

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

**ОТЧЕТ**

**о выполнении индивидуального задания по теме «Записи»**

в рамках освоения дисциплины

«Основы алгоритмизации и программирования»

направления «Программная инженерия»

|  |  |  |
| --- | --- | --- |
| Отчет защищен с оценкой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_Лаптева А.А.         (подпись)  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г. |  | Выполнил студент  группы Б9121-09.03.04  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Р.В. Шевелев  (подпись) |
|  |  | Руководитель  к.ф.-м.н, доцент  Лаптева Анастасия Александровна |
|  |  |  |

Владивосток

2022

**Аннотация**

Алгоритм решения задачи реализован в виде программного кода, написанного на алгоритмическом языке Pascal ABC.

Общее назначение программы: обработка сведений об учащихся. Выборка учащихся, которые соответствуют определенным критериям. Получение подробной статистики. Назначением данной программы является обеспечение удобства и быстродействия обработки предоставленных данных

Проект программы разработан с использованием принципов нисходящегопроектирования; алгоритм программы основан на принципеструктурногопрограммирования.

Программа может обработать любое количество входных данных, при условии, что все они корректны.

**Оглавление**

[Неформальная постановка задачи 4](#_Toc100310416)

[Формальная постановка задачи 4](#_Toc100310417)

[Алгоритм решения задачи 5](#_Toc100310418)

[Спецификация данных 6](#_Toc100310419)

[Спецификация функций 9](#_Toc100310420)

[Проектирование программы 9](#_Toc100310421)

[Описание данных программы 10](#_Toc100310422)

[Алгоритм программы на языке PDL 14](#_Toc100310423)

[Тесты 18](#_Toc100310425)

[Тестирование программы 19](#_Toc100310426)

[Список использованных источников 21](#_Toc100310427)

Неформальная постановка задачи

В школьном медицинском кабинете содержатся следующие сведения об учащихся: фамилия ученика, класс (1–11), литер класса, пол (М/Ж), рост(см), вес(кг). База данных содержит не более 1000 таких сведений. Найти:

1. Рост самого высокого и самого низкого ученика в каждой параллели классов (параллель класса – все литеры для одного значения класса);
2. Процент девочек, чей вес превышает 60 кг;
3. Количество учеников 10 и 11 классов школы, чей ИМТ в пределах от 20 до 25 единиц

Формальная постановка задачи

**Input:**

data\_base = {data[i] | i<=1000} – множество всех учащихся

data[i] = {surname[i],rank[i],liter[i],sex[i],weight[i],height[i]}

surname[i][k] – фамилия учащегося k ∈ {“А”,…..,”Я”,”а”,…….,”я” | k ∈ N, k<=20}

rank[i] ∈ {x < 12 x ∈ N} – класс i – го ученика

liter[i] ∈{“А..я”} – литер класса i – го ученика

sex[i] ∈ {“М”, “Ж”} – пол i – го ученика

weight[i] ∈ {x ∈ R, x>0} – вес i – го ученика

height[i] ∈ {x ∈ R, x>0} - рост i – го ученика

**Output:**

height\_max[i] ∈ R | i ∈ {1...11} i – номер класса, height\_max[i] - максимальный рост среди учеников i-х классов.

height\_min[i] ∈ R | i ∈ {1…11} i – номер класса, height\_max[i] - минимальный рост среди учеников i-х классов.

percent\_of\_girls\_over\_60 ∈ R | ∈ {0…100} – процент девочек среди всех учеников, чей вес превышает 60 кг.

imt\_between\_20\_and\_25 ∈ Z | ∈ {0…1000} – количество учеников 10 и 11 классов, чей индекс массы тела лежит в пределах от 20 до 25 единиц.

**Связи**

1. height\_max[i] = {height[i] >= height[i][k] | ∀ k<=1000, k ∈ Z | i ∈ {1…11}}
2. height\_min[i] = {height[i] <= height[i][k] | ∀ k<=1000, k ∈ Z | i ∈ {1…11}}
3. percent\_of\_girls\_over\_60 = **|** surname[i] , sex[i]=’Ж’ , weight[i]>60 **| /** {i, i<=1000} \* 100 – процент девочек, чей вес превышает 60 кг. i – количество всех учеников школы.
4. imt\_between\_20\_and\_25 = **|** surname[i] , rank[i]>=10 , 20 <= 100\*weight[i]/height[i] <= 25 **|** – количество учеников 10 и 11 классов школы, чей индекс массы тела в пределах от 20 до 25. Вертикальные линии в данном случае означают мощность множества, то есть количество элементов, которые удовлетворяют заданным критериям.

