|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ,,,,МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ | | | | | | | | | | | | |
| Федеральное государственное автономное образовательное учреждение  высшего образования | | | | | | | | | | | | |
| **Дальневосточный федеральный университет** | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ** | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Департамент программной инженерии и искусственного интеллекта** | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **О Т Ч Е Т** | | | | | | | | | | | | |
| по лабораторной работе №1  дисциплина «Основы алгоритмизации» | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | |  |  | Студент гр. Б9121-02.03.03тп | | | |
|  |  |  | | Плахотный А.В. | |
|  | | | | | | |  |  | (подпись) | |  | |
|  | | | | | | | | | | | | |
|  | | | | | | |  |  | Руководитель | | |  |
|  | | | | | | |  |  | ст. преподаватель | | | |
|  | | | |  |  | |  |  |  |  | С.Н. Остроухова | |
|  | | | |  |  | |  |  | (подпись) |  | (И.О. Фамилия) | |
|  |  |  |  | | |  |  |  |  | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| г. Владивосток | | | | | | | | | | | | |
| 2021 | | | | | | | | | | | | |

**Неформальная постановка задачи (НПЗ)**

В отделе кадров хранится информация о сотрудниках. Данные о каждом сотруднике содержат:

* Фамилию и инициалы
* Пол
* Профессию
* Дату рождения
* Дату последней аттестации

Проверить корректность входных данных. Вынести сообщения об ошибках в пустой текстовый файл outFalse. Вывести в файл outTrue полученный список верных строк, отсортированный по фамилии сотрудников и включающий в себя профессию и дату последней аттестации.

**Формальная постановка задачи (ФПЗ)**

X:

**In** – текстовый файл содержащий данные о сотрудниках (фамилия, инициалы, пол, профессия, дата рождения, дата последней аттестации)

Каждая строка имеет длину равную 70 символам и должна быть представлена в следующем формате:

**{Фамилия} {Инициалы} {Пол} {Профессия} {Дата рождения} {Дата аттестации}**

Каждый тип данных отделяется от другого пробелом и должен занимать строго определенную часть строки (фамилия 1-20 символ, инициалы 22-25 символ, пол 27 символ, профессия 29-48 символ, дата рождения 50 – 59 символ, дата последней аттестации 61-70 символ.)

**Фамилия, профессия** – должны быть записана латинскими буквами, начинаться с заглавной буквы, все остальные буквы должны быть строчными. Слово должно иметь длину, не превышающую 20 символов. Если длинна слова меньше 20 символов, то количество пробелов после этого слова будет равно (20 – i)+1, где i = длинна слова

**Инициалы** – Первый и третий символы должны являться заглавными латинскими буквами, а второй и четвертый символы должны быть точками.

Пол – должен быть записан заглавной латинской буквой “M” или “F”

**Дата рождения** – записывается в формате dd.mm.yyyy, где

dd = [01,.., 31]; mm= [01,..,12]; yyyy[1960,..,2000]

**Дата последней аттестации** - записывается в формате dd.mm.yyyy, где

dd = [01,.., 31]; mm= [01,..,12]; yyyy[1982,..,2022]

Y:

outtrue – текстовый файл содержащий в себе все строки из файла In которые были записаны корректно.

outfalse – текстовый файл содержащий в себе сообщения обо всех ошибках найденных в файле In.

Сообщение об ошибке записывается в следующем формате:

Ошибка в строке {a} = {s1} {err}

Где a – это номер строки, в которой найдена ошибка.

s1 – это содержимое той части строки в которой обнаружена ошибка

err – описание ошибки

Возможные описания ошибок в различных типах данных:

1. Фамилия

* Ошибка в первой букве фамилии
* - в фамилии запрещенный символ: + s1[i] + , номер символа: + i
* - в конце фамилии запрещенный символ:' + s1[i] + ', номер символа:' + i

2. Профессия

* - Ошибка в первой букве профессии
* в поле профессии запрещенный символ: + s1[i] + , номер символа: + i
* в конце поля профессии запрещенный символ:' + s1[i] + ', номер символа:' + i

3. Инициалы

* Ошибка инициалы должны быть записаны заглавными буквами
* Ошибка второй и четвертый символ должны быть точками

