МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
**Дальневосточный федеральный университет**

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**Кафедра прикладной математики, механики, управления программного обеспечения.**

**ОТЧЕТ**о прохождении учебной практики  
(практика по получению первичных профессиональных умений и навыков, в том числе научно-исследовательской деятельности)

Выполнил студент гр. Б8104  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отчёт защищён с оценкой Руководители практики:  
 к.т.н., доцент,  
 доцент кафедры ПММУ и ПО.  
\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.И. Антонова   
 (подпись) (И.О.Фамилия) (подпись) (И.О.Фамилия) к.т.н., доцент кафедры ПММУ и ПО  
 С.Н. Остроухова   
 (подпись) (И.О.Фамилия)«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019г. Практика пройдена в срок  
Регистрационный №\_\_\_\_\_\_\_\_\_\_\_\_ с «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019г.  
«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019г. по«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019г.  
\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 (подпись) (И.О.Фамилия)

г. Владивосток  
2019

Оглавление

[ЗАДАЧА “УЧЕТ СОТРУДНИКОВ” 3](#__RefHeading___Toc6091_663852789)

[АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ “УЧЕТ СОТРУДНИКОВ” 4](#__RefHeading___Toc5559_663852789)

[ТРЕБОВАНИЯ К КП “УЧЕТ СОТРУДНИКОВ” 9](#__RefHeading___Toc5561_663852789)

[СПЕЦИФИКАЦИЯ ПРОГРАММЫ “УЧЕТ СОТРУДНИКОВ” 10](#__RefHeading___Toc5563_663852789)

[ПРОЕКТ КП “УЧЕТ СОТРУДНИКОВ” 25](#__RefHeading___Toc5673_663852789)

[ТЕСТИРОВАНИЕ ПРОГРАММЫ “УЧЕТ СОТРУДНИКОВ” 32](#__RefHeading___Toc5567_663852789)

# ЗАДАЧА “УЧЕТ СОТРУДНИКОВ”

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

* фамилию и инициалы,
* пол,
* профессию,
* дату рождения,
* дату последней аттестации

Так же в отделе кадров хранится справочник, информация о том, как часто сотрудники должны проходить аттестацию:

* профессия
* периодичность аттестации

*Требуется разработать для аттестационной комиссии программу для формирования списков сотрудников, которые не прошли аттестации до указанной даты. Человек подлежит аттестацции, если в день заседания аттестационной комиссии разница между датой заседания и датой последней аттестации равна периодичности аттестации с точностью до месяца. Полученный список должен быть отсортирован по профессии и включать в себя фамилиии и инициалы сотрудников и дату последней аттестации.*

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ “УЧЕТ СОТРУДНИКОВ”

1. **Построение ПО “УЧЕТ СОТРУДНИКОВ”**
   1. Выделение профессионалов предметной области
   * Активные: аттестационная комиссия
   * Пассивные: список сотрудников предприятия

Аттестационная комиссия может запросить список сотрудников, которые не прошли аттестацию до указанной даты.

* 1. Название понятий и объектов ПО “УЧЕТ СОТРУДНИКОВ”

Понятие “Строка” – конечная последовательность символов какого-либо алфавита.

Понятие “Дата” – последовательность цифр, имеющая вид:

D/M/Y, где

D ∈ [01,02..31],

M ∈ [01,02..12],

Y ∈ [1950..2017],

Связь между Y, M и D задается следующей таблицей:

|  |  |  |
| --- | --- | --- |
| **Y** | **M** | **D** |
| [1950..2017] | 01,03,05,07,08,10,12 | [01..31] |
| 04,06,09,11 | [01..30] |
| Y делится нацело на 4 | 02 | [01.29] |
| Y не делится нацело на 4 | [01..28] |

Понятие “Файл” – именованная область данных на носителе информации, выглядящая как последовательность строк следующего вида:

s11\*s12\*...\*s1k\n

s21\*s22\*...\*s2f\n

…………….

sd1\*sd2\*...\* sdp, где:

\* - конечная (ненулевая) последовательность символов пробела, \n – символ переноса строки,

k, f, d, p – натуральные числа, sij – объект понятия “Строка”,

число i называется строкой файла, строки sj1, sj2 ... sjk называются полями j строки файла.

“Фамилия и инициалы” – объект понятия “Строка”, имеющая следующий вид:

#\*.#.# , где # - заглавная буква латинского алфавита, \* - строка произвольной (возможно нулевой) длины, состоящая из строчных букв латинского алфавита.

Примеры валидных строк: “Smith.J.G”, “Alse.W.B”

Примеры невалидных строк: “Al.d.R”, “1.223sdaa”, “Str.A”

“Пол” - один из символов {M, F}, где M обозначает мужской пол, F – женский пол.

“Профессия” - объект понятия “Строка”, имеющая следующий вид:

#\* , где # - заглавная буква латинского алфавита, \* - строка произвольной (возможно нулевой) длины, состоящая из строчных букв латинского алфавита.

Примеры валидных строк: “Doctor”, “Worker”

Примеры невалидных строк: “DoCTOr”, ”doctor ”

“Дата рождения” - объект понятия “Дата”, в которой

1950 <= Y <= 2000

Примеры валидных дат: “01/12/1999”, “12/12/2000”

Примеры невалидных дат: “13/12/2010”, “30/11/4444”

“Дата аттестации” - объект понятия “Дата”, в которой

2000 <= Y <= 2017.

Примеры валидных дат: “13/10/2010”, “12/12/2005”

Примеры невалидных дат: “01/12/1999”, “30/11/4444”

“Периодичность аттестации” - натуральное число, лежащее в диапазоне [1..3], обозначающее периодичность аттестации в годах.

“Сотрудник” – объект, являющийся совокупностью атрибутов “Фамилия и инициалы”, “Пол”, “Профессия”, “Дата рождения”, “Дата аттестации”, каждый из атрибутов разделен ненулевым количеством пробелов.

Пример: **Сотрудник\_i**= Smit.J.R\*\*\*M\*\*Doctor\*\*02/05/1975\*18/06/2013,

\* - символ пробела

“Staff” - объект понятия “Файл”, имеющий вид:

**Сотрудник\_1**\n

**Сотрудник\_2**\n

…….

**Сотрудник\_i,**

I – натуральное число.

Пример:

Sith.J.R M Doctor 02/06/1988 20/11/2015

Wale.K.A F Driver 12/10/1973 11/04/2014

“Catalog” - объект понятия “Файл”, имеющий вид:

**Периодичность\_аттестации\_1**\***Профессия\_1**\n

**Периодичность\_аттестации\_2\*Профессия\_2**\n

…….

**Периодичность\_аттестации\_i\*Профессия\_i**,

\* - один символ пробела, I – натуральное число.

Пример:

2 Driver

1 Worker

1 Doctor

“Output” - объект понятия “Файл”, имеющий вид:

Фамилия\_и\_инициалы**\_1\*Профессия\*Дата**\n

**Сотрудник\_2**\n

…….

**Сотрудник\_k**, в котором объекты

“Сотрудник” сортированы по атрибуту “Профессия”, согласно закону сортировки объектов понятий строка (номер 3),

* 1. Отношения между объектами:

Между объектами понятий “Строка” существует бинарное отношение **равно (=)**:

Объекты понятия “Строка” называются равными, если равны соответствующие символы, стоящие в записи данных объектов.