Алгоритм решения задачи

1. Привязка файла с данными об учащихся.
2. Проверка ввода
   1. В случае нахождения ошибок – вывод сообщений с номерами (1–16),выход из программы, иначе – продолжать.
3. Формирование массива записей
4. Работа с массивом записей
   1. Поиск самого высокого и самого низкого ученика в каждой параллели классов внутри цикла. Выборка по критериям: класс, рост.

Запись в файл сообщения 18.

* 1. Поиск процента девочек, вес которых превышает 60 кг. Выборка по критериям: пол, вес.

Запись в файл сообщения 19.

* 1. Подсчет количества учеников 10 и 11 класса, чей ИМТ (индекс массы тела) в пределах от 20 до 25. Выборка по критериям: класс, рост, вес, их соотношение.

Запись в файл сообщения 20.

1. Вывод на экран сообщения 17 об успешном выполнении программы.
2. Конец

Спецификация данных

**Спецификация ввода**

*1000.txt -* текстовый файл, содержащий 6 столбцов и 1000 строк (зависит от выбранного файла, во второй строчке программы можно изменить количество учащихся: 50, 100, 200, 500, 1000)



Порядок данных имеет значение:

<Фамилия> <Класс> <Литер> <Пол> <Вес> <Рост>

. . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . .

<Фамилия> <Класс> <Литер> <Пол> <Вес> <Рост>

**Спецификация вывода**

Вывод всех следующих сообщений осуществляется на экран. В текстовый файл выводятся только сообщения 16–18.

output.txt – текстовый файл, в который выводится результат работы программы (в случае корректного ввода данных).

Данные в файл output.txt выводятся в следующем формате:

Изображение выглядит как текст

Автоматически созданное описание

Сообщения об ошибках, выводящиеся на экран:

Сообщение 1: «Файл не найден! Измените 2-ю строку программы для корректной работы.»

Сообщение 2: «Файл пуст»

Сообщение 3: «Неверное количество столбцов в строке номер: <num>»

Сообщение 4: «Количество строк в файле не совпадает с количеством человек. Проверьте вводимые данные.»

Сообщение 5: «Невозможно распознать фамилию в строке номер: <число>»

Сообщение 6: «В русских школах только 11 классов! Ошибка в строке: <число>»

Сообщение 7: «Невозможно распознать класс в строке номер: <число>»

Сообщение 8: «Литер состоит из одного символа!. Ошибка в строке: <число>»

Сообщение 9: «Литер может содержать только русские символы от А до ж. Ошибка в строке: <число>»

Сообщение 10: «Пол обозначается одним символом! Ошибка в строке: <число>»

Сообщение 11: «Пол обозначается следующими символами: Ж ж М м. Ошибка в строке: <число>»

Сообщение 12: «Невозможно распознать пол ученика в строке номер: <число>»

Сообщение 13: «Невозможно распознать вес ученика в строке номер: <число>»

Сообщение 14: «Невозможно распознать рост ученика в строке номер: <число>»

Сообщение 15: «'Вес ученика обязательно больше нуля! Ошибка в строке: ', <число>»

Сообщение 16: «'Рост ученика обязательно больше нуля! Ошибка в строке: ', <число>»

Сообщение 17: «Программа завершена корректно.»

Сообщения, которые записываются в файл при успешном выполнении программы:

Сообщение 18: «<число>, ' класс: ', 'Минимальный рост: ', <число>, ' Максимальный рост: ', <число>»