4. Пол

* Пол указан некорректно

5. Дата Рождения

* В поле "Дата рождения"(день) замечен(-ы) недопустимый(-ые) символ(-ы)
* В поле "Дата рождения"(месяц) замечен(-ы) недопустимый(-ые) символ(-ы)
* В поле "Дата рождения"(год) замечен(-ы) недопустимый(-ые) символ(-ы)
* В поле «Дата рождения» в значении год встречено недопустимое значение
* В поле «Дата рождения» в значении месяц встречено недопустимое значение
* В поле «Дата рождения» в значении день встречено недопустимое значение

6. Дата Аттестации

* В поле " Дата последней аттестации "(день) замечен(-ы) недопустимый(-ые) символ(-ы)
* В поле " Дата последней аттестации "(месяц) замечен(-ы) недопустимый(-ые) символ(-ы)
* В поле " Дата последней аттестации "(год) замечен(-ы) недопустимый(-ые) символ(-ы)
* В поле «Дата последней аттестации» в значении год встречено недопустимое значение
* В поле «Дата последней аттестации» в значении месяц встречено недопустимое значение
* В поле «Дата последней аттестации» в значении день встречено недопустимое значение

F:

1.Считать входные данные файла In.

2. С помощью процедур проверяем строки In на соответствие заданным критериям.

3. В текстовый файл outtrue выносим строки, которые соответствуют заданным критериям.

4. В текстовый файл outfalse выносим сообщения об ошибках.

**Программный код**

Блок ourprocedures

**unit** ourprocedures;

**interface**

**uses** ourtype;

**procedure** check\_inic(**var** s1: string; **var** err: boolean);

**procedure** check\_fam(**var** s1: string; **var** err: boolean);

**procedure** check\_prof(**var** s1: string; **var** err: boolean);

**procedure** check\_pol(**var** s1: string; **var** err: boolean);

**procedure** check\_DR(**var** s1:string; **var** err:boolean);

**procedure** check\_DPA(**var** s1:string; **var** err:boolean);

**procedure** sort(**var** arr\_Sotr\_true:arr\_Sotr; kol\_true:integer);

**procedure** check\_prob(**var** s1: string; **var** err: boolean);

**implementation**

**procedure** check\_fam(**var** s1: string; **var** err: boolean);

**var**

i: integer;

**begin**

err := false;

**if not** (s1[1] **in** ['A'..'Z'])

**then begin**

err := true;

s1 := s1 + ' - Ошибка в первой букве фамилии';

**end**

**else begin**

i:=2;

**while** (i<=length(s1))**and** (s1[i]<>' ') **and** (err=false) **do**

**begin**

**if not** (s1[i] **in** ['a'..'z', 'A'..'z', '-'])

**then begin**

err := true;//проверка на запрещенные символы

s1 := s1 + ' - в фамилии запрещенный символ:' + s1[i] + ', номер символа:' + i;

**end**;

inc(i);

**end**;

**while** (i<=length(s1))**and** (err=false) **do begin**

**if** s1[i]<>' '

**then begin**

err := true;//проверка на запрещенные символы

s1 := s1 + ' - в конце фамилии запрещенный символ:' + s1[i] + ', номер символа:' + i;

**end**;

inc(i);

**end**;

**end**;

**end**;

**procedure** check\_inic(**var** s1: string; **var** err: boolean);

**begin**

err := false;

**if not** (s1[1] **in** ['A'..'Z']) **and not** (s1[3] **in** ['A'..'Z'])

**then begin**

err := true;

s1 := s1 + ' Ошибка инициалы должны быть записаны заглавными буквами';

**end**

**else begin**

**if** (s1[2] <> '.') **and** (s1[4] <> '.')

**then begin**

err := true;

s1 := s1 + ' Ошибка второй и четвертый символ должны быть точками';

**end**;

**end**;

**end**;

**procedure** check\_prof(**var** s1: string; **var** err: boolean);

**var**

i: integer;

**begin**

err := false;

**if not** (s1[1] **in** ['A'..'Z'])

**then begin**

err := true;

s1 := s1 + ' - Ошибка в первой букве професии';