Если отношение **равно** не выполняется, то выполняется отношение **не равно**

**(!=).**

Пример: “1293” = “1293”, “1293” != “12 93”

Между объектами “Дата” существует бинарное отношение **равно (=)**:

**Дата\_1** = **Дата\_2**, если D1 = D2, M1 = M2, Y1 = Y2.

Если отношение **равно** не выполняется, то выполняется отношение **не равно**

**(!=).**

Между объектами “Дата” существует бинарное отношение **меньше (<)**:

**Дата\_1** < **Дата\_2**, если истинно следующее логическое выражение:

[ (Y1 < Y2) или

((Y1 = Y2) и (M1 < M2)) или

((Y1 = Y2) и (M1 = M2) и (D1 < D2)) ].

Пример: “03/04/1988” < “01/05/2010”,

“04/04/2015” < ”04/11/2015”, “17/03/1965” < “18/03/1965”

Между объектами “Дата”, для которых выполняется

**Дата\_2** < **Дата\_1**, существует бинарная операция **разность (-):**

1) **Дата\_1** – **Дата\_2** = Y1 – Y2, если истинно следующее логическое выражение:

[ (M1 > M2) или

((M1 = M2) и (D1 >= D2)) ].

2) **Дата\_1** – **Дата\_2** = Y1 – Y2 – 1, если истинно следующее логическое выражение:

[ (M1 < M2) или

((M1 = M2) и (D1 < D2)) ].

Разность дат показывает, сколько полных лет прошло между двумя датами.

Пример: “13/04/2018” - “14/04/2000” = 17,

“13/04/2018” - “04/05/2000” = 17,

“13/05/2018” - “20/04/2000” = 18,

“13/05/2018” - “13/05/2000” = 18,

* 1. Законы ПО

1) Для каждого объекта “Сотрудник” разность атрибутов “Дата аттестации” и “Дата рождения” должна быть больше 17.

2) В файле “Catalog” для каждого объекта “Профессия” объект “Периодичность аттестации” уникальна.

Пример некорректного файла:

2 Doctor

3 Doctor

3) Объекты понятия “Строка” могут быть сортированы по латинскому алфавиту, в котором A<B<...<Y<Z<a<b<...<y<z. Отсутствующий символ считается меньше любого другого символа. Объект **Строка\_1** будет встречаться ранее объекта **Строка\_2**, если выполняется одно из двух условий:

1) Первый элемент объекта **Строка\_1** < первого элемента **Строка\_2**.

2) Если любой элемент объекта **Строка\_1** вплоть до N-ого = соответствующему элементу объекта **Строка\_2**, а N+1 элемент объекта **Строка\_1** < N+1 элемента объекта **Строка\_2**.

4) В качестве входных данных, помимо файлов “Staff” и “Catalog”, программа получает на вход 3 целых числа, введенных с клавиатуры, обозначающих число, месяц и год.

5) Сотрудник, содержащийся в файле “Staff” попадает в файл “Output”, если разность между введенной датой и датой последней аттестации данного сотрудника >= периодичности аттестации данного сотрудника.

# ТРЕБОВАНИЯ К КП “УЧЕТ СОТРУДНИКОВ”

1. Функциональные требования к КП
   1. КП должна решать задачу о формировании списка сотрудников, непрошедших аттестацию для любой введенной даты, большей либо равной 01/01/2016 и меньшей либо равной 31/12/2017
   2. КП должна брать исходные данные о сотрудниках из файлов “Staff” и “Catalog”.
   3. КП должна обеспечивать ввод дня, месяца и года с клавиатуры в формате целых чисел.
   4. КП должна формировать файл “Output”, в который она записывает информацию о сотрудниках непрошедших аттестацию, а именно их фамилию и инициалы, профессию и дату последней аттестации.
   5. КП должна выводить на экран сообщения об ошибках в файлах “Staff” и “Catalog” и введенной дате.
   6. КП должна обрабатывать только первые 100 корректных записей в файле “Staff” и “Catalog”.
2. Требования к удобству КП
   1. КП должна выдавать все сообщения на экран на английском языке.
   2. КП должна выдавать подсказки пользователю при вводе исходных дня, месяца и года.
3. Требования к надежности КП
   1. В ходе работы КП значения входных данных не должны быть изменены.
4. Требования к мобильности КП
   1. КП должна быть переносимой без изменений из одной среды в другую в рамках ОС семейства UNIX, Windows 7, Windows 8, Windows 10.

# СПЕЦИФИКАЦИЯ ПРОГРАММЫ “УЧЕТ СОТРУДНИКОВ”

Список ошибок и сообщений КП:

**Сообщения**

* При вводе дня, мясяца и года:
  1. Enter the year [2016..2017]
  2. Enter the month [1..12]
  3. Enter the day [1..31]
  4. Enter the day [1..30]
  5. Enter the day [1..29]
  6. Enter the day [1..28]
  7. All staff passed the certification
  8. File “Staff” has no correct information
  9. File “Catalog” has no correct information
  10. File “Staff” is empty
  11. File “Catalog” is empty
  12. Fatal error occurred during parsing Catalog, terminating
  13. Error has occurred during opening file Staff.txt
  14. Error has occurred during opening file Catalog.txt
  15. Checking file Catalog
  16. Checking file Staff

**Ошибки**

* При вводе дня, мясяца и года:

|  |  |
| --- | --- |
| Название ошибки | Сообщение ошибки |
| INVALID\_GIVEN\_YEAR | Year must be in range [2016..2017] |
| INVALID\_GIVEN\_MONTH | Month must be in range [1..12] |
| INVALID\_GIVEN\_DAY\_31 | Day must be in range [1..31] |
| INVALID\_GIVEN\_DAY\_30 | Day must be in range [1..30] |
| INVALID\_GIVEN\_DAY\_29 | Day must be in range [1..29] |
| INVALID\_GIVEN\_DAY\_28 | Day must be in range [1..28] |

* При проверке файла “Staff”, I – номер строки файла, F – ошибочное поле:

|  |  |
| --- | --- |
| Название ошибки | Сообщение ошибки |
| STAFF\_INVALID\_NAME | (I) ERROR: name (F) must consist of surname, that starts with capital letter and followed by small latin letters and initials, that are capital latin letters separated by dots, for example Surname.S.S |
| STAFF\_INVALID\_GENDER | (I) ERROR: gender (F) must be either F or M |
| STAFF\_INVALID\_PROFESSION | (I) ERROR: profession (F) must start with capital letter and other symbols must be only small latin letters. |
| STAFF\_INVALID\_BIRTH\_DATE\_YEAR | (I) ERROR: birth date year (F) must be in range [1950..2000] |
| STAFF\_INVALID\_BIRTH\_DATE\_MONTH | (I) ERROR: birth date month (F) must be in range [01..12] |
| STAFF\_INVALID\_BIRTH\_DATE\_DAY\_31 | (I) ERROR: birth date day (F) must be in range [01..31] |
| STAFF\_INVALID\_BIRTH\_DATE\_DAY\_30 | (I) ERROR: birth date day (F) must be in range [01..30] |
| STAFF\_INVALID\_BIRTH\_DATE\_DAY\_29 | (I) ERROR: birth date day (F) must be in range [01..29] |
| STAFF\_INVALID\_BIRTH\_DATE\_DAY\_28 | (I) ERROR: birth date day (F) must be in range [01..28] |
| STAFF\_INVALID\_BIRTH\_DATE\_SEPARATOR | (I) ERROR: day, month and year in birth date (F) should be separated with symbol ‘/’ |
| STAFF\_INVALID\_BIRTH\_DATE\_LENGTH | (I) ERROR: Birth date (F) must be 10 symbols long |
| STAFF\_INVALID\_CERTIFICATION\_DATE\_YEAR | (I) ERROR: certification date year (F) must be in range [2000..2017] |
| STAFF\_INVALID\_CERTIFICATION\_DATE\_MONTH | (I) ERROR: certification date month (F) must be in range [01..12] |
| STAFF\_INVALID\_CERTIFICATION\_DATE\_DAY\_31 | (I) ERROR: certification date day (F) must be in range [01..31] |
| STAFF\_INVALID\_CERTIFICATION\_DATE\_DAY\_30 | (I) ERROR: certification date day (F) must be in range [01..30] |
| STAFF\_INVALID\_CERTIFICATION\_DATE\_DAY\_29 | (I) ERROR: certification date day (F) must be in range [01..29] |
| STAFF\_INVALID\_CERTIFICATION\_DATE\_DAY\_28 | (I) ERROR: certification date day (F) must be in range [01..28] |
| STAFF\_INVALID\_CERTIFICATION\_DATE\_SEPARATOR | (I) ERROR: day, month and year in certification date (F) should be separated with symbol ‘/’ |
| STAFF\_INVALID\_CERTIFICATION\_DATE\_LENGTH | (I) ERROR: Certification date (F) must be 10 symbols long |
| STAFF\_INVALID\_DIFFERENCE\_BETWEEN\_BIRTH\_AND\_CERTIFICATION | (I) ERROR: difference between birth date (F) and certification date (F) must be greater than 17 years. |
| STAFF\_NOT\_ENOUGH\_ATTRIBUTES | (I) ERROR: not enough attributes, 5 required. |
| STAFF\_PERIOD\_NOT\_FOUND | (I) ERROR: there was no information about certification period for profession (F) in file “Catalog” . |
| STAFF\_WARNING\_MORE\_ATTRIBUTES | (I) WARNING: line is correct, but there are more than 5 attributes. |
| STAFF\_WARNING\_OVER\_HUNDRED\_CORRECT\_LINES | WARNING: file “Staff” has more than 100 correct lines, data loose is possible. |

При проверке файла “Catalog”, I– номер строки файла, F – ошибочное поле.

|  |  |
| --- | --- |
| Название ошибки | Сообщение ошибки |
| CATALOG\_INVALID\_PROFESSION | (I) ERROR: profession (F) must start with capital letter and other symbols must be only small latin letters. |
| CATALOG\_INVALID\_CERTIFICATION\_PERIOD | (I) ERROR: certification period (F) must be natural number in range [1..3] |
| CATALOG\_INVALID\_STRUCTURE | (I) ERROR: second symbol must be space character |
| CATALOG\_INVALID\_NUMBER\_OF\_SPACE | (I) ERROR: there must be only one space character between certification period and profession |
| CATALOG\_EQUAL\_PROFESSION\_WITH\_DIFFERENT\_PERIOD | (I) FATAL: profession (F) is repeating with different period |
| CATALOG\_EQUAL\_PROFESSION\_WITH\_SAME\_PERIOD | (I) WARNING: profession (F) is repeating with same period |
| CATALOG\_NOT\_ENOUGH\_ATTRIBUTES | (I) ERROR: not enough attributes, 2 required. |
| CATALOG\_WARNING\_OVER\_HUNDRED\_CORRECT\_LINES | WARNING: file “Catalog” has more than 100 correct lines, data loose is possible. |

1. **Спецификация задачи**

**Формальная постановка задачи**

X = {D = [1,2..31] – день, M = [1,2..12] – месяц, Y = [2016..2017] – год. файл “Staff”; файл “Catalog”}.