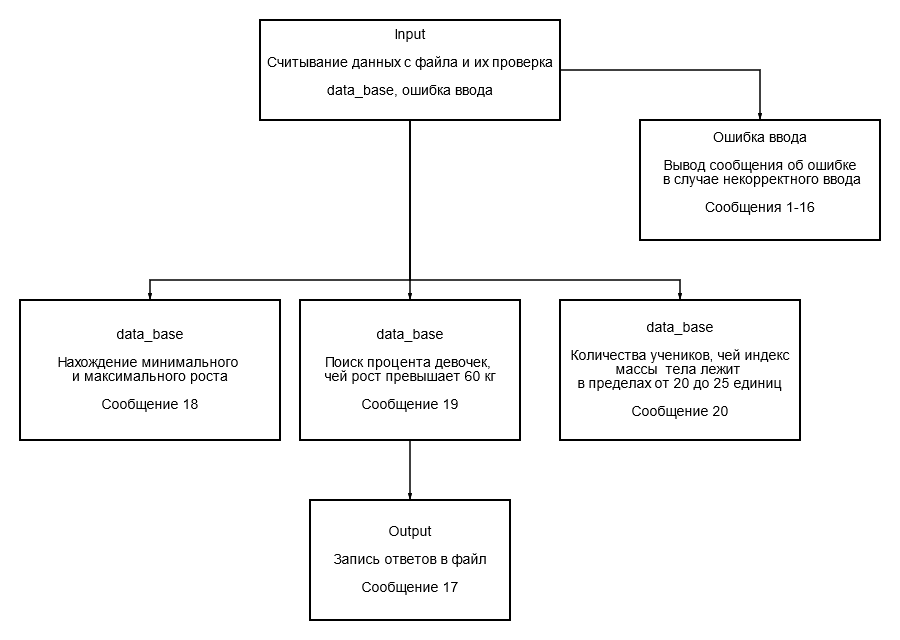
Сообщение 19: «'Процент девочек, чей вес превышает 60кг: ', <число> , ' процентов'»

Сообщение 20: 'Ученики 10 и 11 классов, чей индекс массы тела находится между 20 и 25: ', <число>, ' человек')

Спецификация функций

1. Считывание данных с файла и их проверка
2. Вывод сообщения об ошибке в случае некорректного ввода данных.
3. Обработка данных. Нахождение:
   1. Минимального и максимального роста
   2. Процента девочек, чей рост превышает 60 кг
   3. Количества учеников, чей индекс массы тела лежит в пределах от 20 до 25 единиц
4. Запись ответов в файл

Проектирование программы



Описание данных программы

|  |  |  |
| --- | --- | --- |
| Идентификатор | Назначение | Описание |
| persons | Константа. Количество учащихся в школе. Одновременно используется для указания пути к входным данным. | const |
| data\_record | Тип, созданный для записи данных об учащихся в школе. (Фамилия, класс, литер, пол, вес, рост) | record |
| surname | Строковая переменная. Фамилия ученика. До 20 символов | string[20] |
| rank | Числовая переменная.Класс ученика | 1..11 integer |
| liter | Строковая переменная. Литер ученика. 1 символ | string[1] |
| sex | Строковая переменная Пол ученика. 1 символ | string[1] |
| height | Переменная, хранящая рост ученика. Вещественное положительное число. | real |
| weight | Переменная, хранящая вес ученика. Вещественное положительное число. | real |
| data\_base | Переменная файлового типа. Для работы с текстовым документом. Input | text |
| otvet | Переменная файлового типа. Для работы с текстовым документом. Output | text |
| date | Переменная типа data\_record | record |
| data | Динамический массив записей. Размер задается в соответствии с количеством учащихся | array of data\_record |
| line | Переменная строкового типа для обработки данных из файла | String |
| num | Переменная целочисленного типа для отслеживания номера текущей строки в файле | Integer |
| quit | Переменная логического типа для выхода из программы при наличии ошибок ввода/ошибок данных | Boolean |
| en3 | Переменная, наследующая подкласс класса Encoding. Создана для корректной привязки файла с последующим отображением русских символов. | Encoding |
| vvod | Процедура для отработки ошибок ввода. Не принимает переменных на вход. При наличии таковых, выдает сообщения (1-4) | Procedure |
| records | Процедура для обработки учащихся и добавления записей в массив data | Procedure |
| max\_min\_parallels | Процедура для поиска ответа на Вопрос 1. Не принимает переменных на вход. Включает в себя обработку данных по определенным критериям | procedure |
| percent\_of\_girls | Процедура для поиска ответа на Вопрос 2. Не принимает переменных на вход. Включает в себя обработку данных по определенным критериям | procedure |
| imt | Процедура для поиска ответа на Вопрос 3. Не принимает переменных на вход. Включает в себя обработку данных по определенным критериям | procedure |
| Локальный контекст процедуры vvod | | |
| road | Локальная переменная процедуры vvod строкового типа. Содержит в себе путь до файла с базой учащихся. | String |
| file\_chars | Локальная переменная процедуры vvod строкового типа. Для считывания строки в файле с данными учащихся. | String |
| count | Локальная переменная процедуры vvod целочисленного типа. Для проверки на пустой файл | Integer |
| Локальный контекст процедуры records | | |
| road | Локальная переменная процедуры records строкового типа. Содержит в себе путь до файла с базой учащихся. | String |
| Локальный контекст процедуры max\_min\_parallels | | |
| rost\_min | Локальная переменная процедуры max\_min\_parallels. Массив вещественных чисел размером 11.  Содержит в себе минимальный рост учеников 1-11 классов | array[1..11] of real |
| rost\_max | Массив вещественных чисел размером 11.  Содержит в себе максимальный рост учеников 1-11 классов | array[1..11] of real |
| Локальный контекст процедуры percent\_of\_girls | | |
| count | Локальная переменная целочисленного типа процедуры percent\_of\_girls. Содержит в себе количество девочек, чей вес превышает 60 кг | integer |
| Локальная переменная процедуры imt | | |
| count | Локальная переменная целочисленного типа процедуры imt. Содержит в себе количество учеников 10 и 11 классов, чей индекс массы тела находится в пределах от 20 до 25 единиц. | integer |

