ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ  
Ордена Трудового Красного Знамени федеральное государственное   
бюджетное учреждение высшего образования

Московский технический университет связи и информатики

МТУСИ

Кафедра информационной безопасности

Лабораторная работа

Вариант 2

Выполнил студент группы БИН1703 ОТФ-2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бен Режеб Софиэн

Старший преподаватель кафедры ИТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Барков

**Практикум №6**  
**Структуры, объединения, перечисления и битовые поля**

**Цель практикума:** Овладеть навыками работы со структурами, объединениями, перечислениями и битовыми полями в языке C.

***Задание 1***

Разработать структуру «Зачётка». Структура включает в себя фамилию, имя, отчество студента, номер зачётной книжки, массив, хранящий информацию о каждом семестре. Размер массива равен количеству разворотов в зачётной книжке, предназначенных для проставления зачётов и экзаменов. Информация о семестре включает в себя номер семестра, учебный год и массив, хранящий информацию о предметах. Размер массива равен количеству строк в Вашей зачётке (левая и правая сторона). Информация о предмете включает в себя название предмета, тип оценки (зачет или экзамен), оценку, дату, фамилию преподавателя. Оценка является объединением из двух полей: логическое поле (зачёт или незачёт) и перечисление, представляющее оценку. Разработать консольное приложение, которое является электронной копией Вашей зачётки. Вся инфорация вводится статически (при желании можно вводить информацию с клавиатуры). Приложение должно показвать информацию об успеваемости студента по семестрам и при этом корретно различать экзамены и зачёты.

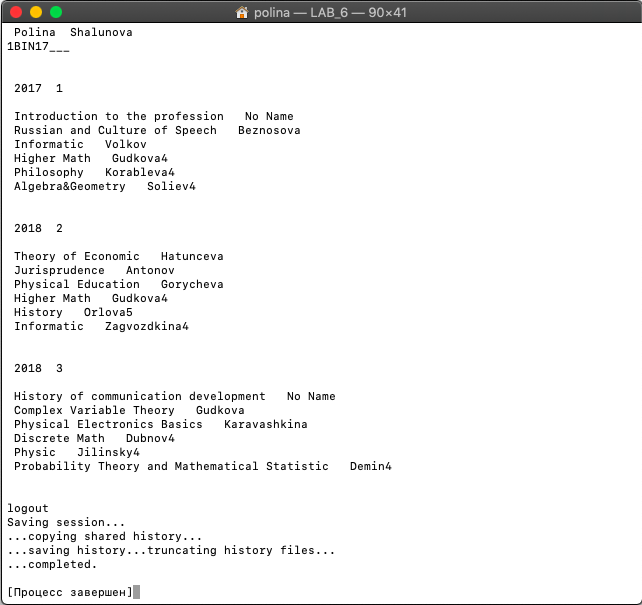


Рисунок 1

Листинг 1

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <math.h>

enum TypeOf{Cr,Ex};

enum Exam{o,F,D,C,B,A};

union MarkValue

{

\_Bool Passed;

enum Exam MyMark;

};

struct Mark

{

enum TypeOf Type;

union MarkValue Value;

};

struct SubjectInfo

{

char name[80];

char teacher[80];

struct Mark Type;

};

struct Info

{

int Year;

int Semester;

struct SubjectInfo Theory[5];

struct SubjectInfo Practice[4];

};

struct Person {

char FirstName[30];

char LastName[30];

char Number\_Rb[30];

struct Info Study[10];

};

void StudentID()

{

struct Person Me = {"Polina", "Shalunova", "1BIN17\_\_\_",

{

{2017, 1,

{{"Introduction to the profession","No Name",{Cr,{.Passed=1}}},

{"Russian and Culture of Speech","Beznosova",{Cr,{.Passed=1}}},

{"Informatic","Volkov",{Cr,{.Passed=1}}},

{"English","Gromova",{Cr,{.Passed=1}}},

{"Socio-ethical problems of information technology","Korableva",{Cr,{.Passed=1}}}},

{{"Higher Math","Gudkova",{Ex,{.MyMark=B}}},

{"Philosophy","Korableva",{Ex,{.MyMark=B}}},

{"Algebra&Geometry","Soliev",{Ex,{.MyMark=B}}},

}},

{2018,2,

{{"Theory of Economic","Hatunceva",{Cr,{.Passed=1}}},

{"Jurisprudence","Antonov",{Cr,{.Passed=1}}},

{"Physical Education","Gorycheva",{Cr,{.Passed=1}}},

{"Physic","Jilinsky",{Cr,{.Passed=1}}},

{"Technical Drawing","Evstratov",{Cr,{.Passed=1}}}},

{{"Higher Math","Gudkova",{Ex,{.MyMark=B}}},

{"History","Orlova",{Ex,{.MyMark=A}}},

{"Informatic","Zagvozdkina",{Ex,{.MyMark=B}}},

{"English","Kamenskaya",{Ex,{.MyMark=B}}},

}},

{2018,3,

{{"History of communication development","No Name",{Cr,{.Passed=1}}},

{"Complex Variable Theory","Gudkova",{Cr,{.Passed=1}}},

{"Physical Electronics Basics","Karavashkina",{Cr,{.Passed=1}}},

{"Electric Circuit Theory","Frisk",{Cr,{.Passed=1}}},

{"Introduction to Math Packages","Semenova",{Cr,{.Passed=1}}}},

{{"Discrete Math","Dubnov",{Ex,{.MyMark=B}}},

{"Physic","Jilinsky",{Ex,{.MyMark=B}}},

{"Probability Theory and Mathematical Statistic","Demin",{Ex,{.MyMark=B}}},

{"Computational Math","Semenova",{Ex,{.MyMark=C}}},

}},}};

printf("\n\n %s %s \n%s \n", Me.FirstName, Me.LastName, Me.Number\_Rb);

int m = 3;

for (int j = 0; j < m; ++j)

{

printf("\n\n %i %i\n", Me.Study[j].Year, Me.Study[j].Semester);

int n = 3;

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

{

printf("\n %s %s ", Me.Study[j].Theory[i].name, Me.Study[j].Theory[i].teacher);

printf("%c", Me.Study[j].Theory[i].Type.Value);

}

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

{

printf("\n %s %s", Me.Study[j].Practice[i].name, Me.Study[j].Practice[i].teacher);

printf("%i", Me.Study[j].Practice[i].Type.Value);

}

printf("\n");

}}

int main ()

{{

StudentID();

printf(" \n");

getchar();

}

}