Y = {файл “Output”, сообщения и ошибки}

F = {

* + - Если в I строке файла “Staff” имя неверно, то выводится ошибка STAFF\_INVALID\_NAME
    - Если в I строке файла “Staff” пол не является символом F или M, то выводится ошибка STAFF\_INVALID\_GENDER
    - Если в I строке файла “Staff” профессия неверна, то выводится ошибка STAFF\_INVALID\_PROFESSION
    - Если в I строке файла “Staff” год у даты рождения меньше 1950 или больше 2000, то выводится ошибка STAFF\_INVALID\_BIRTH\_DATE\_YEAR
    - Если в I строке файла “Staff” месяц даты рождения меньше 1 или больше 12, то выводится ошибка STAFF\_INVALID\_BIRTH\_DATE\_MONTH
    - Если в I строке файла “Staff” месяц даты рождения равняется 1,3,5,7,8,10 или 12, а день даты рождения меньше 1 или больше 31, то выводится ошибка STAFF\_INVALID\_BIRTH\_DATE\_DAY\_31
    - Если в I строке файла “Staff” месяц даты рождения равняется 4,6,9, или 11, а день даты рождения меньше 1 или больше 30, то выводится ошибка STAFF\_INVALID\_BIRTH\_DATE\_DAY\_30
    - Если в I строке файла “Staff” месяц даты рождения равняется 2 и год даты рождения делится на 4, а день даты рождения меньше 1 или больше 29, то выводится ошибка STAFF\_INVALID\_BIRTH\_DATE\_DAY\_29
    - Если в I строке файла “Staff” месяц даты рождения равняется 2 и год даты рождения не делится на 4, а день даты рождения меньше 1 или больше 28, то выводится ошибка STAFF\_INVALID\_BIRTH\_DATE\_DAY\_28
    - Если в I строке файла “Staff” между днем, месяцем и годом даты рождения стоят символы, отличные от символа ‘/’, то выводится ошибка STAFF\_INVALID\_BIRTH\_DATE\_SEPARATOR
    - Если в I строке файла “Staff” количество символов в дате рождения не равно 10, то выводится ошибка STAFF\_INVALID\_BIRTH\_DATE\_LENGTH
    - Если в I строке файла “Staff” год у даты аттестации меньше 2000 или больше 2017, то выводится ошибка STAFF\_INVALID\_CERTIFICATION\_DATE\_YEAR
    - Если в I строке файла “Staff” месяц даты даты аттестации меньше 1 или больше 12, то выводится ошибка STAFF\_INVALID\_CERTIFICATION\_DATE\_MONTH
    - Если в I строке файла “Staff” месяц даты аттестации равняется 1,3,5,7,8,10 или 12, а день даты аттестации меньше 1 или больше 31, то выводится ошибка STAFF\_INVALID\_CERTIFICATION\_DATE\_DAY\_31
    - Если в I строке файла “Staff” месяц даты аттестации равняется 4,6,9, или 11, а день даты аттестации меньше 1 или больше 30, то выводится ошибка STAFF\_INVALID\_CERTIFICATION\_DATE\_DAY\_30
    - Если в I строке файла “Staff” месяц даты аттестации равняется 2 и год даты аттестации делится на 4, а день даты аттестации меньше 1 или больше 29, то выводится ошибка STAFF\_INVALID\_BIRTH\_DATE\_DAY\_29
    - Если в I строке файла “Staff” месяц даты аттестации равняется 2 и год даты аттестации не делится на 4, а день даты аттестации меньше 1 или больше 28, то выводится ошибка STAFF\_INVALID\_BIRTH\_DATE\_DAY\_28
    - Если в I строке файла “Staff” между днем, месяцем и годом даты аттестации стоят символы, отличные от символа ‘/’, то выводится ошибка STAFF\_INVALID\_CERTIFICATION\_DATE\_SEPARATOR
    - Если в I строке файла “Staff” количество символов в дате аттестации не равно 10, то выводится ошибка STAFF\_INVALID\_CERTIFICATION\_DATE\_LENGTH
    - Если в I строке файла “Staff” разность между датой рождения и датой аттестации меньше либо равна 17, то выводится ошибка STAFF\_INVALID\_DIFFERENCE\_BETWEEN\_BIRTH\_AND\_CERTIFICATION
    - Если в I строке файла “Staff” количество полей меньше 5, то выводится ошибка STAFF\_NOT\_ENOUGH\_ATTRIBUTES
    - Если в I строке файла “Staff” встретилась профессия, для которой не нашлась периодичность аттестации в файле “Catalog”, то выводится ошибка STAFF\_PERIOD\_NOT\_FOUND
    - Если в I строке файла “Staff” первые 5 полей оказались правильными, но в этой строке полей больше, чем 5, то выводится предупреждение STAFF\_WARNING\_MORE\_ATTRIBUTES
    - Если в файле “Staff” имеется ,более 100 правильных строк, то выводится предупреждение STAFF\_WARNING\_OVER\_HUNDRED\_CORRECT\_LINES
    - Если в I строке файла “Catalog” профессия неверна, то выводится ошибка CATALOG\_INVALID\_PROFESSION
    - Если в I строке файла “Catalog” периодичность аттестации меньше 1 или больше 3, то выводится ошибка CATALOG\_INVALID\_CERTIFICATION\_PERIOD
    - Если в I строке файла “Catalog” второй символ не является символом пробела, то выводится ошибка CATALOG\_INVALID\_STRUCTURE
    - Если в I строке файла “Catalog” между периодичностью аттестации и профессией количество символов пробела больше 1, то выводится ошибка CATALOG\_INVALID\_NUMBER\_OF\_SPACE
    - Если в I строке файла “Catalog” встретилась профессия, которая уже ранее встречалась, при этом с другим периодом аттестации, то выводится ошибка CATALOG\_EQUAL\_PROFESSION\_WITH\_DIFFERENT\_PERIOD
    - Если в I строке файла “Catalog” встретилась профессия, которая уже ранее встречалась, при этом с тем же периодом аттестации, то выводится ошибка CATALOG\_EQUAL\_PROFESSION\_WITH\_SAME\_PERIOD
    - Если в I строке файла “Catalog” количество полей меньше 2, то выводится ошибка CATALOG\_NOT\_ENOUGH\_ATTRIBUTES
    - Если в файле “Catalog” имеется более 100 правильных строк, то выводится предупреждение CATALOG\_WARNING\_OVER\_HUNDRED\_CORRECT\_LINES
    - Если после работы КП файл “Output” оказался пуст, то на экран выводится сообщение “All staff passed the certification”
    - Если в файле “Staff” не оказалось корректных данных, то на экран выводится сообщение “File “Staff” has no correct information”
    - Если в файле “Catalog” не оказалось корректных данных, то на экран выводится сообщение “File “Catalog” has no correct information”
    - Если файл “Staff” оказался пуст, то выводится сообщение “File “Staff” is empty”
    - Если файл “Catalog” оказался пуст, то выводится сообщение “File “Catalog” is empty”
    - Если в файле “Catalog” оказались одинаковые профессии с разными периодами аттестации либо в файле не оказалось верных строк, то выводится сообщение “Fatal error occurred during parsing Catalog, terminating”
    - Если файла “Staff” не существует, то выводится сообщение “Error has occurred during opening file Staff.txt”
    - Если файла “Catalog” не существует, то выводится сообщение “Error has occurred during opening file Catalog.txt”
    - Если файл “Catalog” существует и не пуст, то выводится сообщение “Checking file Catalog”
    - Если файл “Staff” существует и не пуст, а в файле “Catalog” не оказалось одинаковых профессий с разным периодом аттестации, то выводится сообщение “Checking file Staff”
    - Если как и в файле “Staff”, так и в файле “Catalog” были найдены корректные данные, а в файле “Catalog” не оказалось одинаковых профессий с разным периодом аттестации, то все сотрудники, у которых разность между введенной датой и датой последней аттестации данного сотрудника >= периодичности аттестации данного сотрудника сортируются по профессии, а их фамилия и инициалы, профессия и дата последней аттестации заносятся в файл Output

}

**Метод решения**

* 1. КП запрашивает у пользователя год, месяц и день с клавиатуры.
  2. Если данные, введенные с клавиатуры верны, то КП проверяет файлы “Staff” и “Catalog” на корректность, иначе КП выводит соответствующие сообщения об ошибках и просит ввести данные заново.
  3. КП построчно проверяет файл “Catalog” на корректность:
     1. Если файл пуст или не существует, то КП выводит соответствующее сообщение и завершает работу.
     2. Если в файле найдены одинаковые профессии с разными периодами аттестации, то КП выводит на экран соответствующее сообщение о ошибке, проверяет оставшуюся часть файла “Catalog” и завершает работу.
     3. Если каждая профессия в файле имеет уникальный период аттестации, но присутствуют другие ошибки, то КП выводит на экран соответствующие сообщения об ошибках и проверяет файл “Staff”.
     4. Если все данные корректны, то КП проверяет файл “Staff”.
     5. Если корректные данные не найдены, то КП выводит соответствующее сообщение об ошибке и завершает работу.
  4. КП построчно проверяет файл “Staff” на корректность:
     1. Если файл пуст или не существует, то КП выводит соответствующее сообщение и завершает работу
     2. Если каждая строка файла Staff корректна, то КП начинает выборку сотрудников, непрошедших аттестацию.
     3. Если найдены как корректные, так и некорректные строки в файле Staff, то КП выводит соответствующие сообщения об ошибках и начинает выборку сотрудников, непрошедших аттестацию.
     4. Если корректные данные не найдены, то КП выводит соответствующее сообщение об ошибке и завершает работу.
  5. Из файла “Catalog” КП считывает информацию о связи профессии и периодичности аттестации.
  6. КП определяет, какие сотрудники не прошли аттестацию, сортирует их по атрибуту “Профессия” и выводит каждого сотрудника в файл “Output”.

1. **Спецификация входных данных**
   1. Каждая строка файла “Staff” представлена в виде:

“Фамилия и инициалы”\*”Пол”\*”Профессия”\*”Дата рождения”\*”Дата аттестации”, где \* - ненулевое количество пробелов. Перед полем “Фамилия и инициалы” может стоять произвольное кол-во символов пробела. Например:

Arky.G.G F Worker 12/10/1978 02/03/2016

Smith.J.L M Doctor 29/08/1968 11/12/2015

Games.D.Q Director 15/02/1985 02/01/2015

“Фамилия и инициалы” – строка, имеющая следующий вид:

#\*.#.# , где # - заглавная буква латинского алфавита, \* - строка произвольной (возможно нулевой) длины, состоящая из строчных букв латинского алфавита.