Алгоритм программы на языке PDL

**const**

persons = 1000; //50 или 100 или 200 или 500 или 1000

**type**

data\_record = **record**

surname: string[20];

rank: 1..11;

liter, sex: string[1];

height, weight: real;

**end**;

**var**

data\_base, otvet: text;

date: data\_record;

data: **array of** data\_record;

line: string;

num: integer;

quit: boolean;

en3 := Encoding.GetEncoding(65001);

**procedure** vvod();

**var**

road, file\_chars: string;

count: integer;

**begin**

road := persons + '.txt';

assign(data\_base, road);

**if not** fileexists(road) **then begin**

writeln('Файл ', road, ' не найден! Измените 2-ю строку программы для корректной работы.');

quit := True;

**end**;

reset(data\_base, en3);

**while not** EOF(data\_base) **do**

**begin**

readln(data\_base, file\_chars);

num := num + 1;

count += length(file\_chars);

**end**;

**if** count = 0 **then begin**

writeln('Файл пуст');

quit := True;

**end**;

close(data\_base);

reset(data\_base, en3);

num := 0;

**while not** EOF(data\_base) **do**

**begin**

readln(data\_base, file\_chars);

num := num + 1;

**if** (file\_chars.ToWords(' ')).Length <> 6 **then begin**

writeln('Неверное количество столбцов в строке номер: ', num);

quit := True;

**end**;

**end**;

close(data\_base);

**if** num <> persons **then begin**

writeln('Количество строк в файле не совпадает с количеством человек. Проверьте вводимые данные.');

quit := True;

**end**;

**end**;

**procedure** records();

**var**

road: string;

**begin**

road := persons + '.txt';

assign(data\_base, road);

reset(data\_base, en3);

num := 1;

**while not** EOF(data\_base) **do**

**begin**

readln(data\_base);

num := num + 1;

**end**;

close(data\_base);

data := **new** data\_record[num];

num := 0;

assign(data\_base, road);

reset(data\_base, en3);

**while not** EOF(data\_base) **do**

**begin**

readln(data\_base, line);

date.surname := line.ToWords(' ')[0].ToString;

**for var** i := 1 **to** length(date.surname) **do**

**begin**

**if** date.surname[i] **not in** 'абвгдеёжзийклмнопрстуфхцчшщъыьэюяАБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ' **then begin**

quit := True;

**if** quit **then** writeln('Невозможно распознать фамилию в строке номер: ', num + 1);

**end**;

**end**;

**try**

date.rank := line.ToWords(' ')[1].ToInteger;

**if** (date.rank > 11) **or** (date.rank < 1) **then begin**

writeln('В русских школах только 11 классов! Ошибка в строке: ', num + 1);

quit := True;

**end**;

**except**

writeln('Невозможно распознать класс в строке номер: ', num + 1);

quit := True;

**end**;

**if** line.ToWords(' ')[2].ToString.Length <> 1 **then begin**

writeln('Литер состоит из одного символа!. Ошибка в строке: ', num + 1);

quit := True;

