МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

имени В.Ф. УТКИНА»

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ И ПРИКЛАДНОЙ МАТЕМАТИКИ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе по дисциплине

«АЛГОРИТМИЧЕСКИЕ ЯЗЫКИ И ПРОГРАММИРОВАНИЕ»

на тему

«РАБОТА СО СЛОВАРЯМИ»

Выполнил: студент гр. 245

Луковкин Иван Владимирович

Проверил: ст. преподаватель ВПМ

Москвитина Оксана Александровна

Рязань 2023

Оглавление

[Задание на курсовую работу: 3](#_Toc133202626)

[Введение 4](#_Toc133202627)

[Основная часть 5](#_Toc133202628)

[Анализ и математическая постановка задачи 5](#_Toc133202629)

[Разработка укрупненной структуры и детализированных блок - схем алгоритма и их описание 5](#_Toc133202630)

[Разработка программы, описание этапов разработки, назначение основных методов 6](#_Toc133202631)

[Анализ переменных 6](#_Toc133202632)

[Инструкция по использованию разработанной программы 9](#_Toc133202633)

[Результаты отладки 13](#_Toc133202634)

[Решение контрольных примеров и проверка правильности функционирования программы по результатам контрольных просчётов 14](#_Toc133202635)

[Заключение 16](#_Toc133202636)

[Библиографический список 17](#_Toc133202637)

[Приложение А. Детализация блока «Основное меню» 18](#_Toc133202638)

[Приложение Б. Детализация блока «Подсчёт очков» 19](#_Toc133202639)

[Приложение В. Детализация блока «Вывод таблицы» 20](#_Toc133202640)

[Приложение Г. Детализация блока «Добавление команд» 21](#_Toc133202641)

[Приложение Д. Детализация блока «Удаление команд». 22](#_Toc133202642)

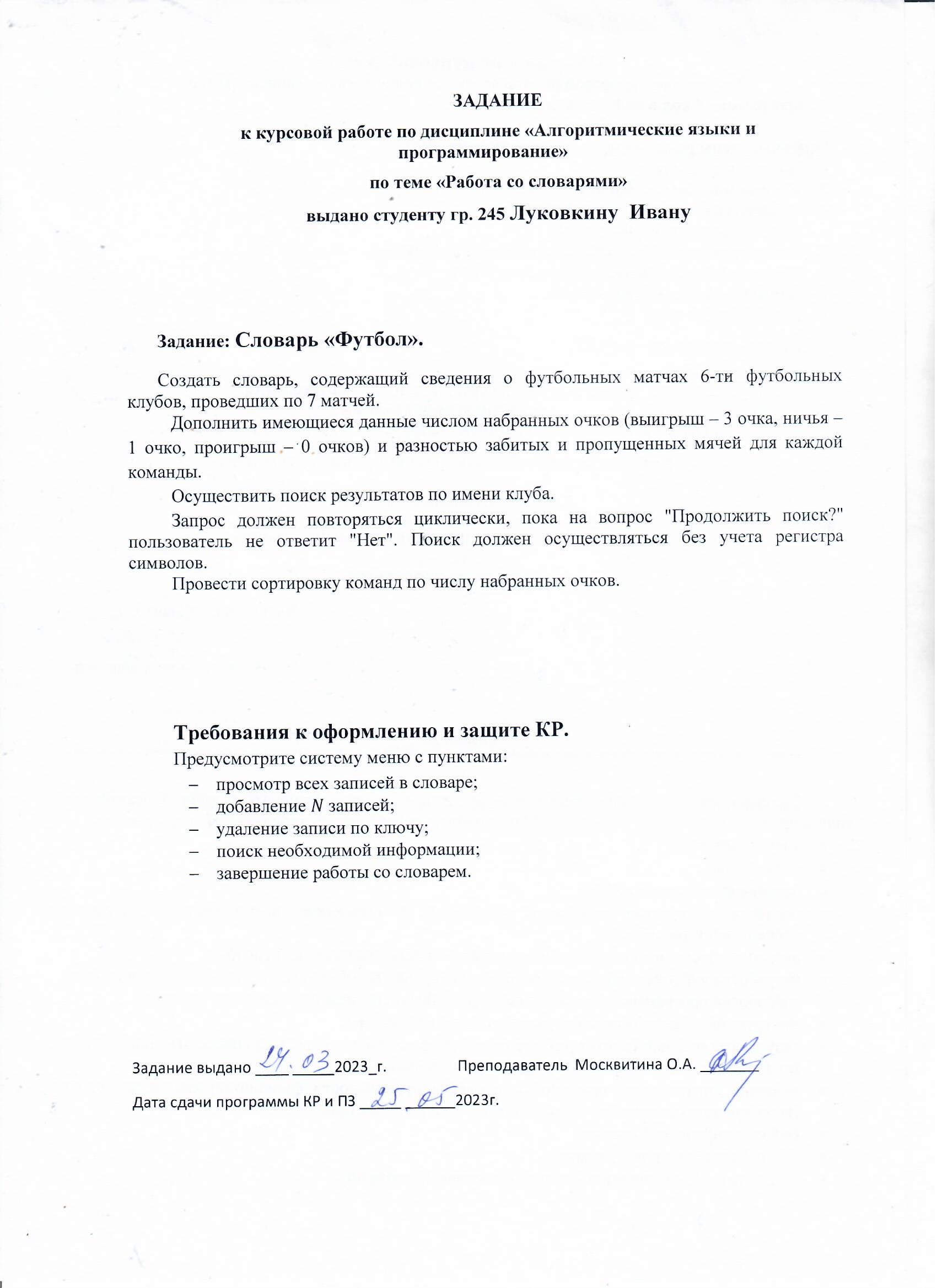
[Приложение Е. Детализация блока «Поиск информации» 23](#_Toc133202643)