Например: “Smith.J.G”, “Alse.W.B”

“Пол” - один из символов {M, F}, где M обозначает мужской пол, F – женский пол.

“Профессия” - строка, имеющая следующий вид:

#\* , где # - заглавная буква латинского алфавита, \* - строка произвольной (возможно нулевой) длины, состоящая из строчных букв латинского алфавита.

Например: “Doctor”, “Worker”

“Дата рождения” - объект понятия “Дата”, в которой

1950 <= Y <= 2000

Например “01/12/1999”, “12/12/2000”

“Дата аттестации” - объект понятия “Дата”, в которой

2000 <= Y <= 2017.

Например: “13/10/2010”, “12/12/2005”

* 1. Каждая строка файла “Catalog” представлена в виде:

“Периодичность аттестации”\*”Профессия”, где \* - ровно один символ пробела. Если в какой-либо строке файла встретилась профессия с данным периодом аттестации, то в другой строке файла эта же профессия не может встретиться с другим периодом аттестации. Например:

1 Doctor

2 Worker

1 Doctor

Пример некорректного файла:

1 Doctor

2 Worker

3 Doctor

“Периодичность аттестации” - натуральное число в диапазоне [1..3]

“Профессия” - строка, имеющая следующий вид:

#\* , где # - заглавная буква латинского алфавита, \* - строка произвольной (возможно нулевой) длины, состоящая из строчных букв латинского алфавита.

Например: “Doctor”, “Worker”

* 1. Год Y, введенный пользователем – натуральное число в диапазоне [2016..2017]
  2. Месяц M, введенный пользователем – натуральное число в диапазоне [1..12]
  3. День D, введенный пользователем – натуральное число в диапазоне [1..31]

1. **Спецификация выходных данных –**
   1. Сообщения и ошибки КП.
   2. Каждая строка файла “Output” представлена в виде:

“Фамилия и инициалы”\*”Профессия”\*”Дата аттестации”, где \* - ровно один символ пробела, а строки файла сортированы по полю “Профессия”, согласно закону ПО #3. Например:

Smith.J.L Doctor 11/12/2015

Arky.G.G Worker 02/03/2016

“Фамилия и инициалы” – строка, имеющая следующий вид:

#\*.#.# , где # - заглавная буква латинского алфавита, \* - строка произвольной (возможно нулевой) длины, состоящая из строчных букв латинского алфавита.

Например: “Smith.J.G”, “Alse.W.B”

“Профессия” - строка, имеющая следующий вид:

#\* , где # - заглавная буква латинского алфавита, \* - строка произвольной (возможно нулевой) длины, состоящая из строчных букв латинского алфавита.

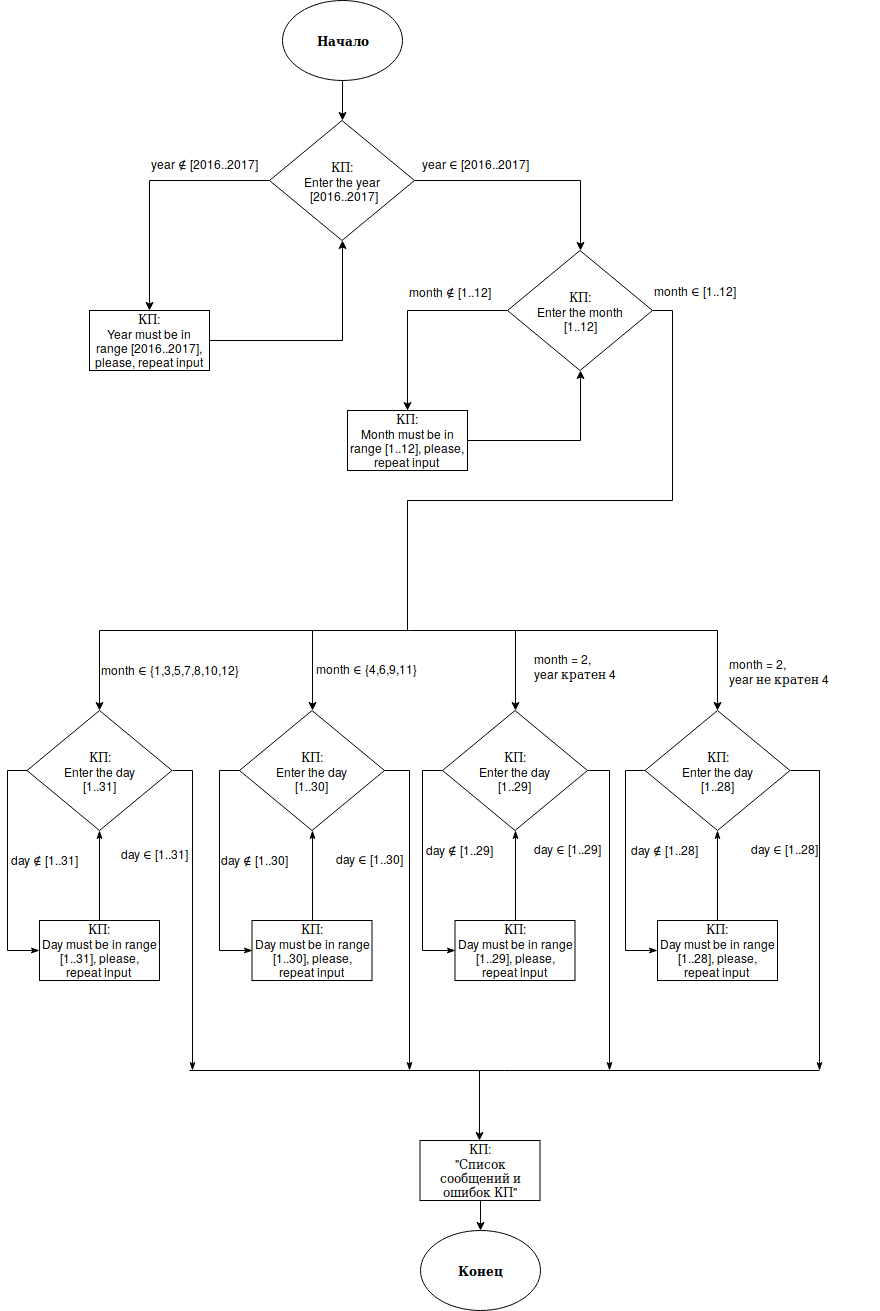
Например: “Doctor”, “Worker”

“Дата аттестации” - объект понятия “Дата”, в которой

2000 <= Y <= 2017.

