

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

На конкурс на лучшую работу студентов по разделу
Техническая кибернетика, информатика и вычислительная техника

СТУДЕНЧЕСКАЯ НАУЧНАЯ РАБОТА

На тему: «Электронная лабораторная работа на тему
" Изучение поглощения γ -излучения в веществе"»

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

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

ст. пр. кафедры физики Товстухи В.С.

2002 г.

РЕФЕРАТ

Отчет содержит с., рис., таблицы, 2 приложения, источника.

Объект исследования – методика проведения виртуального эксперимента по определению линейный коэффициент поглощения γ -излучения в различных веществах.

Цель работы – разработать программу, реализующую виртуальную лабораторную работу по теме "Изучение поглощения γ -излучения в веществе".

Метод исследования – ознакомление с основными законами поглощения γ -излучения в веществе, написание алгоритма моделирующего данную лабораторную работу и последующая разработка программы, реализующей данный алгоритм.

Результат работы – отчет, описывающий методику проведения виртуальной лабораторной работы по физике, алгоритм моделирующий данную лабораторную работу и последующая разработка программы, реализующей данный алгоритм.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 ОБЗОР	5
2 ПОСТАНОВКА ЗАДАЧИ	6
2.1 Исходные данные	6
2.2 Ограничения	6
2.3 Результаты	6
2.4 Связь	6
3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ЭКСПЕРИМЕНТА	7
3.1 Метод	7
3.2 Блок-схема алгоритма	10
4 СТРУКТУРА МОДУЛЕЙ ПРОГРАММЫ	13
4.1 Структура программы	13
4.2 Описание модулей	14
ЗАКЛЮЧЕНИЕ	16
ПЕРЕЧЕНЬ ССЫЛОК	17
ПРИЛОЖЕНИЕ А. ЛИСТИНГ ПРОГРАММЫ	18
ПРИЛОЖЕНИЕ Б. ПРИМЕР ИСПОЛЬЗОВАНИЯ	29

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.

В настоящее время компьютерное моделирование физических процессов широко применяется. Существуют программы моделирующие различные физические явления, программы контроля знаний и программы, предназначенные для обработки данных измерений, для получения результатов и оценки погрешности [1].

В данной работе создана модельная компьютерная лабораторная работа 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. Программа также содержит систему тестирования и систему обработки полученных результатов по методу наименьших квадратов.

1 ОБЗОР

В проведении физического практикума в вузах можно выделить следующие тенденции:

- использование реальных физических установок для проведения эксперимента;
- использование ЭВМ для сбора, обработки и анализа результатов физического эксперимента;
- использование учебных систем-тренажеров [2];
- полная реализация физического эксперимента с помощью компьютерного моделирования [3].

Каждый из подходов имеет свои преимущества и недостатки. Использование реальных физических установок для проведения лабораторных работ требует больших затрат и целесообразно их использовать в вузах, где физика является профилирующим предметом. Путем создания физических тренажеров можно удешевить и упростить проведение физического эксперимента. При этом достигается высокая степень безопасности и экологическая чистота эксперимента. В тренажерных системах органы управления и отображение информации осуществляется также как и в реальной физической установке, а физическая среда реализована путем компьютерного моделирования. Но они требуют разработки достаточно сложных технических и программных средств, и не каждый вуз имеет возможность приобретения таких систем.

Последний подход к проведению физического практикума позволяет использовать современные возможности компьютеров для моделирования эксперимента. Он легко реализуем в вузах хорошо оснащенных вычислительной техникой. На одном компьютере можно провести большое количество экспериментов с последующей обработкой результатов и их визуализации в удобной форме. Использование компьютера также позволяет проводить предварительный контроль знаний студентов.

2 ПОСТАНОВКА ЗАДАЧИ

2.1 Исходные данные

Исходными данными в программе является прямоугольный массив данных `mas[9][5]` типа `float`, содержащий значения линейного коэффициента поглощения гамма-лучей в веществе. Данные в каждой строке массива соответствуют определенной энергии излучения. Столбцы соответствуют одному из пяти веществ (свинец, железо, медь, алюминий, бетон). Текстовые данные содержатся в файлах прямого и последовательного доступа.

2.2 Ограничения

Ограничения состоит в том, что величина энергии излучения и количество различных веществ ограничено и встроено в программу. Это ограничивает выбор пользователя параметров эксперимента.

2.3 Результат

Результатом программы является компьютерное моделирование физического эксперимента по определению линейного коэффициента поглощения гамма-излучения в веществе.

2.4 Связь

Пользователь при помощи графического интерфейса, представляющего собой вертикальное меню. Перемещение по меню осуществляется стандартным образом с помощью клавиш "Стрелка вверх", "Стрелка вниз". Выбор опции меню осуществляется с помощью клавиши `Enter`. Некоторые опции носят информационный характер ("О программе", "Порядок выполнения работы", "Помощь"). Информация содержится в файлах прямого доступа.

3 РЕАЛИЗАЦИЯ ЛАБОРАТОРНОЙ РАБОТЫ

3.1 Метод

Физические сведения. Гамма излучение представляет собой жесткое электромагнитное излучение, обладает большой проникающей способностью и проходит в воздухе сотни метров, а в твердых телах – сантиметры или даже дециметры. Гамма – излучение характеризуется интенсивностью – I – величиной, численно равной количеству гамма-квантов, проходящих в единицу времени через поверхность единичной площади. При прохождении через слой вещества толщиной x интенсивность гамма-излучения ослабляется по закону: $I = I_0 \exp(-\mu x)$ (*), где μ - линейный коэффициент поглощения гамма-лучей в веществе. 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 прибора: $N = k I t$, где N – количество импульсов, зарегистрированных за время t , а коэффициент пропорциональности k определяется параметрами счетчика. Поэтому уравнение (*) можно записать в следующем виде: $\ln N = \ln N_0 - \mu x$.

Модельная установка лабораторной работы (источник гамма-излучения, набор из 10 пластин исследуемого материала, счетчик Гейгера со счетным устройством, циферблаты секундомера и счетчика импульсов) выводится на экран при помощи графических средств TurboC.

Программная реализация. Метод реализации данного физического процесса программно состоит в использовании прерываний DOS (модуль timer.c). Процесс измерений интенсивности гамма-излучения реализован в функциях fon_measuring и radiant_measuring (модуль opyt.c).