**end**

**else begin**

i:=2;

**while** (i<=length(s1))**and** (s1[i]<>' ') **and** (err=false) **do begin**

**if not** (s1[i] **in** ['a'..'z', 'A'..'Z', '-'])

**then begin**

err := true;//проверка на запрещенные символы

s1 := s1 + ' - в поле професии запрещенный символ:' + s1[i] + ', номер символа:' + i;

**end**;

inc(i);

**end**;

**while** (i<=length(s1))**and** (err=false) **do begin**

**if** s1[i]<>' '

**then begin**

err := true;//проверка на запрещенные символы

s1 := s1 + ' - в конце поля професии запрещенный символ:' + s1[i] + ', номер символа:' + i;

**end**;

inc(i);

**end**;

**end**;

**end**;

**procedure** check\_pol(**var** s1: string; **var** err: boolean);

**begin**

err := false;

**if not** (s1 = 'M') **and not** (s1 = 'F')

**then begin**

err := true;

s1 := s1 + ' - Пол указан некоректно';

**end**;

**end**;

**procedure** check\_DR(**var** s1 : string; **var** err : boolean);

**var**

d, m, y : integer; //d - day, m - month, y - year

err\_d, err\_m, err\_y : integer;

**begin**

err:= false;

val(copy(s1,1,2), d, err\_d); //переводим день в число

val(copy(s1,4,2), m, err\_m); //переводим месяц в число

val(copy(s1,7,4), y, err\_y); //переводим год в число

**if** (err\_d <> 0) **then begin** err := true;

s1 := s1 + ' - В поле "Дата рождения"(день) замечен(-ы) недопустимый(-ые) символ(-ы)';

**end**

**else if** (err\_m <> 0) **then begin** err := true;

s1 := s1 + ' - В поле "Дата рождения"(месяц) замечен(-ы) недопустимый(-ые) символ(-ы)';

**end**

**else if** (err\_y <> 0) **then begin** err := true;

s1 := s1 + ' - В поле "Дата рождения"(год) замечен(-ы) недопустимый(-ые) символ(-ы)';

**end**

**else if** (y < 1960) **or** (y > 2000) **then begin** err := true;

s1 := s1 + ' - В поле «Дата рождения» в значении год встречено недопустимое значение';

**end**

**else if** (m < 1) **or** (m > 12) **then begin** err := true;

s1 := s1 + ' - В поле «Дата рождения» в значении месяц встречено недопустимое значение';

**end**

**else case** m **of**

2: **begin**

**if** (y **mod** 4 = 0) **or** (y **mod** 100 <> 0) **and** (y **mod** 400 = 0)

**then if** (d >= 1) **and** (d <= 29) **then** err := false

**else begin** err := true;

s1 := s1 + ' - В поле «Дата рождения» в значении день встречено недопустимое значение';

**end**

**else if** (d >= 1) **and** (d <= 28) **then** err := false

**else begin** err := true;

s1 := s1 + ' - В поле «Дата рождения» в значении день встречено недопустимое значение';

**end**;

**end**;

4, 6, 9, 11: **if** (d >= 1) **and** (d <= 30) **then** err := false

**else begin** err := true;

s1 := s1 + ' - В поле «Дата рождения» в значении день встречено недопустимое значение';

**end**;

**else if** (d >= 1) **and** (d <= 31) **then** err := false

**else begin** err := true;

s1 := s1 + ' - В поле «Дата рождения» в значении день встречено недопустимое значение';

**end**;

**end**;

**end**;

**procedure** check\_DPA(**var** s1 : string; **var** err : boolean);

**var**

d, m, y : integer; //d - day, m - month, y - year

err\_d, err\_m, err\_y : integer;

**begin**

err:= false;

val(copy(s1,1,2), d, err\_d); //переводим день в число

val(copy(s1,4,2), m, err\_m); //переводим месяц в число

val(copy(s1,7,4), y, err\_y); //переводим год в число

**if** (err\_d <> 0) **then begin** err := true;

s1 := s1 + ' - В поле "Дата последней аттестации"(день) замечен(-ы) недопустимый(-ые) символ(-ы)';