Например: “13/10/2010”, “12/12/2005”

1. **Сценарий диалога с КП**

****

# ПРОЕКТ КП “УЧЕТ СОТРУДНИКОВ”

1. ПРОЕКТ СТРУКТУР ДАННЫХ

**const**

**MAX\_CORRECT\_LINES = 100;** {Максимальное количество корректных строк}

**type**

{Описание вспомогательных типов, использющихся для проверки файлов}

**Name\_States = (InitialName, Surname, SurnameDot, Name, NameDot, EndOfName);**

**Gender\_States = (InitialGender, EndOfGender);**

**Profession\_States = (InitialProfession, LowerCaseProfession, EndOfProfession);**

**Date = record** {Описание типа Дата}

**day : shortint;** {День}

**month : shortint;** {Месяц}

**year : smallint;** {Год}

**end;**

{Используем shortint и smallint для того, чтобы КП не завершалась аварийно при попытке ввести отрицательное число}

**Worker = record** {Описание типа Сотрудник}

**name : string;**

**gender : char;**

**profession : string;**

**birth : Date;**

**certification : Date;**

**end;**

**Info = record** {Описание выходной информации}

**name : string;**

**profession : string;**

**certification : Date;**

**end;**

**Qualification = record** {Описание типа, в котором хранится связь между профессией и периодичностью аттестации}

**period : byte;**

**profession : string;**

**end;**

**TableOfWorkers = array [0..MAX\_CORRECT\_LINES-1] of Worker;**

**TableOfQualification = array [0..MAX\_CORRECT\_LINES-1] of Qualification;**

**InfoTable = array [0..MAX\_CORRECT\_LINES-1] of Info;**

**var**

**staff : text;** {Файл Staff}

**catalog : text;** {Файл Catalog}

**output\_file : text;** {Файл Output}

**input\_date : Date;** {Входная дата}

**table\_of\_workers : TableOfWorkers;**

**table\_of\_qualification : TableOfQualification;**

{Массив сотрудников, непрошедших аттестацию}

**output\_table : InfoTable;**

**fatal\_error : boolean**

1. ПРОЕКТ АЛГОРИТМА ОБРАБОТКИ ДАННЫХ

p1: **STAFF\_ACCOUNTING** →

<Ввод и проверка исходных данных>;

if fatal\_error = false then <Формирование списка сотрудников, непрошедших аттестацию и вывод результата в файл **Output**>

p2: **<**Ввод и проверка исходных данных**>**→

if <**Catalog** существует> and <**Staff** существует> then

begin

<Ввод года **input\_year**>;

<Ввод месяца **input\_month**>;

<Ввод дня **input\_day**>;

if <**Catalog** не пуст> then

<Проверка файла **Catalog**>;

else fatal\_error := true;

if <**Staff** не пуст> and fatal\_error = false then

<Проверка файла **Staff**>;

else fatal\_error := true;

p3: <Проверка файла Catalog>→

line\_parsed := 0;

line\_counting := 1;

while (not EOF(Catalog)) and (line\_parsed <= MAX\_CORRECT\_LINES) do

<Проверка строки файла **Catalog**>;

if (line\_parsed = MAX\_CORRECT\_LINES + 1) then

writeln(‘WARNING: file Catalog has more 100 correct lines, data loose is possible’)

else if <Нет корректных данных в файле **Catalog**> then

fatal\_error := true;

p4: <Проверка строки файла Catalog>→

error := false;

readln(Catalog, s);

<Проверка на количество атрибутов в файле Catalog>;

if error = false then <Проверка на структуру строки>;

if error = false then <Проверка периода>;

if error = false then <Проверка профессии>;

if error = false then <Проверка на повторение профессии с другим периодом>;

if <Повторилась профессия с другим периодом> then fatal\_error := true;

if <Повторилась профессия с тем же периодом> then is\_duplicate := true;

if (error = false) and (is\_duplicate = false) then

begin

if line\_parsed <> MAX\_CORRECT\_LINES then

<Добавление данных в массив table\_of\_qualification>;

line\_parsed := line\_parsed + 1;

end;

line\_counting := line\_counting + 1;

p5: <Проверка файла Staff>→

line\_parsed := 0;

line\_counting := 1;

while (not EOF(Staff)) and (line\_parsed <= MAX\_CORRECT\_LINES) do

<Проверка строки файла Staff>;

if (line\_parsed = MAX\_CORRECT\_LINES + 1) then

writeln(‘WARNING: file Staff has more than 100 correct lines, data loose is possible’)

else if <Нет корректных данных в файле **Staff**> then

fatal\_error := true;

p6: <Проверка строки файла Staff>→

error := false;

readln(staff, s);

<Проверка на количество атрибутов в файле Staff>

if error = false then

begin

<Проверка имени>

<Проверка пола>

<Проверка профессии>

<Проверка даты аттестации>

<Проверка даты рождения>

<Проверка разности даты рождения и аттестации>

end;

if error = false then

begin

<Добавление данных в массив table\_of\_person>

line\_parsed := line\_parsed + 1;

end;

line\_counting := line\_counting + 1;

p7: <Формирование списка сотрудников, непрошедших аттестацию и вывод результата в файл **Output**> →

i := 0;

non\_passed := 0;

while table\_of\_person.name[i] <> ‘’ do

begin

is\_period\_exist := false;

j := 0;

while (j < MAX\_CORReCT\_LINES) and (is\_period\_exist = false) do

begin

if table\_of\_person[i].profession = table\_of\_qualification[j].profession then

begin

is\_period\_exist := true;

if (input\_date – table\_of\_person[i].certificate >= table\_of\_qualification[j].period) then

begin

<Добавляем данные в output\_table>

non\_passed := non\_passed + 1;

end;

end;

j := j + 1;

end;

if is\_period\_exist = false then

writeln(‘ERROR : there was no information about certification period for profession ‘, table\_of\_person[i].profession, ’ in file Catalog’);

i := i + 1;

end;

<Сортировать массив output\_table>;

for i := 0 to MAX\_CORRECT\_LINES – 1 do

if output\_table[i].name <> ‘’ then

<Записываем информацию в файл Output>;