Сначала пользователь выбирает энергию гамма-излучения (imas) источника и исследуемый материал (jmas). Из матрицы – mas[9][5], содержащей значения коэффициента поглощения, берется значение соответствующее выбранным параметрам. Далее рассчитывается толщина

одной пластины исследуемого вещества – b – с учетом половинного ослабления интенсивности гамма-излучения. Также рассчитывается N_{ϕ} :

$$nf_nach=5+(int)(16*random(1000)/1000.0)$$

– количество импульсов, зарегистрированных за время 1с. и соответствующее фоновому излучению. Значение nf_nach соответствует природному, т.е. изменяется от 5 до 21. Оно высчитывается при каждом вхождении в программу и остается постоянным в течении всего сеанса работы. При каждом обращении к опции «Фон» вычисляется nf : $nf=nf_nach*(1+eps)$, где eps изменяется от -1 до 1 , т.е. вводится погрешность для измерения фонового излучения. Для измерения излучения источника также рассчитывается nt :

$$nt = N_0 * \exp(b * num * mas[imas][jmas])$$

- количество импульсов, зарегистрированных за время 1с., соответствующее излучению источника и вводится погрешность: $nt=(int)nt*(1+eps)$. num – число опущенных пластин; $N_0=100$ – константа для данной лабораторной работы. Далее вычисляется $nt1$:

$$nt1=nf+nt$$

- количество импульсов, зарегистрированных за время 1с., соответствующее реальному излучению при открытом источнике, учитывая фоновое излучение.

В зависимости от того в каком режиме находится пользователь («Фон» или «Источник»), вычисляется $period$:

$$period=1000/nf \text{ или } period=1000/nt1.$$

Константа 1000 соответствует 1000 миллисекундам. $period$ – период времени, через которое счетчик Гейгера должен регистрировать гамма-квант после прохождения излучения (заданной энергии) через вещество (заданной толщины). После этого запускается функция интервального таймера - $set_time()$, которая работает по прерыванию с дискретностью 1 миллисекунда. В функции обработки прерывания таймера - $timer()$ – возвращает значения $flag$ и $flag2$. Если $flag=1$, то прошла секунда и на

циферблате секундомера выводится новое время, если $\text{flag2} = 1$, то прошло period мс. Так как реально счетчик Гейгера регистрирует примерно каждый пятый импульс, то новое значение числа зарегистрированных импульсов на циферблате счетчика импульсов выводится если $\text{gandom}(1000) < 184$. Время проведения измерения не ограничено. После нажатия пользователем любой клавиши выполнение функции измерения заканчивается. Восстанавливается старый режим работы таймера. На циферблатах секундомера и счетчика импульсов останутся результаты измерения.

