МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет прикладной математики, информатики и механики

Кафедра математического обеспечения ЭВМ

## КУРСОВАЯ РАБОТА

по дисциплине «Информатика и программирование»

*Направление 02.03.02 – Фундаментальная информатика и информационные технологии*

*Профиль – Инженерия программного обеспечения (ФГОС3++)*

Зав. Кафедрой, доктор техн. наук, проф. Ф.И.Зав кафедрой

Обучающийся, студент 1 курса Ф.И. Студента

Руководитель, канд. техн. наук, доцент Ф.И. Препода

### Город Год

Содержание

1.Постановка задачи1

2.Описание данных и алгоритма работы программы2

3.Описание структуры программы3

4.Результаты тестирования4

5.Список использованных источников5  
 6.Приложение6

1. Постановка задачи

С клавиатуры вводится информация об итогах последней экзаменационной сессии. Эта информация включает в себя: 1) целое число *n* – количество студентов; 2) *n* объединенных в структуру данных:

<имя ><фамилия>  <оценка>  <оценка> <оценка>  <оценка> <оценка>

где <имя>, <фамилия> ― символьные строки, содержащие не более 20 символов, оценка за экзамен ― десятичная цифра из диапазона ‘2’..’5’.

Требуется сформировать совокупность структур, в котором каждый элемент совокупности содержит фамилию студента и его средний балл, а элементы располагаются в порядке убывания средних баллов, и вывести элементы совокупности на экран. Вывести также фамилии и оценки студентов, имеющих наибольшее количество отличных оценок.

*Технические требования.*

 В основной памяти исходные данные и формируемую совокупность структур следует хранить в виде массива указателей на структуры.

Не допускается использование дополнительных массивов.

Для реализации некоторых действий следует определить и применить функции.

2. Описание данных и алгоритма работы программы

Для реализации решения задачи с использованием динамических массивов использовались следующие простые и структурированные данные: количество студентов **number\_of\_students,** динамические массивы структур **students** и **new\_students** (для вводимой и требуемой информации соответственно), структура **Student**, содержащая фамилию **surname** и имя **name** – статические массивы типа char с ограничением в 20 элементов, целочисленный массив **marks** оценок студента типа short размера 5. Структура **ShortStudent** содержит меньше данных относительно структуры **Student**, только фамилию студента **surname** - статический массив типа char с ограничением в 20 элементов и средний балл студента - переменная **ball** типа float.

В программе используются функции:

input\_number – функция, с помощью которой осуществляется ввод количества студентов с клавиатуры.

initialization\_students – фукнция для ввода первоначальных данных: имени и фамилии студентов, а также пяти оценок, которая включает в себя input\_mark – функцию ввода оценок с проверкой, что введенное число действительно является оценкой – число в диапазоне [2; 5].

write\_middle\_ball\_to\_new\_srtucture - фукнция, которая по уже полученной информации о студентах формирует динамический массив структур типа ShortStudent, содержащий в себе number\_of\_students структур типа ShortStudent.

sorting\_structures – функция, сортирующая массив структур по убыванию среднего балла студентов методом пузырьковой сортировки, в которой используется встроенная функция, меняющая местами два элемента, swap

print\_students – функция, выводящая всех студентов в цикле по шаблону вида:  
‘номер по списку’ ‘,’ surname = ‘фамилия студента’ \t ball = ‘средний балл i-го студента’

print\_students\_with\_most\_ex\_marks - функция, выводящая студентов с наибольшим количеством отличных оценок, которое определяется в ходе выполнения данной функции. Эта функция включает в себя number\_of\_ex\_mark - функцию, определяющую количество отличных оценок в заданной структуре студента.

deleting – функция, которая очищает память двум массивов указателей.