# ТЕСТИРОВАНИЕ ПРОГРАММЫ “УЧЕТ СОТРУДНИКОВ”

1. Тестирование по методу “черного ящика”.
   1. Классы эквивалентности

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Входные**  **данные** | **Номер правильного к.э** | **Правильный к.э** | **Номер неправильного к.э** | **Неправильный к.э** |
| Файл Staff содержит информацию о персонале | 1 | Staff существует и не пуст | 19 | Staff не существует |
| 20 | Staff пуст |
| Файл Catalog содержит информацию о периоде аттестации | 2 | Catalog существует и не пуст | 21 | Catalog не существует |
| 22 | Catalog пуст |
| Файл Staff содержит не более 100 правильных строк | 3 | Кол-во правильных строк в Staff <= 100 | 23 | Кол-во правильных строк в Staff > 100 |
| Файл Catalog содержит не более 100 правильных строк | 4 | Кол-во правильных строк в Catalog <= 100 | 24 | Кол-во правильных строк в Catalog > 100 |
| Входная дата – последовательность чисел year, month, day | 5 | year = [2016..2017]  month = [01..12]  day = [01..31], если month = {1,3,5,7,8,10,12}  day = [01.30], если month = {4,6,9,11}  day = [01..29], если month = 2 и year делится на 4  day = [01..28], если month = 2 и year не делится на 4 | 25 | year < 2016 |
| 26 | year > 2017 |
| 27 | month < 1 |
| 28 | month > 12 |
| 29 | day < 1 |
| 30 | day > 31, если month = {1,3,5,7,8,10,12} |
| 31 | day > 30, если month = {4,6,9,11} |
| 32 | day > 29, если month = 2 и year кратен 4 |
| 33 | day > 28 если month = 2 и year не кратен 4 |
| Строка S – строка файла Staff | 6 | S = \*Фамилия\_и\_инициалы\*Пол\*Профессия\*Дата\_рождения\*Дата аттестации , \* - ненулевое количество символов пробела | 34 | Количество полей меньше 5 |
| Строка S – строка файла Catalog | 7 | S = Период\*Профессия , \* - ровно один символ пробела | 35 | Количество полей меньше 2 |
| 36 | Больше одного пробела между полями |
| Фамилия\_и\_инициалы – последовательность символов лат.алфавита, первый символ – заглавная буква, далее идут строчные, после которых идет точка, затем инициалы – заглавные буквы, разделенные точкой | 8 | Фамилия\_и\_инициалы = #\*.#.# , где # - заглавная буква латинского алфавита, \* - строка произвольной длины, состоящая из строчных букв латинского алфавита | 37 | Первый символ – не заглавная буква лат. алфавита |
| 38 | Хотя бы один не первый символ до первой точки – не строчная буква лат.алфавита |
| 39 | Символ после первой точки – не заглавная буква лат.алфавита |
| 40 | Символ после второй точки – не заглавная буква лат.алфавита |
| 41 | После первого инициала идут символы, отличные от точки |
| 42 | После второго инициала идут символы |
| 43 | Количество точек отлично от двух |
| Профессия – последовательность символов лат. алфавита, первый символ – заглавная буква, остальные, если существуют - строчные | 9 | Профессия = #\* , где  где # - заглавная буква латинского алфавита, \* - строка произвольной длины, состоящая из строчных букв латинского алфавита | 44 | Первый символ – не заглавная буква лат. алфавита |
| 45 | Хотя бы один не первый символ – не строчная буква лат. алфавита |
| Пол – Один из символов M или F | 10 | Пол = M | 46 | Пол не равен M или F |
| 11 | Пол = F |
| Дата рождения – состоит из 10 символов, символы 3 и 6 = ‘/’, первые два – число в диапазоне [01..31], символы 4 и 5 – число в дииапазоне [01..12], последние четыре – число в диапазоне [1950..2000]   |  |  | | --- | --- | | **Месяц** | **Кол-во дней** | | 1,3,5,7,8,10,  12 | 31 | | 4,6,9,11 | 30 | | 2 | 29, год кратен 4 | | 28, иначе | | 12 | Дата = D/M/Y,  Y= [1950..2000]  M = [01..31]  D = [01..31], если M = {1,3,5,7,8,10,12}  D = [01.30], если M = {4,6,9,11}  D = [01..29], если M = 2 и Y делится на 4  D = [01..28], если M = 2 и Y не делится на 4 | 47 | 3 или 6 символ не символы ‘/’ |
| 48 | Дата длиннее или короче 10 символов |
| 49 | Y < 1950 |
| 50 | Y > 2000 |
| 51 | M < 1 |
| 52 | M > 12 |
| 53 | D < 1 |
| 54 | D > 31, если M = {1,3,5,7,8,10,12} |
| 55 | D > 30, если M = {4,6,9,11} |
| 56 | D > 29, если M = 2 и Y кратен 4 |
| 57 | D > 28 если M = 2 и Y не кратен 4 |
| Дата аттестации – состоит из 10 символов, символы 3 и 6 = ‘/’, первые два – число в диапазоне [01..31], символы 4 и 5 – число в дииапазоне [01..12], последние четыре – число в диапазоне [2000..2017]   |  |  | | --- | --- | | **Месяц** | **Кол-во дней** | | 1,3,5,7,8,10,  12 | 31 | | 4,6,9,11 | 30 | | 2 | 29, год кратен 4 | | 28, иначе | | 13 | Дата = D/M/Y,  Y= [2000..2017]  M = [01..31]  D = [01..31], если M = {1,3,5,7,8,10,12}  D = [01.30], если M = {4,6,9,11}  D = [01..29], если M = 2 и Y делится на 4  D = [01..28], если M = 2 и Y не делится на 4 | 58 | 3 или 6 символ не символы ‘/’ |
| 59 | Дата длиннее или короче 10 символов |
| 60 | Y < 2000 |
| 61 | Y > 2017 |
| 62 | M < 1 |
| 63 | M > 12 |
| 64 | D < 1 |
| 65 | D > 31, если M = {1,3,5,7,8,10,12} |
| 66 | D > 30, если M = {4,6,9,11} |
| 67 | D > 29, если M = 2 и Y кратен 4 |
| 68 | D > 28 если M = 2 и Y не кратен 4 |
| Количество полных лет, прошедших между датой рождения и датой аттестации сотрудника | 14 | Количество полных лет, прошедших между датой рождения и датой аттестации сотрудника > 17 | 69 | Количество полных лет, прошедших между датой рождения и датой аттестации сотрудника <= 17 |
| Период – натуральное число в диапазоне, является первым символом в строке файла Catalog[1..3] | 15 | Период принадлежит [1..3] и период – первый символ в строке файла Catalog | 70 | Период > 3 |
| 71 | Период < 1 |
| 72 | Период принадлежит диапазону [1..3], период не является первым символом в строке файла Catalog |
| Уникальность периода аттестации для каждой профессии | 16 | В файле Catalog нет одинаковых профессий с разными периодами аттестаций | 73 | В файле Catalog есть одинаковые профессии с разными периодами аттестаций |
| Повторение профессии с тем же периодом аттестации | 17 | В файле Catalog нет одинаковых профессий с одинаковыми периодами аттестаций | 74 | В файле Catalog есть одинаковые профессии с одинаковыми периодами аттестаций |
| Отсутствие информации о периоде аттестации данной профессии | 18 | В файле Catalog для всех профессий, что находятся в корректных строках файла Staff есть информация о периоде аттестации | 75 | В корректных строках файла Staff есть профессии, для которых отсутствует информация о периоде аттестации в файле Catalog |
| Корректные строки в файле Staff | 76 | В файле Staff присутствуют корректные строки | 78 | В файле Staff отсутствуют корректные строки |
| Корректные строки в файле Catalog | 77 | В файле Catalog присутствуют корректные строки | 79 | В файле Catalog отсутствуют корректные строки |