3.2 Блок-схема алгоритма

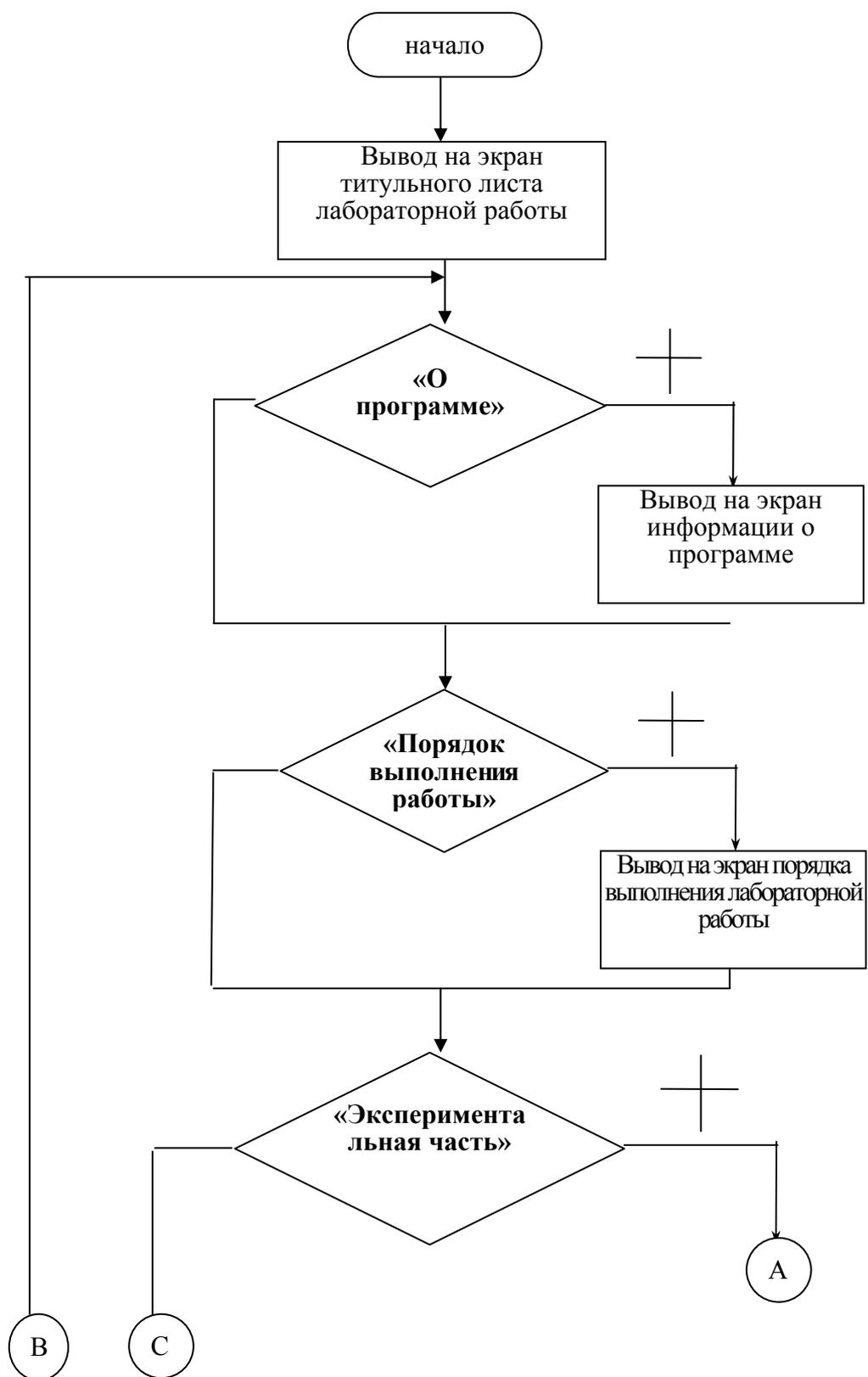


Рисунок 3.1-Блок-схема алгоритма

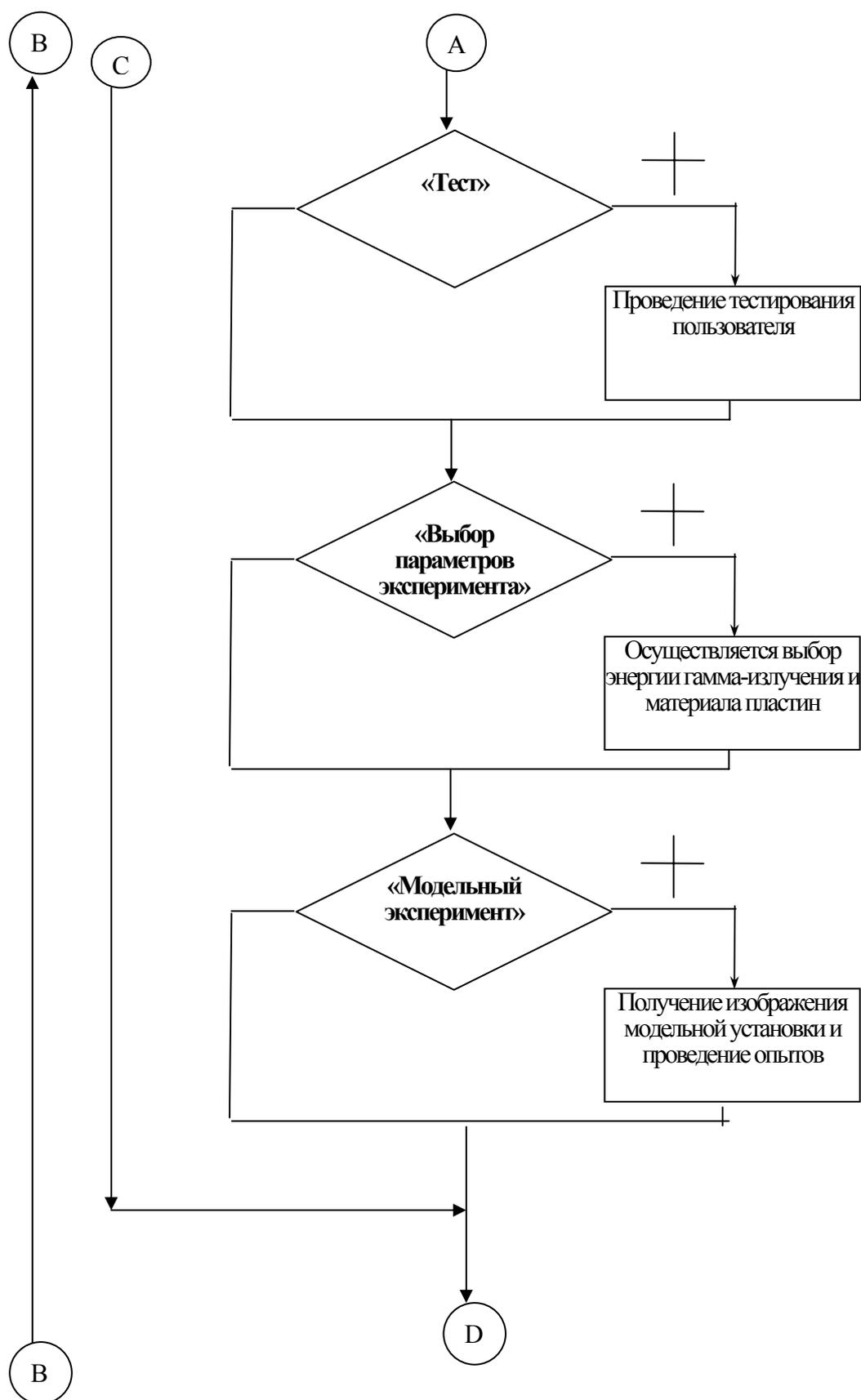


Рисунок 3.1-Блок-схема алгоритма, лист 2

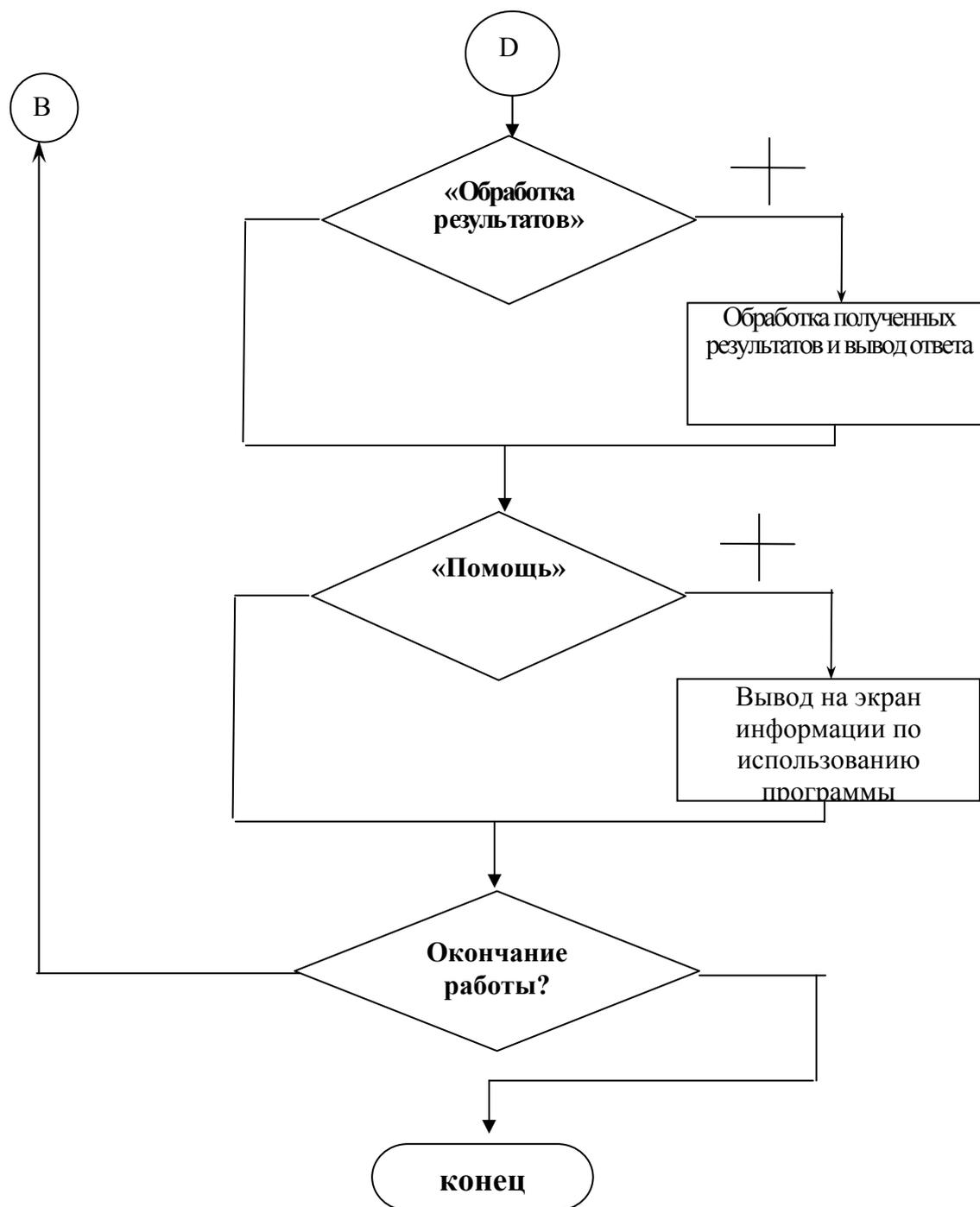


Рисунок 3.1-Блок-схема алгоритма, лист 3

4 СТРУКТУРА МОДУЛЕЙ ПРОГРАММЫ

4.1 Структура программы

Данная программа обращается к разным модулям, в которых объединены функции по своей тематике для упрощения написания и прочтения программы.

