263,y+243,x+352,y+267);
            setcolor(LIGHTCYAN);
            outtextxy(345+78,348,str1);
            flag2=0;
        }
        if(flag)
        {

```



```

        if(random(1000)<184)
        {
            nt=nt+1;

            bar(x+383,y+243,x+472,y+267);
            itoa(nt, str,10);
            setcolor(LIGHTCYAN);
            outtextxy(460+80,348, str);
        }
        flag=0;
    }
}
while(!bioskey(1));
close_time();
settextjustify(0,1);
getch();
}

/*|-----|*/
/*|Функция измерения излучения источника |*/
/*|-----|*/
void radiant_measuring(int x1,int y1,int x,int
y)
{
    extern int nf_nach;
    extern void set_time();
    extern void close_time();
    void put_plate(int ,int ,int* );
    extern unsigned long counts;
    extern float b;
    extern float mas[9][5];
    extern int imas,jmas;
    int
typeline,typeline1,nt,ntl,rand_num,num,maxx,ma
xy,userfont;
    char str[8],str1[8];
    float eps, rab = 0, rab2 = 0, rab3 = IM;
    settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
    setcolor(WHITE);
    setbkcolor(BLUE);
    setfillstyle(SOLID_FILL,BLUE);
    maxx=getmaxx();
    maxy=getmaxy();
    userfont=installuserfont("simp.chr");
    str[0] = 0;
    str1[0] = 0;
    flag2 = 0;
    flag = 0;
    randomize();
    put_plate(x, y, &num);
    bar(x+70,y+135,x+80,y+145);
    rab2=mas[imas][jmas];
    rab=b*num*rab2;
    rab2=exp(-rab);
    rab2=rab3*rab2;
    nt=(int)(rab2);
    bar(0,maxy-30,maxx,maxy);
    setfillstyle(SOLID_FILL,LIGHTCYAN);
    bar(0,maxy-15,maxx,maxy);
    settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
    setcolor(BLUE);
    outtextxy(50,maxy-7,m1[language]);/*" Нажмите
любую клавишу для остановки "*/
    settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
    setcolor(WHITE);
    setbkcolor(BLUE);
    setfillstyle(SOLID_FILL,BLUE);
    userfont=installuserfont("simp.chr");

    rand_num = random(1000);
    if(rand_num < 184)

        eps = 0.0;
        else if(rand_num < 363)
            eps = 0.01;
        else if(rand_num < 525)
            eps = 0.02;
        else if(rand_num < 664)
            eps = 0.03;
        else if(rand_num < 776)
            eps = 0.04;
        else if(rand_num < 860)
            eps = 0.05;
        else if(rand_num < 920)
            eps = 0.06;
        else if(rand_num < 960)
            eps = 0.07;
        else if(rand_num < 985)
            eps = 0.08;
        else eps = 0.09;
        if( random(1000) <500) eps=-eps;
        nt=(int)nt*(1+eps);

/*-----Определение фонового излучения-----*/
    rand_num = random(1000);
    if(rand_num < 184)
        eps=0;
        else if(rand_num < 363)
            eps=0.01;
        else if(rand_num < 525)
            eps=0.02;
        else if(rand_num < 664)
            eps=0.03;
        else if(rand_num < 776)
            eps=0.04;
        else if(rand_num < 860)
            eps=0.05;
        else if(rand_num < 920)
            eps=0.06;
        else if(rand_num < 960)
            eps=0.07;
        else if(rand_num < 985)
            eps=0.08;
        else eps=0.09;
        if( random(1000) <500) eps=-1*eps;
        nf=nf_nach*(1+eps);
/*-----*/
    ntl=nf+nt;
    nt=0;
    period=1000/ntl;
    typeline=typeline1=0x1;
    settextjustify(2,1);
    set_time();
    do
    {
        setcolor(WHITE);

setlinestyle(USERBIT_LINE,typeline,NORM_WIDTH)
;
        line(x+80,y+140,x+150,y+140);
        line(x+150+num*20,y+140,x1+395,y+140);
        line(x+80,y+140,x+150,y+140+12);
        line(x+150+num*20,y+160,x1+395,y+140+48);
        line(x+80,y+140,x+150,y+140+6);
        line(x+150+num*20,y+150,x1+395,y+140+24);

setlinestyle(USERBIT_LINE,typeline1,NORM_WIDTH)
);
        line(x+80,y+140,x+150,y+140-12);
        line(x+150+num*20,y+130,x1+395,y+140-24);
        line(x+80,y+140,x+150,y+140-6);
        line(x+150+num*20,y+120,x1+395,y+140-48);
        setcolor(BLUE);

setlinestyle(USERBIT_LINE,typeline,NORM_WIDTH)
;
        line(x+80,y+140,x+150,y+140);
        line(x+150+num*20,y+140,x1+395,y+140);
        line(x+80,y+140,x+150,y+140+6);
        line(x+150+num*20,y+150,x1+395,y+140+24);
        line(x+80,y+140,x+150,y+140+12);
        line(x+150+num*20,y+160,x1+395,y+140+48);

```