После выполнения всех этих операций, программа выводит на экран совокупность структур поэлементно, предварительно отсортированную по убыванию среднего балла, указанного в структуре, по шаблону вида

‘номер по списку’,surname = ‘фамилия студента’ \t ball = ‘средний балл i-го студента’

Следом она выводит фразу “Students with most excellent marks”, за которой следует список студентов, имеющих наибольшее количество отличных оценок, выведенных по шаблону вида

surname = ‘фамилия студента’ ‘оценка’ ‘оценка’ ‘оценка’ ‘оценка’ ‘оценка’

Алгоритм работы программы

1. Пользователь вводит число студентов до тех пор, пока не будет введено корректное целое положительное число.

2. Резервируется память для входных данных под введённое на шаге 1 количество студентов. После чего пользователь вводит информацию о студентах до тех пор, пока она не будет соответствовать шаблону текста задачи.

3. Резервируется память для массива структур под введённое на шаге 1 количество студентов и определяется средний балл каждого студента, записывая его в новый массив структур.

4. Происходит сортировка массива студентов методом пузырьковой сортировки по убыванию среднего балла, посчитанного на предыдущем шаге.

5. Выводится отсортированный массив студентов на экран в соответствии с шаблоном, указанным перед алгоритмом работы программы.

6. Выводится студентов, имеющих наибольшее количество отличных оценок после соответствующего обозначения путем вывода строки “Students with most excellent marks”.

7. Высвобождается зарезервированная память.

Для наглядности далее представлен алгоритм решения поставленной задачи в виде блок-схемы для основной части программы и блок-схем для функций, используемых в программе.

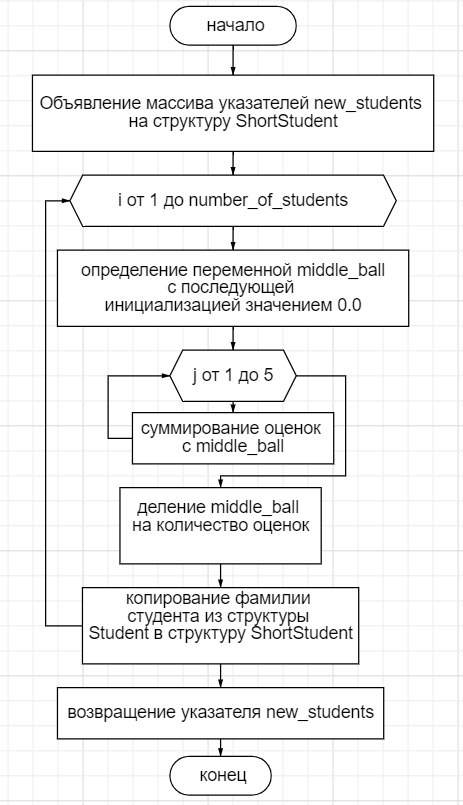
3. Описание структуры программы

3.1 Описание функции **main**



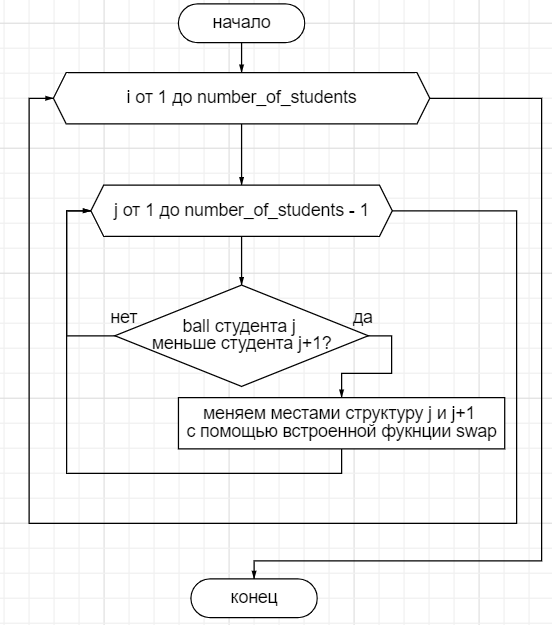
3.2 Описание фукнции **write\_middle\_ball\_to\_new\_srtucture**:

Функция считает средний балл студента из структуры одного массива и записывает в новую структуру другого массива