Структура программы приведена на рисунке 4.1, в пункте 4.2 описаны модули программы, к которым обращается программа в результате работы.

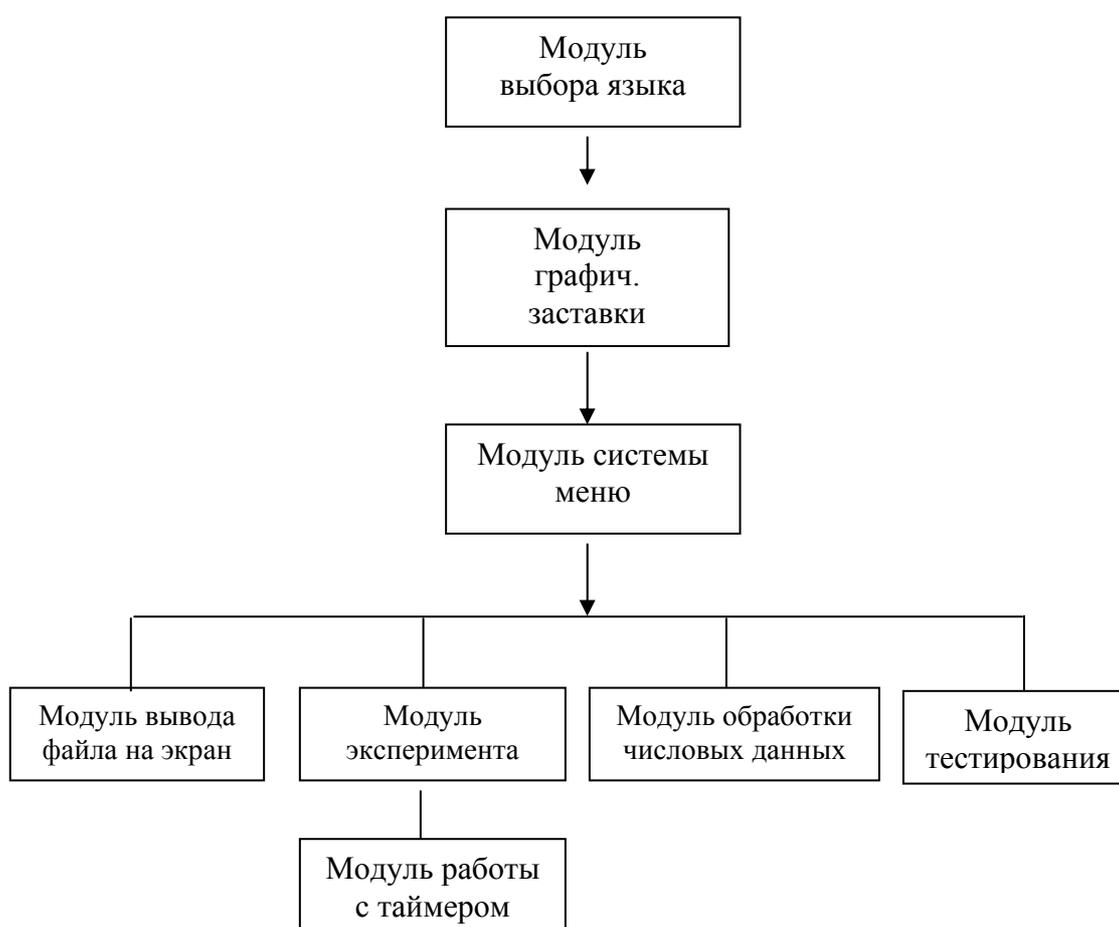


Рисунок 4.1 – Структура программы "Электронная лабораторная работа по теме "Изучение поглощения гамма-излучения в веществе" и взаимодействие ее модулей

4.2 Описание модулей

Программа состоит из нескольких модулей:

- Модуль выбора языка – language.c Пользователь с помощью меню выбирает язык общения – русский или английский.
- Модуль графической заставки – zastavka.c – отвечает за вывод на экран информации о названии работы и анимационный элемент на тему работы.
- Модуль системы меню – mmenu.c – отвечает за выбор опций программы. Таких, как «О программе», «Порядок выполнения работы», «Экспериментальная часть», «Обработка результатов», «Помощь», «Выход». Опция «Экспериментальная часть» включает в себя опции «Тест», «Выбор параметров эксперимента», «Модельный эксперимент». «Выбор параметров эксперимента» реализуется с помощью меню. Здесь пользователь выбирает энергию гамма-излучения и исследуемый материал.
- Модуль вывода файла на экран – scr. – отвечает за вывод файла прямого доступа на экран в режиме скроллинга.
- Модуль эксперимента – opyt.c – отвечает за проведение виртуального эксперимента. Появляется заставка с изображением экспериментальной установки. Показаны источник гамма-излучения, 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:
 - 1) Сброс. Эта опция возвращает систему в исходное состояние с обнулением счетчиков и переводом пластин в исходное состояние.
 - 2) Фон. В этом режиме начинается счет импульсов счетчика Гейгера с закрытым источником излучения. Время измерения определяется пользователем.
 - 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.

4) Выход в главное меню.

Доступ к этому модулю открыт если пользователь выбрал параметры эксперимента и прошел тест.

- Обработка результатов измерений. Появляется таблица, в которую пользователь должен занести результаты измерений – время, число зарегистрированных импульсов, количество пластин. 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.

- Модуль тестирования – test.c – отвечает за тестирование пользователя на тему лабораторной работы. Пользователю предлагается ответить на пять вопросов, выбранных из имеющихся десяти случайным образом. Также предлагаются 4-5 вариантов ответа, из которых несколько могут быть верными. Ответы к каждому вопросу также выводятся в произвольном порядке. Свои ответы пользователь вводит в специальное окно. Неправильные ответы выделяются цветом. После завершения тестирования программа выдает результат в процентах: более 50% - тест пройден, иначе – не пройден.

- Модуль работы с таймером – timer.c – Обеспечивает работу с таймером.

ЗАКЛЮЧЕНИЕ

Данная программа обладает удобным и понятным графическим интерфейсом, который обеспечивает диалог с пользователем при помощи меню.

Получаемые результаты полностью соответствуют результатам, получаемым при реальном проведении опыта. Программа также предоставляет пользователю возможность обработать полученные результаты по методу наименьших квадратов.

Какие либо изменения в текстовых файлах, изменения проведения эксперимента можно осуществлять только в текстовом редакторе языка Си.

Программа может использоваться для проведения лабораторных работ, а также для демонстрации физического опыта на лекциях по данной теме. Использование данной программы помогает значительно упростить учебный процесс.

ПЕРЕЧЕНЬ ССЫЛОК

1. Синельник И.В., Пятков А.Г., Сук А.Ф. Применение персональных ЭВМ при обработке результатов лабораторного эксперимента в режиме диалога./ Нові технології навчання – 1992. – Науково-методичний збірник. – Київ, НМК ВО, 1992. – с.156-160.

2. Белавин В.А., Квливидзе В.А., Петухов В.П., Радченко В.В. Новые принципы создания учебных лабораторных установок по атомной и ядерной физике. – Москва, МГУ, НИИЯФ. – 2002.

3. Горин В.В., Ильин В.А., Петрова Е.Б. Адаптация современного физического эксперимента к условиям специального физического практикума педагогического вуза. - Москва, МПГУ, физический факультет. – 2002.