```

setlinestyle(USERBIT_LINE,typeline1,NORM_WIDTH
);
line(x+80,y+140,x+150,y+140-6);
line(x+150+num*20,y+130,x1+395,y+140-24);
line(x+80,y+140,x+150,y+140-12);
line(x+150+num*20,y+120,x1+395,y+140-48);
typeline=typeline<<1;
if(!typeline) typeline=0x1;
typeline1=typeline1>>1;
if(!typeline1) typeline1=0x80;
if(flag2)
{
itoa(counts,str1,10);
bar(x1+263,y1+243,x1+352,y1+267);
setcolor(LIGHTCYAN);
outtextxy(345+77,348,str1);
flag2=0;
}
if(flag)
{
if(random(1000)<184)
{
nt=nt+1;
bar(x1+383,y1+243,x1+472,y1+267);
itoa(nt,str,10);
setcolor(LIGHTCYAN);
outtextxy(460+80,348,str);
}
flag=0; }
}
while(!bioskey(1));
close_time();
settextjustify(0,1);
getch();
}
/*|-----|*/
/*|        Функция опускает пластины        |*/
/*|-----|*/
void put_plate(int x,int y,int* num)
{
extern float vvod_chisla(int,int,int*);
int i, j, n, err,err1,maxx,maxy;
unsigned size;
char bufer[15];
extern void *imaget;
extern char *s;
extern FILE * in;
char* m[2]=
{
"Введите число пластинок (0-10):",
"Put the number of plates (0-10):"
};
maxx=getmaxx();
maxy=getmaxy();
bar(0,maxy-30,maxx,maxy);
setcolor(WHITE);
outtextxy(15,getmaxy()-
15,m[language]);/*"Введите число
пластинок:";*/
*num=(int)(vvod_chisla(520, 465, &err));
if(*num > 10) *num=10;
setcolor(BLUE);
line(x+150,y,x+150,y+90);
if(*num!=0)
{
for(i=1;i<=95;i++)
{
putimage(x+151,y-2+i,imaget,COPY_PUT);
delay(5);
}
}
n=20*( *num-2);
for(j=0;j<=n;j=j+20)
{
for(i=1;i<=95;i++)
{
putimage(x+171+j,y-
2+i,imaget,COPY_PUT);
delay(5);
}
}
}
/* ТАЙМЕР */
#include <dos.h>
#include <stdlib.h>
#include <time.h>
#include <graphics.h>
#define MS_PER_TICK 53 /* число
миллисекунд в одном "тике" таймера */
unsigned long count, count1, counts;
unsigned ticks;
void interrupt timer(),
interrupt(*old_handler)();

/*-----*/
/* Функция задания режима работы таймера */
/*-----*/

void set_time()
{
count = 0;
ticks = 0;
counts = 0;
count1 = 0;
/* запоминание адреса текущего вектора
прерывания таймера*/
old_handler = getvect(8);
/* установка нового вектора прерывания
таймера */
setvect(8,timer);
/* разрешить контроллеру 8253 прерывание
INT0 (таймер) */
outportb(0x21, inportb(0x21)&0xfe);
/* установить режим работы таймера */
outportb(0x43,0x34);
/* установка константы таймера */
outportb(0x40,0xcd);
outportb(0x40,0x04);
}

/*-----*/
/* Функция восстановления старого режима
работы таймера */
/*-----*/

void close_time()
{
outportb(0x43, 0x34);
outportb(0x40, 0);
outportb(0x40, 0);
setvect(8, old_handler);
}

/*-----*/
/* Функция обработки прерывания таймера */
/*-----*/

void interrupt timer()
{
extern int flag;
extern int flag2;
extern int period;
count++;
count1++;
/* передача управления таймеру MS-DOS
приблизительно 18 раз в сек */
if(++ticks == MS_PER_TICK)
{

```

```
    old_handler();  
    ticks = 0;  
}  
if (count==period)  
{  
    count=0;  
    flag = 1;  
}  
if (count1 ==1000)
```

```
{  
    counts++;  
    flag2=1;  
    count1=0;  
}  
/* разрешение прерывания */  
outportb(0x20,0x20);  
}
```

## ПРИЛОЖЕНИЕ 3

## Модельная лабораторная установка