**end**;

date.liter := line.ToWords(' ')[2].ToString;

**if** date.liter **not in** 'абвгдеёжзийклмнопрстуфхцчшщъыьэюяАБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ' **then begin**

writeln('Литер может содержать только русские символы от А до я. Ошибка в строке: ', num + 1);

quit := True;

**end**;

date.sex := line.ToWords(' ')[3];

**if** (line.ToWords(' ')[3].Length <> 1) **then begin**

writeln('Пол обозначается одним символом! Ошибка в строке: ', num + 1);

quit := True;

**end**;

**if** (date.sex **in** 'qwertyuiopasdfghjklzxcvbnm') **then begin**

writeln('Невозможно распознать пол ученика в строке номер: ', num + 1);

quit := True;

**end**;

**if** (date.sex **not in** 'ЖжМм') **then begin**

writeln('Пол обозначается следующими символами: Ж ж М м. Ошибка в строке: ', num + 1);

quit := True;

**end**;

**try**

date.weight := line.ToWords(' ')[4].ToReal;

**if** (date.weight) <= 0 **then begin**

writeln('Вес ученика обязательно больше нуля! Ошибка в строке: ', num + 1);

quit := True;

**end**;

**except**

writeln('Невозможно распознать вес ученика в строке номер: ', num + 1);

quit := True;

**end**;

**try**

date.height := line.ToWords(' ')[5].ToReal;

**if** (date.height) <= 0 **then begin**

writeln('Рост ученика обязательно больше нуля! Ошибка в строке: ', num + 1);

quit := True;

**end**;

**except**

writeln('Невозможно распознать рост ученика в строке номер: ', num + 1);

quit := True;

**end**;

data[num] := date;

num := num + 1;

**end**;

close(data\_base);

**end**;

**procedure** max\_min\_parallels();

**var**

rost\_min: **array**[1..11] **of** real;

rost\_max: **array**[1..11] **of** real;

**begin**

**for var** i := 1 **to** 11 **do**

**begin**

rost\_min[i] := 9999;

rost\_max[i] := -1;

**end**;

**for var** i := 0 **to** persons - 1 **do**

**begin**

**if** data[i].height < rost\_min[data[i].rank] **then**

rost\_min[data[i].rank] := data[i].height;

**if** data[i].height > rost\_max[data[i].rank] **then**

rost\_max[data[i].rank] := data[i].height;

**end**;

assign(otvet, 'output.txt');

rewrite(otvet, en3);

writeln(otvet, '1 вопрос: ');

writeln(otvet);

**for var** i := 1 **to** 11 **do**

**begin**

write(otvet, i, ' класс: ', 'Минимальный рост: ', rost\_min[i], ' Максимальный рост: ', rost\_max[i]);

write(otvet, #10);

**end**;

writeln(otvet);

close(otvet);

**end**;

**procedure** percent\_of\_girls();

**var**

count: integer;

**begin**

count := 0;

**for var** i := 0 **to** persons - 1 **do**

**begin**

**if** (data[i].sex = 'Ж') **and** (data[i].weight > 60) **then** count += 1;

**end**;

assign(otvet, 'output.txt');

append(otvet, en3);

writeln(otvet, '2 вопрос: ');

writeln(otvet);

writeln(otvet, 'Процент девочек, чей вес превышает 60кг: ', round(count / persons \* 100, 2), ' процентов');

close(otvet);

**end**;

**procedure** imt();

**var**

count: integer;

**begin**

count := 0;

**for var** i := 0 **to** persons - 1 **do**

**begin**

**if** (data[i].rank >= 10) **and** ((100 \* data[i].weight / data[i].height) <= 25) **and** ((100 \* data[i].weight / data[i].height) >= 20) **then begin**

count := count + 1;

**end**;

**end**;

assign(otvet, 'output.txt');

append(otvet, en3);

writeln(otvet);

writeln(otvet, '3 вопрос: ');

writeln(otvet);

writeln(otvet, 'Ученики 10 и 11 классов, чей индекс массы тела находится между 20 и 25: ', count, ' человек');

close(otvet);

**end**;

**begin**

vvod();

**if** quit **then exit**();

records();

**if** quit **then exit**();

max\_min\_parallels();

percent\_of\_girls();

imt();

writeln('Программа завершена корректно.');

end.