ПРИЛОЖЕНИЕ А

ЛИСТИНГ ПРОГРАММЫ

Приведен текст функций, реализующих модельный эксперимент.

```

/*|-----|*/
/*|          Файл Опыт.c - виртуальный эксперимент          |*/
/*|-----|*/

#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 NO 100; /* интенсивность гамма-излучения */

extern int language;
int nf, /* текущее число импульсов в секунду для фона */
    period,
    flag,
    flag2;
void *imaget;
char* m1[2] =
{
    " Нажмите любую клавишу для остановки ",
    " Press any key to stop "
};

void opyt()
{
    /* функции управления меню */
    extern int pulldown(int);
    void run_menu(int,int,int,int);
    /* функции обеспечивает работу с таймером */
    extern void set_time();
    extern void close_time();
    extern float vvod_chisla(int,int,int*);
    extern char * s;
    extern FILE * in;
    extern int nf_nach; /* постоянное число импульсов в секунду для фона */
    extern unsigned long counts;
    extern float b, /* толщина пластины */
        mas[9][5]; /* массив, содержащий значения линейного коэффициента */
                    /* поглощения гамма-лучей в веществе */
    extern int imas,jmas; /*индексы определяющие материал и энергию излучения, */
                        /* выбранную пользователем */

    char* filename="";
    int x=20,y=20,x1,y1,rand_num,i,j,num,n,userfont,typeline,typeline1;
    int 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(language)
filename="опыт1_e.txt";
else filename="опыт1.txt";
if ((in=fopen(filename, "r"))==NULL)
{
printf("Файл опыт1 не открыт!!!\n");
exit(-1);
};
/* изображение модельной установки */
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[language]); /* 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);
/*запуск меню*/
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)
    {
        /* автоматический «сброс»*/
        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:
            /* «сброс»*/
            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 ();
        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 ("опыт_e.txt", "r");
    else in=fopen ("опыт.txt", "r");