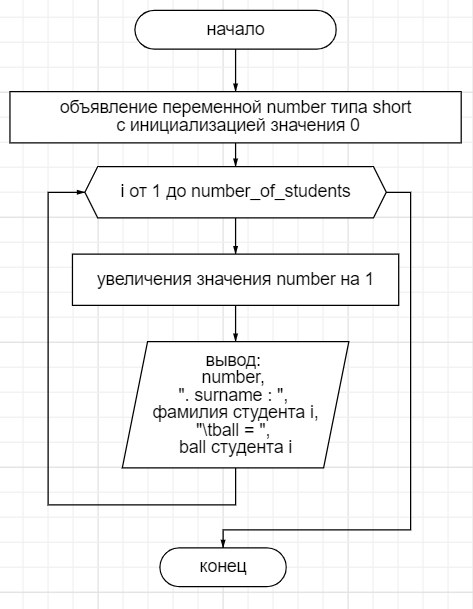
3.3 Описание фукнции **sorting\_structures**:

Фукнция сортирует массив структур, располагая по убыванию среднего балла студента



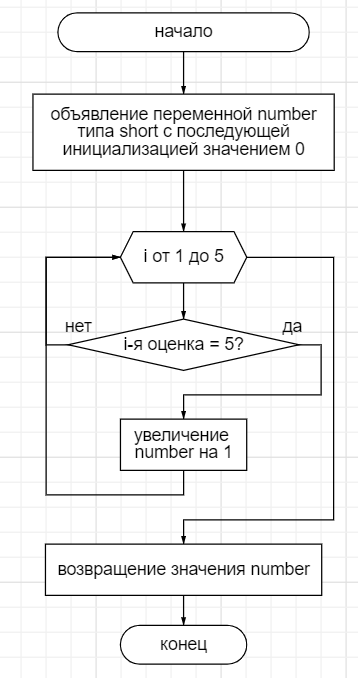
3.4 Описание функции **print\_students**

Функция выводит массив студентов по указанному в схеме вывода шаблону



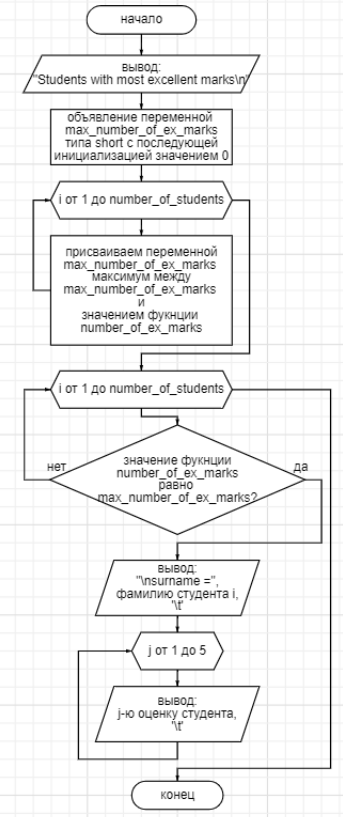
3.5 Описание функции **number\_of\_ex\_marks**:

Функция считает количество отличных оценок у студента, структуру которого она получает в качестве аргумента



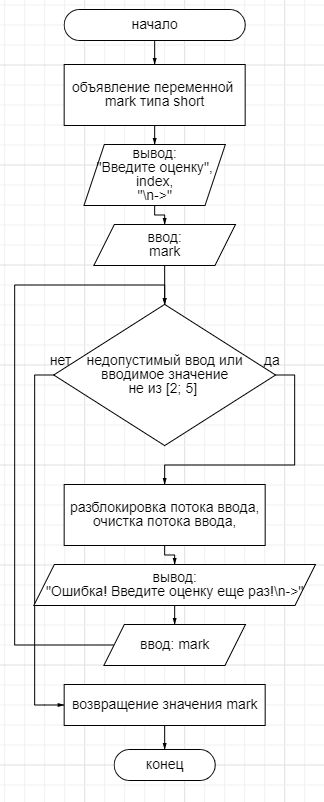
3.6 Описание функции **print\_students\_with\_most\_ex\_marks**:

Функция выводит по описанному в поле вывода шаблону студентов, имеющих наибольшее количество отличных оценок



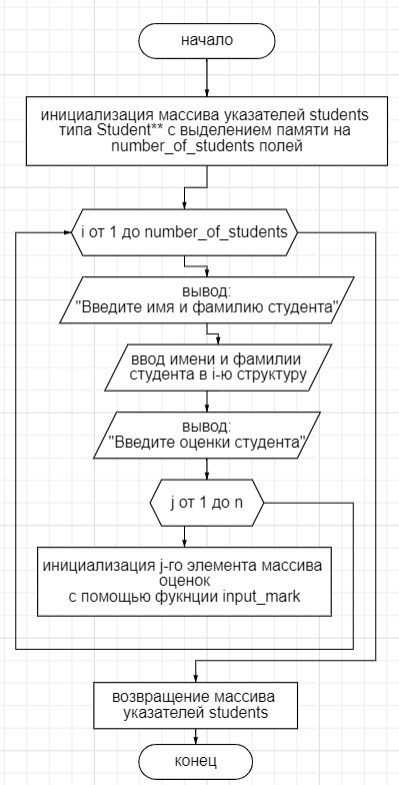
3.7 Описание функции **input\_mark:**

Функция обрабатывает вводимую оценку на предмет существования и соответствия федеральному стандарту оценивания в РФ на данный момент



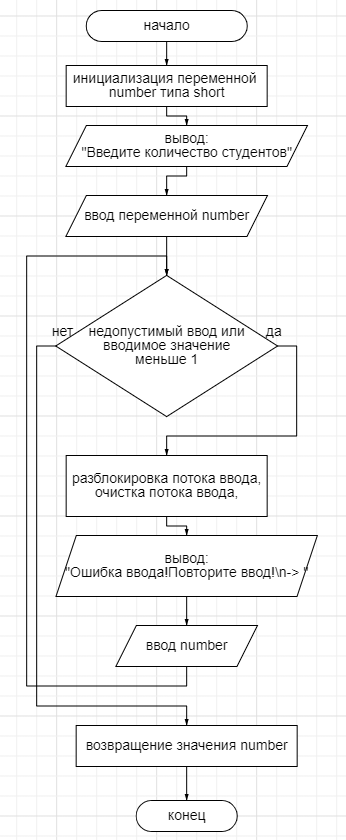
3.8 Описание фукнции **initialization\_students**:

Функция, которая записывает вводимые данные о студентах в массив



3.9 Описание фукнции **input\_number**:

Функция, которая определяет вводимое число, отвечающее за количество студентов на соответствие натуральному числу



4. Результаты тестирования

1) Входные данные:

3

Фиксик Нолик 4 5 5 5 4

Фиксик Симка 4 5 4 5 4

Дим Димыч 4 5 5 5 5

1)Выходные данные:

1. surname : Димыч ball = 4.8

2. surname : Нолик ball = 4.6

3. surname : Симка ball = 4.4

Students with most excellent marks

surname = Димыч 4 5 5 5 5

2)Входные данные

5

Фиксик Нолик 5 4 5 5 5

Фиксик Симка 4 4 5 4 4

Фикскик Верта 5 5 4 5 5

Фиксик Фаер 3 2 3 4 2

Дим Димыч 5 5 5 4 5

2) Выходные данные

1. surname : Нолик ball = 4.8

2. surname : Верта ball = 4.8

3. surname : Димыч ball = 4.8

4. surname : Симка ball = 4.2

5. surname : Фаер ball = 2.8

Students with most excellent marks

surname = Нолик 5 4 5 5 5

surname = Верта 5 5 4 5 5

surname = Димыч 5 5 5 4 5

5. Список использованных источников

1. Вирт Н. Алгоритмы и структуры данных / Н. Вирт. – М.: Мир, 2016. -360

2. Структуры в языке C++ [Электронный ресурс]- https://ravesli.com/urok-61-struktury/