[Приложение Ж. Детализация блока «Сортировка команд» 24](#_Toc133202644)

[Приложение З. Текст программы 25](#_Toc133202645)

[Приложение И. Содержимое Словаря 30](#_Toc133202646)

## Задание на курсовую работу:



# Введение

Последние несколько десятилетий с помощью вычислительной техники решается множество задач. Одной из немаловажных является хранение информации на носителях. Цель курсовой работы – получить опыт алгоритмизации, реализации программ для широкого применения на основе программирования информационной системы «Футбол». В качестве языка программирования был выбран язык “Python” в среде программирования PyCharm CE. Данный язык позволяет научиться основам алгоритмизации, реализовать алгоритм программы и создать текстовый интерфейс, что играет важную роль в данной курсовой работе.

Разработка информационной системы состоит из следующих этапов:

1. Анализ задания и математическая постановка задачи;
2. Разработка схемы алгоритма и ее описание;
3. Разработка программы и ее описание;
4. Инструкция по использованию разработанной программы;
5. Результаты отладки.

# Основная часть

## Анализ и математическая постановка задачи

Необходимо разработать программу информационной системы «Футбол». База данных, в которой будет содержаться информация о футбольных матчах, должна храниться в типизированных файлах. Исполнение алгоритма программы заключается в обработке данных о футбольных матчах.

Для реализации курсовой работы необходимо провести различные действия над элементами словаря. Для каждого действия создается собственная подпрограмма: процедуры создания и удаления элементов, добавления, анализа, сортировки, вывода таблиц на экран.

## Разработка укрупненной структуры и детализированных блок - схем алгоритма и их описание

По результатам анализа задания составляется схема алгоритма информационной системы «Футбол». (Рисунок 1)

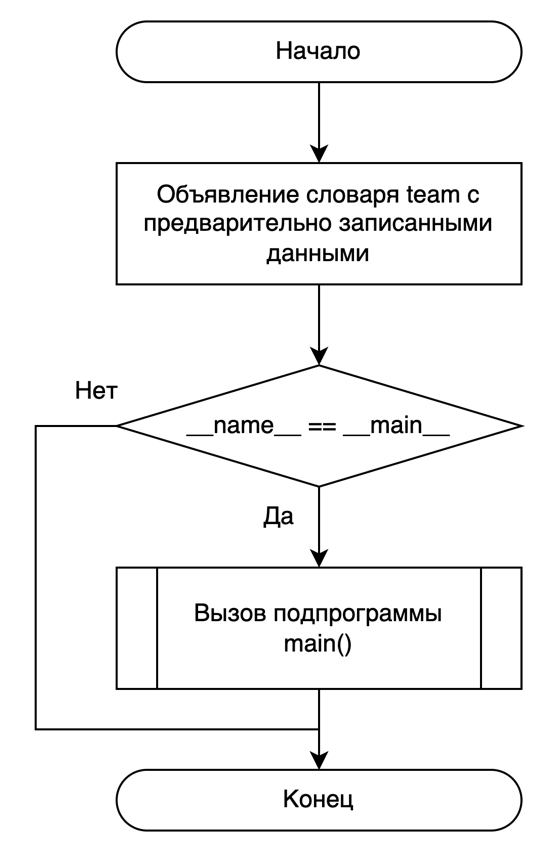


Рисунок 1 – Основная программа

I этап детализации

Детализация блока «Основное меню». (Приложение А)