        /*-----Рисует строку сообщения-----*/
        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");*/
        /*-----*/
    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 остановки
";*/
settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
setcolor(WHITE);
setbkcolor(BLUE);
setfillstyle(SOLID_FILL,BLUE);
/* внесение погрешности */
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; /* определение периода (в миллисекундах) импульсов*/
                /* источника излучения */
settextjustify(2,1);
/* активизация таймера */
set_time();
do
{
    if(flag2)/* прошла секунда */
    {
        /*вывод в окне «Время,с» времени*/
        itoa(counts,str1,10);
        bar(x+263,y+243,x+352,y+267);
        setcolor(LIGHTCYAN);
        outtextxy(345+78,348,str1);
        flag2=0;
    }
    if(flag)/*прошло period миллисекунд*/
    {
        /*реально на счетчик Гейгера регистрирует*/
        /*каждую пятую гамма-частицу:*/
        if(random(1000)<184)
        {
            /*вывод в окне «Импульсы» числа зарегистрированных частиц*/
            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,maxy,userfont;
    char str[8],str1[8];
    float eps, rab = 0, rab2 = 0, rab3 = N0;

    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);
    /*расчет интенсивности излучения источника*/
    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");
    /* внесение погрешности */
    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);
/*-----*/
nt1=nf+nt;
nt=0;
period=1000/nt1;
typeline=typeline1=0x1;
settextjustify(2,1);

set_time();
do
{
  /*анимация - бегущие точки - «излучение»*/
  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) /* прошла секунда */
{
/*вывод в окне «Время,с» времени*/
itoa(counts,str1,10);
bar(x1+263,y1+243,x1+352,y1+267);
setcolor(LIGHTCYAN);
outtextxy(345+77,348,str1);
flag2=0;
}
if(flag) /* прошло period миллисекунд */
{
/*реально на счетчик Гейгера регистрирует*/
/*каждую пятую гамма-частицу:*/
if(random(1000)<184)
{
/*вывод в окне «Импульсы» числа зарегистрированных частиц*/