Тесты

Метод тестирования: Белый ящик.

|  |  |  |
| --- | --- | --- |
| № | Input | Output |
| 1 | Отсутствие файла | Файл 1000.txt не найден! Измените 2-ю строку программы для корректной работы. |
| 2 | Пустой файл | Файл пуст |
| 3 | Шевченко 5 а Ж 32.4 180.5 s s s | Неверное количество столбцов в строке номер: 1 |
| 4 | Persons = 2  Шевченко 5 а Ж 32.4 180.5  Калугина 6 г Ж 41.9 131.5  Розанова 8 а Ж 56.6 148 | Количество строк в файле не совпадает с количеством человек. Проверьте вводимые данные. |
| 5 | Шев34ченко 5 а Ж 32.4 180.5 | Невозможно распознать фамилию в строке номер: 1 |
| 6 | Шевченко 5 а Ж 32.4 180.5  Калугина 6 г Ж 41.9 131.5  Розанова 18 а Ж 56.6 148 | В русских школах только 11 классов! Ошибка в строке: 3 |
| 7 | Шевченко s а Ж 32.4 180.5 | Невозможно распознать класс в строке номер: 1 |
| 8 | Шевченко 5 а2 Ж 32.4 180.5  Калугина 6 г Ж 41.9 131.5 | Литер состоит из одного символа!. Ошибка в строке: 1 |
| 9 | Шевченко 5 а Ж 32.4 180.5  Калугина 6 г Ж 41.9 131.5  Розанова 8 1 Ж 56.6 148 | Литер может содержать только русские символы от А до я. Ошибка в строке: 3 |
| 10 | Сорокин 10 б Мужской 54.8 221.5  Дмитриев 4 б М 39.7 147.5  Ефимов 6 в М 30.7 130  Нестеров 4 б М 46.6 143 | Пол обозначается одним символом! Ошибка в строке: 1 |
| 11 | Сорокин 10 б 0 54.8 221.5  Дмитриев 4 б М 39.7 147.5 | Пол обозначается следующими символами: Ж ж М м. Ошибка в строке: 1 |
| 12 | Сорокин 10 б d 54.8 221.5 | Невозможно распознать пол ученика в строке номер: 1 |
| 13 | Сорокин 10 б М -54.8 221.5  Дмитриев 4 б М 39.7 147.5  Ефимов 6 в М 30.7 130  Нестеров 4 б М 46.6 143 | Вес ученика обязательно больше нуля! Ошибка в строке: 1 |
| 14 | Чеботарёва 8 б Ж фыв 149  Вишняков 8 б М 46.5 168  Зыкова 10 б Ж 70.5 210.5  Харитонова 1 б Ж 27.4 123.5 | Невозможно распознать вес ученика в строке номер: 1 |
| 15 | Чеботарёва 8 б Ж 54 149  Вишняков 8 б М 46.5 168  Зыкова 10 б Ж 70.5 -210.5  Харитонова 1 б Ж 27.4 123.5 | Рост ученика обязательно больше нуля! Ошибка в строке: 3 |
| 16 | Нестеров 4 б М 46.6 143  Шевченко 5 а Ж 32.4 13asd  Калугина 6 г Ж 41.9 131.5 | Невозможно распознать рост ученика в строке номер: 2 |
| 17 | Нестеров 4 б М 46.6 143  Шевченко 5 а Ж 32.4 136.5  Калугина 6 г Ж 41.9 131.5  Чеботарёва 8 б Ж 54 149  Вишняков 8 б М 46.5 168  Зыкова 10 б Ж 70.5 210.5  Орлова 2 ж Ж 26.7 110  Моисеева 2 е Ж 23.1 122.5  Рябинина 1 а Ж 23.3 113.5  Макарова 10 в Ж 71.8 220.5  Фёдоров 6 ж М 40.0 150.5 | Программа завершена корректно. |

Тестирование программы

|  |  |
| --- | --- |
| № | Результат |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |
| 16 |  |
| 17 |  |

Список использованных источников

1. PascalABC.NET. Современное программирование на языке Pascal. Записи. Электронный ресурс. – Режим доступа: https://pas1.ru/record [дата обращения 17.03.2022].
2. PascalABC.NET: Введение в современное программирование. – Ростов-на-Дону, 2019 – 572с.