II этап детализации

Детализация блока «Подсчёт очков». (Приложение Б)

Детализация блока «Вывод таблицы». (Приложение В)

Детализация блока «Добавление команд». (Приложение Г)

Детализация блока «Удаление команд». (Приложение Д)

Детализация блока «Поиск информации». (Приложение Е)

Детализация блока «Сортировка команд». (Приложение Ж)

## Разработка программы, описание этапов разработки, назначение основных методов

По результатам анализа необходимо определить пример вывода словаря (Таблица 1).

Таблица 1 – Пример вывода словаря

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Название команды | 1 Матч | 2 Матч | 3 Матч | 4 Матч | 5 Матч | 6 Матч | 7 Матч |
| Club1 | 0 | 3 | 1 | 1 | 3 | 0 | 0 |
| Club2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| Club3 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| Club4 | 3 | 0 | 1 | 1 | 3 | 3 | 0 |
| Club5 | 1 | 1 | 1 | 3 | 0 | 1 | 3 |
| Club6 | 1 | 1 | 3 | 0 | 3 | 3 | 3 |

## Анализ переменных

team *–* Словарь, содержащий в себе ключ (название команды) и значение (массив данных, включающий в себя целочисленные очки, набранные командой за турнир; глобальная переменная).

**Процедура calc\_score**

matches – массив, хранящий в себе результаты матчей команды;

score – число набранных очков;

goals\_scored – число голов;

goals\_conceded – число пропущенных мячей;

match – элемент массива, хранящий о себе результат конкретного матча;

goal\_difference – разница между забитыми и пропущенными голами.

**Процедура print\_team**

data - обрабатываемый словарь;

table – таблица, использующаяся для корректного вывода данных;

headers - массив, хранящий в себе названия заголовков таблиц;

teams - массив, хранящий в себе названия команд;

matches - массив, хранящий в себе результаты матчей;

mass - массив результатов текущего матча. Используется для заполнения таблицы столбцами.

**Процедура add\_team**

team - обрабатываемый словарь;

n - число записей, необходимых для дополнения словаря;

club - название добавляемого клуба;

matches - массив, заполняющийся результатами матчей;

match - вводимый пользователь результат матча.

**Процедура del\_team**

team - обрабатываемый словарь;

club - названиеудаляемой команды;

**Процедура search\_team**

team - обрабатываемый словарь;

club - название искомого клуба;

score, goal\_difference - переменные, получаемые в результате работы процедуры *calc\_score*, описанной ранее;

choice - вводимый пользователем ответ на вопрос "Продолжить поиск?"

**Процедура sort\_team**

data - обрабатываемый словарь;

scores - массив, хранящий в себе число очков, набранных командой

score, goal\_difference - переменные, получаемые в результате работы процедуры *calc\_score*, описанной ранее;

dictionary - словарь, хранящий в себе ключ (название команды) и значение (число набранных очков)

sorted\_dict - отсортированный словарь dictionary

table - таблица, используемая для оформления удобного для пользователя вывода результатов

last\_key, last\_value - последние ключ и значение соответственно, являющиеся названием победителя и числом набранных очков

**Процедура main**

choice - выбор номера действия

**Общие описания блок-схемы основных подпрограмм**

**Процедура calc\_score** -данная процедура осуществляет подсчет набранных командой очков и разницу между забитыми и пропущенными мячами. Входные данные - обрабатываемый словарь. Выходные данные - число очков, разность голов и пропусков мячей.

**Процедура print\_team** - данная процедура осуществляет комфортный для пользователя вывод промежуточных результатов всех команд. Входные данные - обрабатываемый словарь. Выходные данные отсутствуют.

**Процедура add\_team** - данная процедура осуществляет добавление в исходный словарь новые данные о вводимых командах. Входные данные - обрабатываемый словарь. Выходные данные отсутствуют.

**Процедура del\_team** - данная процедура осуществляет удаление выбранной по желанию пользователя команды. Входные данные - обрабатываемый словарь. Выходные данные отсутствуют.