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,errl,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);
    }
}

/*-----*/
/* Файл Timer.c-обеспечивает работу с таймером по прирыванию*/
/*-----*/

#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 Б ПРИМЕР ИСПОЛЬЗОВАНИЯ

Рассмотрим пример использования программы.

При входе в программу возникает окно:

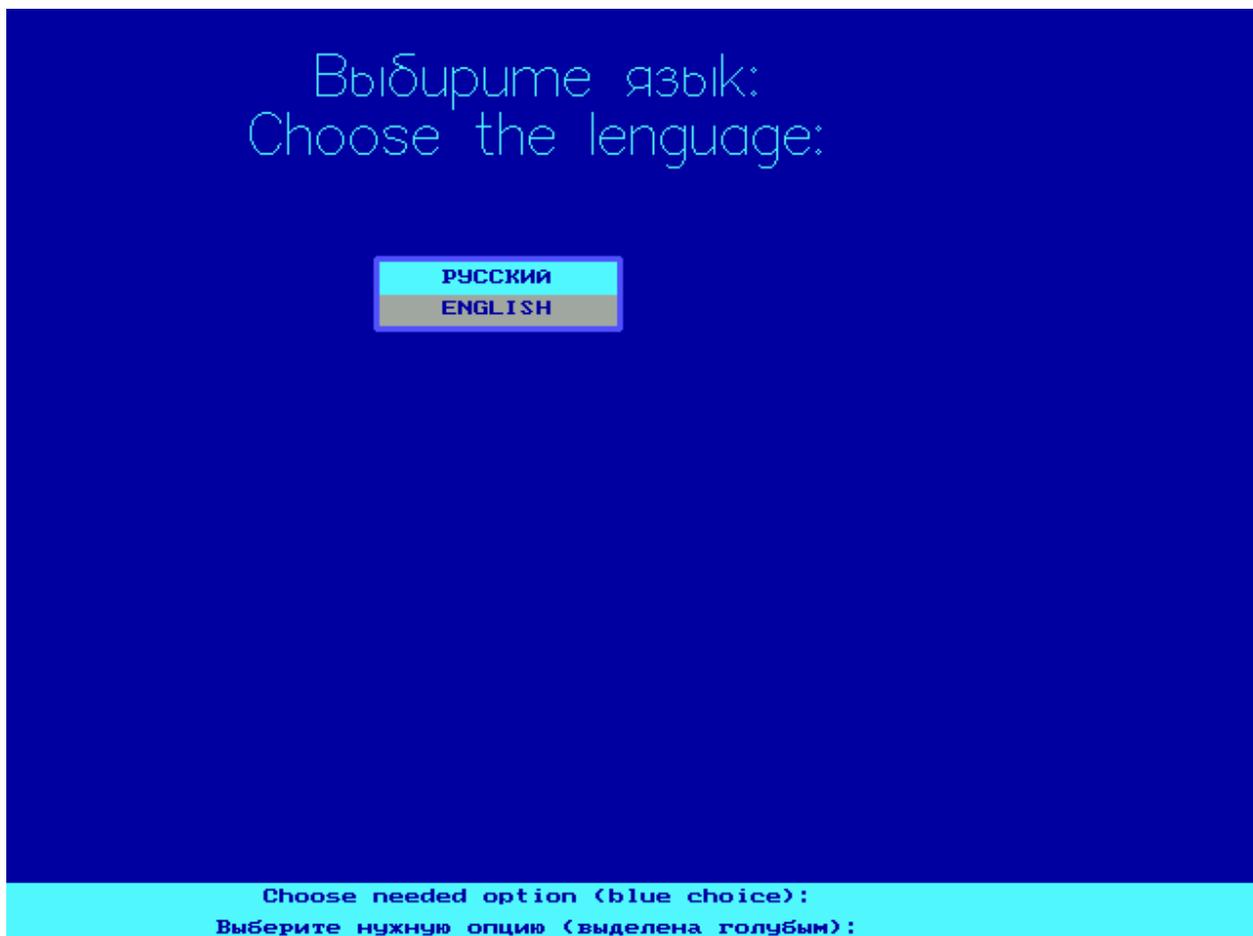


Рисунок Б.1 – Режим «Выбор языка»

Пользователю предлагается выбрать язык общения – русский или английский.

Возникает меню где пользователь должен выбрать нужный режим работы:

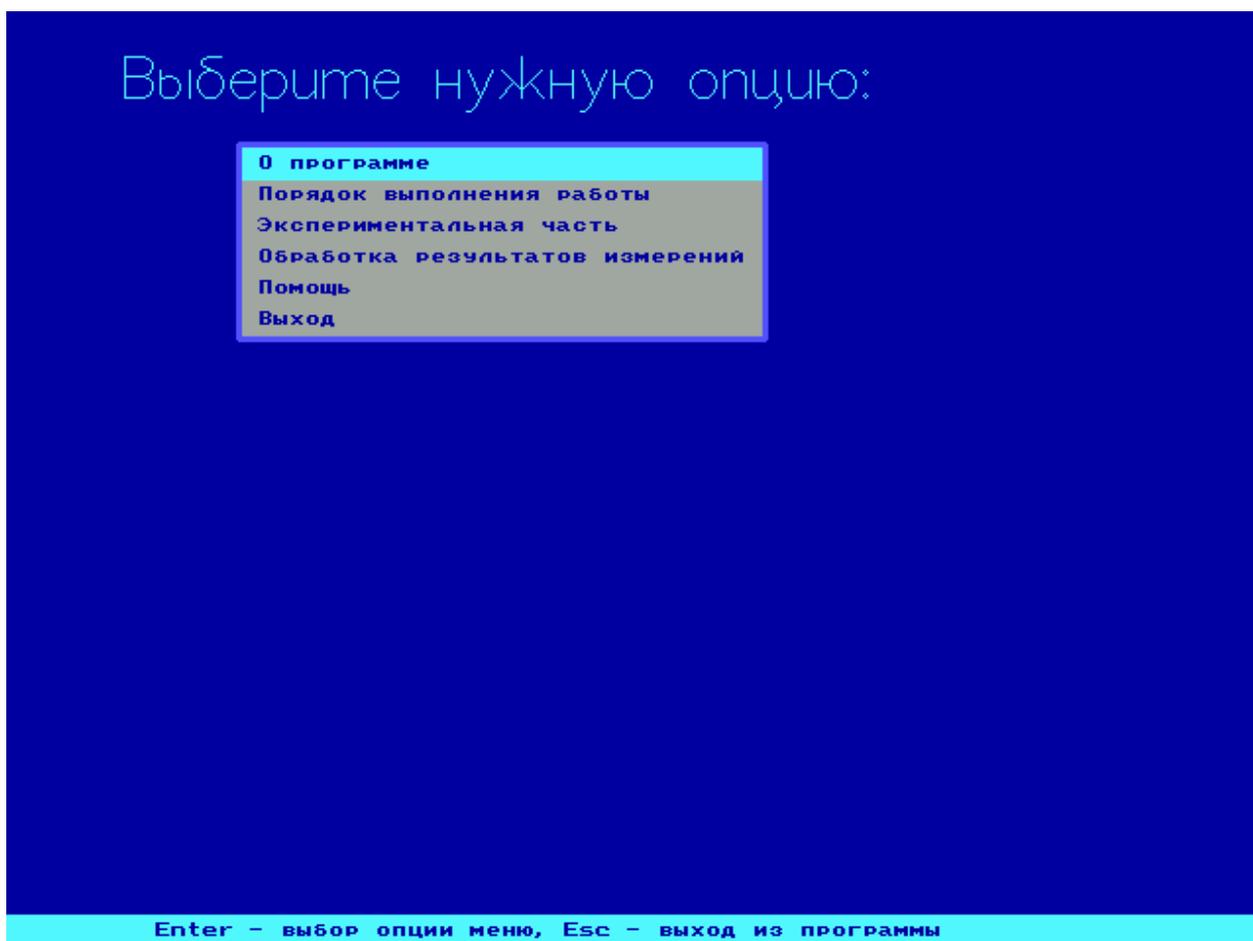


Рисунок Б.2 – Главное меню

Для проведения эксперимента пользователь должен выбрать его параметры – энергию гамма-излучения и исследуемый материал:

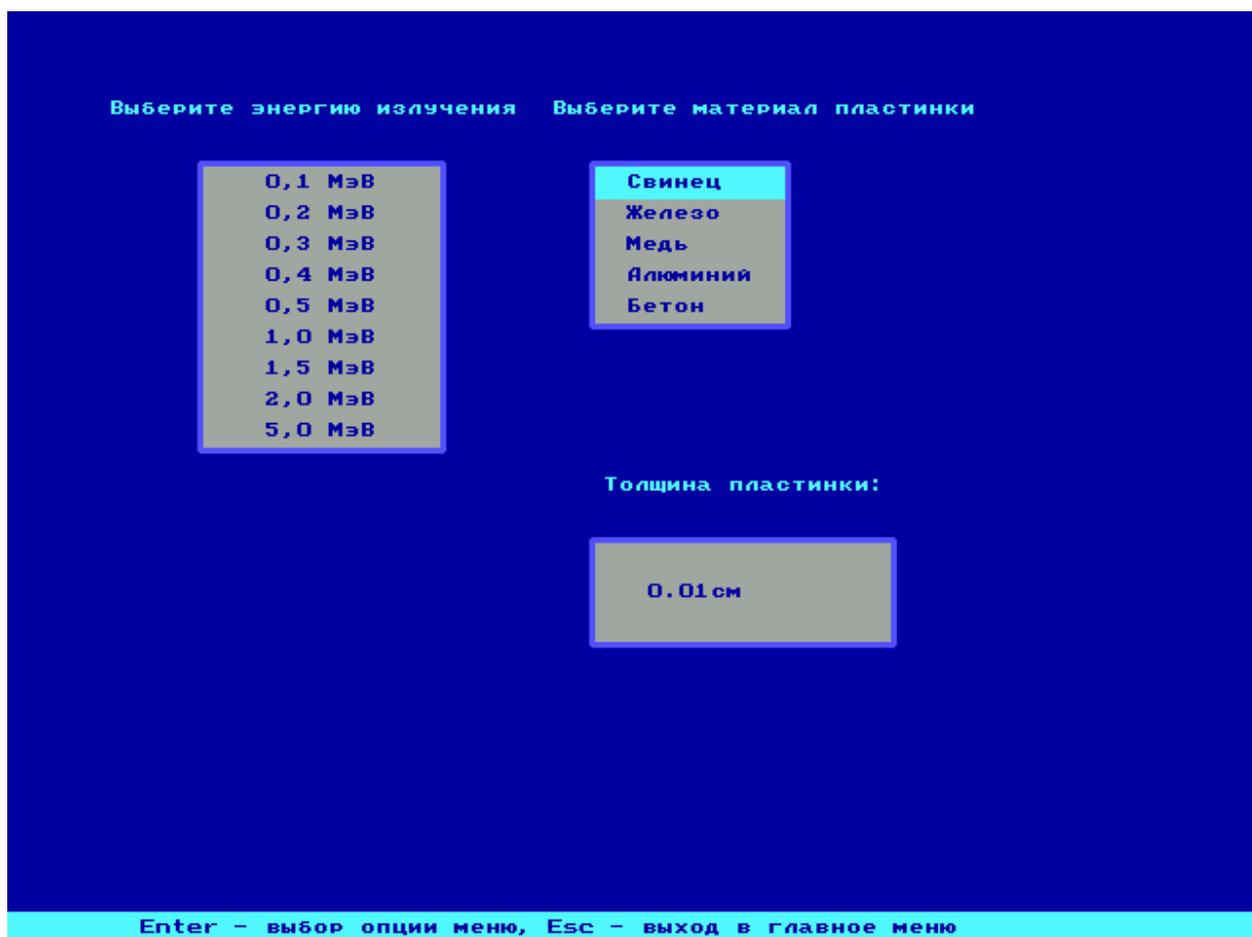


Рисунок Б.3 – Режим «Выбор параметров эксперимента»

Перед выполнением лабораторной работы пользователь должен пройти тест:

1. Какие частицы испускаются при радиоактивных превращениях?

1. — альфа—частицы;
2. — нейтроны;
3. — протоны.
4. — электроны;

Введите правильные ответы:

1	2	3	4	_
---	---	---	---	---

Синим цветом выделены неправильные ответы
Enter - ввод вариантов ответов, Esc - выход в главное меню

Рисунок Б.4 – Режим «Тест»

Проведение измерений:

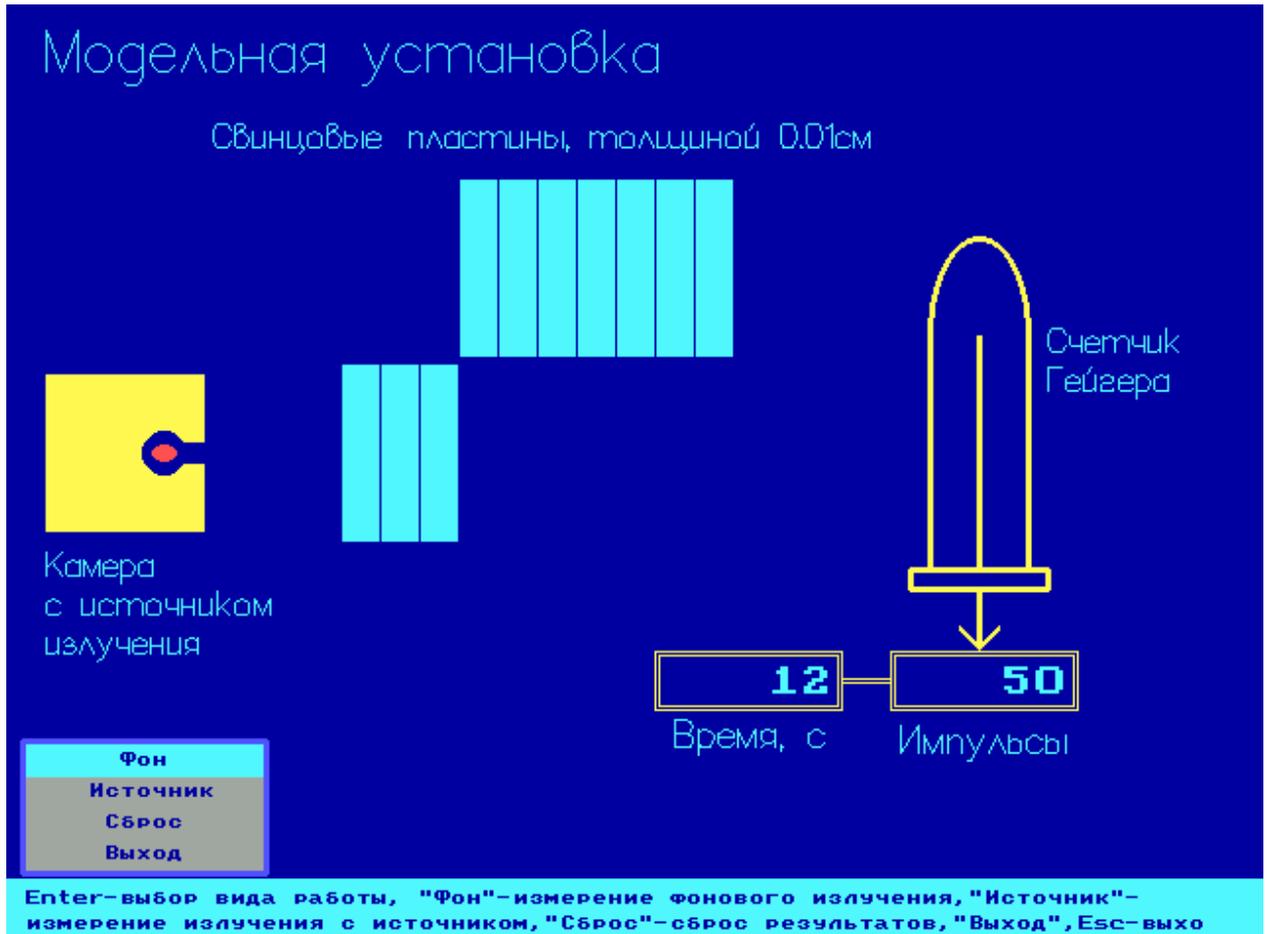


Рисунок Б.5 – Режим «Модельный эксперимент»

Пользователю предоставляется возможность обработать полученные результаты по методу наименьших квадратов:

Введите результаты измерений:

№ п/п	Время t регистрации импульсов, с	Число импульсов N зарегистрированных за время t	Количество поглощающих пластин
1	100	1973	2
2	100	298	6
3	100	135	7

Материал пластинки: Свинец
Энергия излучения: 0,1 МэВ
Толщина пластинки 0,01

Линейный коэффициент поглощения гамма-лучей в исследуемом веществе рассчитан по методу наименьших квадратов и имеет значение:

$$\mu = (51.81 \pm 22.64) \text{ см}^{-1}$$

с надежностью $\alpha = 95\%$.

Относительная погрешность составляет $\varepsilon = 0.43$

Enter - ввод числа (10 знаков), Ctrl-Z - ввод таблицы, Esc - выход

Рисунок Б.6 – Режим «Обработка результатов»