1.2 Тесты

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Выходные данные** | **Номера к.э.** |
| Файл Staff отсутствует, файл Catalog отсутствует | Error has occurred during opening file Staff.txt  Error has occurred during opening file Catalog.txt | 19,21 |
| input\_year = 2016, input\_month = 2, input\_day = 2  Файл Staff:  Файл Catalog: | File Catalog is empty  File Staff is empty | 20,22 |
| input\_year = 2000, input\_month = 1, input\_day = 23 | Year must be in range [2016..2017], please, repeat input | 25 |
| input\_year = 2022, input\_month = 1, input\_day = 23 | Year must be in range [2016..2017], please, repeat input | 26 |
| input\_year = 2017, input\_month = -4, input\_day = 23 | Month must be in range [1..12], please, repeat input | 27 |
| input\_year = 2016, input\_month = 31, input\_day = 23 | Month must be in range [1..12], please, repeat input | 28 |
| input\_year = 2017 input\_month = 8, input\_day = 0 | Day must be in range [1..31], please, repeat input | 29 |
| input\_year = 2017 input\_month = 1, input\_day = 40 | Day must be in range [1..31], please, repeat input | 30 |
| input\_year = 2017 input\_month = 4, input\_day = 31 | Day must be in range [1..30], please, repeat input | 31 |
| input\_year = 2016 input\_month = 2, input\_day = 46 | Day must be in range [1..29], please, repeat input | 32 |
| input\_year = 2017 input\_month = 2, input\_day = 112 | Day must be in range [1..28], please, repeat input | 33 |
| input\_year = 2017 input\_month = 2, input\_day = 12  Файл Staff:  Файл Catalog:  2 Worker  3 Worker | Checking file Catalog  (2)FATAL: profession Worker is repeating with different period  Fatal error occurred during parsing Catalog, terminating  File Staff is empty | 1,2,3,4,5, 17, 73 |
| input\_year = 2017 input\_month = 2, input\_day = 12  Файл Staff:  Sean.Q.Q M Worker 01/02/1968 12/12/2015  Sean.Q.Q M Worker 01/02/1999 12/12/2010  sean.Q.Q M borker 29/02/1969 29/02/2009  Sean.Q.Q M borker 29/02.1969 29e02/2009  SeAn.Q.Q M WAker 30/02/1968 30/02/2015  SeAn.Q.Q M Wker 30/0002/1968 1232/12/2015  Sean.2.Q M Worker 31/04/1939 31/04/1215  Sean.Q.3 M Worker 32/01/9999 32/01/2015  Sean.Qasd.Q M Worker 01/00/1968 12/00/2077  Sean.Q.Qsdfdsf F Worker -6/13/1968 12/21/2015  Sean.Q...Q Male Director 01/02/1968 00/12/2015    Файл Catalog:  2 Worker  2 Worker  0 Doctor  3 as  5 Salt | Checking file Catalog  (2) WARNING: profession Worker is repeating with same period  (3) ERROR: period must be first character in line, in range [1..3]  (4) ERROR: second symbol must be space character  (5) ERROR: period must be first character in line, in range [1..3]  Checking file Staff  (2) ERROR: difference between birth date and certification date must be greater than 17 years  (3) ERROR: name sean.Q.Q must consist of surname, that starts with with capital letter, followed by small latin and initials, that are capital lati letters, separated by dots, for example: Surname.S.S  (3) ERROR: profession borker must start with capital letter and other symbols must be only small latin letters  (3) ERROR: Birth date day 29/02/1969 must be in range [1..28]  (3) ERROR: Certification date day 29/02/2009 must be in range [1..28]  (4) ERROR: profession borker must start with capital letter and other symbols must be only small latin letters  (4) ERROR: day, month and year in Birth date 29/02.1969 must be separated with symbol /  (4) ERROR: day, month and year in Certification date 29e02/2009 must be separated with symbol /  (5) ERROR: name SeAn.Q.Q must consist of surname, that starts with with capital letter, followed by small latin and initials, that are capital lati letters, separated by dots, for example: Surname.S.S  (5) ERROR: profession WAker must start with capital letter and other symbols must be only small latin letters  (5) ERROR: Birth date day 30/02/1968 must be in range [1..29]  (5) ERROR: Certification date day 30/02/2015 must be in range [1..28]  (6) ERROR: name SeAn.Q.Q must consist of surname, that starts with with capital letter, followed by small latin and initials, that are capital lati letters, separated by dots, for example: Surname.S.S  (6) ERROR: Birth date 30/0002/1968 must be 10 symbols long  (6) ERROR: Certification date 1232/12/2015 must be 10 symbols long  (7) ERROR: name Sean.2.Q must consist of surname, that starts with with capital letter, followed by small latin and initials, that are capital lati letters, separated by dots, for example: Surname.S.S  (7) ERROR: Birth date year 31/04/1939 must be in range [1950..2000]  (7) ERROR: Birth date day 31/04/1939 must be in range [1..30]  (7) ERROR: Certification date year 31/04/1215 must be in range [2000..2017]  (7) ERROR: Certification date day 31/04/1215 must be in range [1..30]  (8) ERROR: name Sean.Q.3 must consist of surname, that starts with with capital letter, followed by small latin and initials, that are capital lati letters, separated by dots, for example: Surname.S.S  (8) ERROR: Birth date year 32/01/9999 must be in range [1950..2000]  (8) ERROR: Birth date day 32/01/9999 must be in range [1..31]  (8) ERROR: Certification date day 32/01/2015 must be in range [1..31]  (9) ERROR: name Sean.Qasd.Q must consist of surname, that starts with with capital letter, followed by small latin and initials, that are capital lati letters, separated by dots, for example: Surname.S.S  (9) ERROR: Birth date month 01/00/1968 must be in range [1..12]  (9) ERROR: Certification date year 12/00/2077 must be in range [2000..2017]  (9) ERROR: Certification date month 12/00/2077 must be in range [1..12]  (10) ERROR: name Sean.Q.Qsdfdsf must consist of surname, that starts with with capital letter, followed by small latin and initials, that are capital lati letters, separated by dots, for example: Surname.S.S  (10) ERROR: Certification date month 12/21/2015 must be in range [1..12]  (11) ERROR: name Sean.Q...Q must consist of surname, that starts with with capital letter, followed by small latin and initials, that are capital lati letters, separated by dots, for example: Surname.S.S  (11) ERROR: gender Male must be either F or M  (11) ERROR: Certification date day 00/12/2015 must be in range [1..31]  All staff has passed certification | 6, 7, 8, 9, 10, 11, 12, 13, 14,15, 16, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46. 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 74, 75, 76, 77 |
| input\_year = 2017 input\_month = 2, input\_day = 12  Файл Staff:  Sean.Q.Q M Worker 01/02/1968 12/12/2015  Файл Catalog:  2 Worker |  | 18 |
| input\_year = 2017 input\_month = 2, input\_day = 12  Файл Staff:  Sean.Q.Q Male Worker 01/02/1968 12/12/2015  Файл Catalog:  2 dosrker | Checking file Catalog  (1) ERROR: profession dosrker must start with capital letter and other symbols must be only small latin letters  File Catalog has no correct information  Fatal error occured during parsing Catalog, terminating | 79 |
| input\_year = 2017 input\_month = 2, input\_day = 12  Файл Staff:  Sean.Q.Q Male Worker 01/02/1968 12/12/2015  Файл Catalog:  2 Salty | Checking file Catalog  Checking file Staff  (1) ERROR: gender Male must be either F or M  File Staff has no correct information | 78 |
| input\_year = 2017 input\_month = 8, input\_day = 26  Файл Staff:  Arky.Q.Q M Guardian 10/10/1985 20/03/2015  Celestine.D.A F Doctor 20/03/1980 20/05/2014  Файл Catalog:  2 Guardian  3 Doctor | Checking file Catalog  Checking file Staff  Файл Output:  Celestine.D.A Doctor 20/5/2014  Arky.Q.Q Guardian 20/3/2015 |  |

2. Структурное тестирование

Пример кода:

function IsLesser(left, right : Date):boolean;

begin

IsLesser := false;

1) if (left.year < right.year) then

IsLesser := true

2) else if (left.year = right.year) and (left.month < right.month) then

IsLesser := true

3) else if (left.year = right.year) and (left.month = right.month) and (left.day < right.day) then

IsLesser := true;

end;

|  |  |
| --- | --- |
| **Входные данные** | **Номер покрываемого оператора** |
| Left.year = 1987, left.month = 12, left.day = 11,  right.year = 2013, right.month = 10, right.day = 17 | 1) |
| Left.year = 1987, left.month = 3, left.day = 11,  right.year = 1987, right.month = 10, right.day = 17 | 2) |
| Left.year = 2013, left.month = 12, left.day = 11,  right.year = 2013, right.month = 12, right.day = 17 | 3) |