**end**

**else if** (err\_m <> 0) **then begin** err := true;

s1 := s1 + ' - В поле "Дата последней аттестации"(месяц) замечен(-ы) недопустимый(-ые) символ(-ы)';

**end**

**else if** (err\_y <> 0) **then begin** err := true;

s1 := s1 + ' - В поле "Дата последней аттестации"(год) замечен(-ы) недопустимый(-ые) символ(-ы)';

**end**

**else if** (y < 1982) **or** (y > 2022) **then begin** err := true;

s1 := s1 + ' - В поле «Дата последней аттестации» в значении год встречено недопустимое значение';

**end**

**else if** (m < 1) **or** (m > 12) **then begin** err := true;

s1 := s1 + ' - В поле «Дата последней аттестации» в значении месяц встречено недопустимое значение';

**end**

**else case** m **of**

2: **begin**

**if** (y **mod** 4 = 0) **or** (y **mod** 100 <> 0) **and** (y **mod** 400 = 0)

**then if** (d >= 1) **and** (d <= 29) **then** err := false

**else begin** err := true;

s1 := s1 + ' - В поле «Дата последней аттестации» в значении день встречено недопустимое значение';

**end**

**else if** (d >= 1) **and** (d <= 28) **then** err := false

**else begin** err := true;

s1 := s1 + ' - В поле «Дата последней аттестации» в значении день встречено недопустимое значение';

**end**;

**end**;

4, 6, 9, 11: **if** (d >= 1) **and** (d <= 30) **then** err := false

**else begin** err := true;

s1 := s1 + ' - В поле «Дата последней аттестации» в значении день встречено недопустимое значение';

**end**;

**else if** (d >= 1) **and** (d <= 31) **then** err := false

**else begin** err := true;

s1 := s1 + ' - В поле «Дата последней аттестации» в значении день встречено недопустимое значение';

**end**;

**end**;

**end**;

**procedure** sort(**var** arr\_Sotr\_true:arr\_Sotr; kol\_true:integer);

**var** c: Sotr;

i, j:integer;

**begin**

**for** i := kol\_true **downto** 2 **do**

**for** j := 1 **to** kol\_true-1 **do begin**

**if** arr\_Sotr\_true[j].Fam > arr\_Sotr\_true[j+1].Fam **then begin**

c := arr\_Sotr\_true[j];

arr\_Sotr\_true[j] := arr\_Sotr\_true[j+1];

arr\_Sotr\_true[j+1] := c;

**end**;

**end**;

**end**;

**procedure** check\_prob(**var** s1: string; **var** err: boolean);

**begin**

err := false;

**if not** (s1 = ' ')

**then begin**

err := true;

s1 := s1 + ' Между словами нет пробела';

**end**;

**end**;

**end**.

Блок ourtype

**unit** ourtype;

**interface**

**const** N = 10;

**type** data = **record**

dd: string[2]; //31

mm: string[2]; //12

yyyy: string[4]; //1960-2001

**end**;

**type** Sotr = **record**

Fam: string[20]; //фамилия

i: string[4]; //инициалы

pol: string[1]; //пол

prof: string[20]; //профессия

DR: data; //дата рождения

DPA: data; //Дата атестации

**end**;

arr\_Sotr = **array**[1..N] **of** Sotr;

**implementation**

**end**.

Блок lr1\_1

**program** lr1\_1;

**uses** ourtype,ourprocedures;

**var**

f1, f2, f3: text;

s, s1: string;

k, num, a: integer;

kol\_true:integer;

rec\_sotr\_true: Sotr;

arr\_sotr\_true:arr\_Sotr;

err: boolean;

**begin**

assign(f1, 'in.txt'); reset(f1);

assign(f2, 'outTrue.txt'); rewrite(f2);

assign(f3, 'outFalse.txt'); rewrite(f3);

a := 0;

kol\_true:=0;

**while not** eof(f1) **and** (a<=n) **do**

**begin** //начинаем работу со всем файлом

readln(f1, s);

a:=a+1;

//проверяем длину строки

**if** length(s)=70 **then begin**

num := 0;

//проверка фамилии

s1 := copy(s, 1, 20);

check\_fam(s1, err);

**if** err = false

**then begin** num:=num+1;

rec\_sotr\_true.Fam := s1;

**end**

**else** writeln(f3, 'Ошибка в строке ', a, ' = ', s1);

//проверка инициалов

s1 := copy(s, 22, 4);

check\_inic(s1, err);

**if** err = false

**then begin** num:=num+1;

rec\_sotr\_true.i := s1;

**end**

**else** writeln(f3, 'Ошибка в строке ', a, ' = ', s1);

//проверка пола

s1 := copy(s, 27, 1);

check\_pol(s1, err);

**if** err = false

**then begin** num:=num+1;

rec\_sotr\_true.pol:=s1;

**end**

**else** writeln(f3, 'Ошибка в строке ', a, ' = ', s1);

//проверка профессии

s1 := copy(s, 29, 20);

check\_prof(s1, err);

**if** err = false

**then begin** num:=num+1;

rec\_sotr\_true.prof:=s1;

**end**

**else** writeln(f3, 'Ошибка в строке ', a, ' = ', s1);

//проверка даты рождения

s1:=copy(s, 50, 10);

check\_DR(s1, err);

**if** err = false

**then begin** num:=num+1;

rec\_sotr\_true.DR.dd:=copy(s1,1,2);

rec\_sotr\_true.DR.mm:=copy(s1,4,2);

rec\_sotr\_true.DR.yyyy:=copy(s1,7,4);

**end**

**else** writeln(f3, 'Ошибка в строке ', a, ' = ', s1);

//проверка даты последней аттестации

s1:=copy(s, 61, 10);

check\_DPA(s1, err);

**if** err = false

**then begin** num:=num+1;

rec\_sotr\_true.DPA.dd:=copy(s1,1,2);

rec\_sotr\_true.DPA.mm:=copy(s1,4,2);

rec\_sotr\_true.DPA.yyyy:=copy(s1,7,4);

**end**

**else** writeln(f3, 'Ошибка в строке ', a, ' = ', s1);

//проверка наличия пробелов

s1 := copy(s, 21, 1);

check\_prob(s1, err);

**if** err = false

**then begin** num:=num+1;

**end**

**else** writeln(f3, 'Ошибка в строке ', a, ' = ', s1);

s1 := copy(s, 26, 1);

check\_prob(s1, err);

**if** err = false

**then begin** num:=num+1;

**end**

**else** writeln(f3, 'Ошибка в строке ', a, ' = ', s1);

s1 := copy(s, 28, 1);

check\_prob(s1, err);

**if** err = false

**then begin** num:=num+1;

**end**

**else** writeln(f3, 'Ошибка в строке ', a, ' = ', s1);

s1 := copy(s, 49, 1);

check\_prob(s1, err);

**if** err = false

**then begin** num:=num+1;

**end**

**else** writeln(f3, 'Ошибка в строке ', a, ' = ', s1);

s1 := copy(s, 60, 1);

check\_prob(s1, err);

**if** err = false

**then begin** num:=num+1;

**end**

**else** writeln(f3, 'Ошибка в строке ', a, ' = ', s1);

//конец цикла работы со строкой

**if** (num = 11)

**then begin**

kol\_true:=kol\_true+1;

writeln(kol\_true:5,a:5);

arr\_Sotr\_true[kol\_true]:=rec\_Sotr\_true;

**end**;

**end**

**else** writeln(f3, 'Ошибка в строке ', a, ' = неверный формат', length(s), ' ', s);

**end**;

**if not** eof(f1) **then** writeln(f3,'Кол-во строк не соответсвует значению');

sort(arr\_Sotr\_true, kol\_true);

**for** k:=1 **to** kol\_true **do**

**with** arr\_sotr\_true[k] **do**

writeln(f2, Fam:20,' ', Prof:20,' ', DPA.dd, '.', DPA.mm, '.', DPA.yyyy);

//writeln(f2, Fam:20, ' ', i:4, ' ', Pol:1, ' ', Prof:20, ' ', DR:10, ' ', DPA:10);}

close(f1);

close(f2);

close(f3);

**end**.