**Процедура sort\_team** - данная процедура осуществляет сортировку всех команд по общему числу набранных очков и выводит таблицу, где отображены итоговые баллы. Входные данные - обрабатываемый словарь. Выходные данные отсутствуют.

**Процедура main** - данная процедура осуществляет выбор пользователем действий, включающих в себя процедуры, описанные ранее.

## Инструкция по использованию разработанной программы

Системные требования:

1. Система – Windows XP/Vista/7/10, macOS 10.13.0 и выше;
2. Процессор – Intel Pentium II, III class 2327 и выше;
3. ОЗУ – 256 Мб и выше;
4. Память – не менее 1 Мб свободного места на диске;
5. Наличие клавиатуры обязательно; компьютерная мышь по желанию;
6. Экран – ЖК монитор с соотношением сторон 1920 x 1080 (в случае более низкого разрешения программа может некорректно выводить список книг).

Тестирование программы проводилось на операционной системе macOS Ventura 12.6.7 на ЖК мониторе с соотношением сторон 1366 x 768.

Для работы программы на компьютере должен быть предварительно установлен интерпретатор Python не ниже 3 версии для стабильной работы всех используемых функций и библиотек.

Наличие ранее подготовленного словаря не обязательно, т.к. в программе заранее имеется подготовленный словарь данных, с которым пользователь может работать.

Отладка и тестирование программы проводились в среде PyCharm CE, но программа может быть использована как и в другом IDE, так и при работе в командной строке операционной системы.

В главном меню путём выбора предложенных пунктов пользователь может ввести номер необходимого действия.

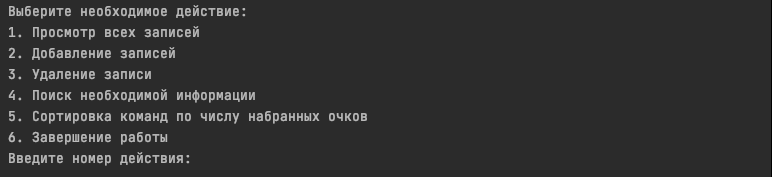


Рисунок 2 - Главное меню программы

Если был выбран номер, не входящий в список предложенных, программа выведет сообщение о неверном вводе и предложит выбрать действие снова (Рисунок 3).

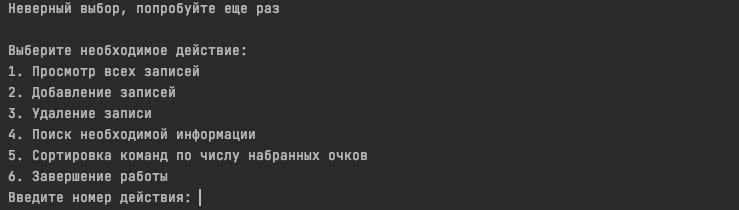


Рисунок 3 - Результат неверного ввода

При выборе первого действия программа отобразит таблицу промежуточных результатов и предложит снова выбрать действие из главного меню (Рисунок 4).

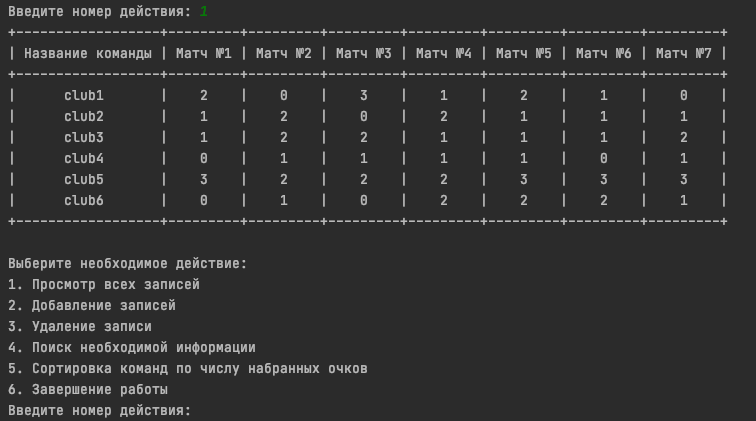


Рисунок 4 - Выполнение 1-го действия

При выборе второго действия программа будет выполнять процедуру добавления новой записи в словарь. Здесь будет предложено ввести число записей, название команды и промежуточные результаты матчей (Рисунок 5).

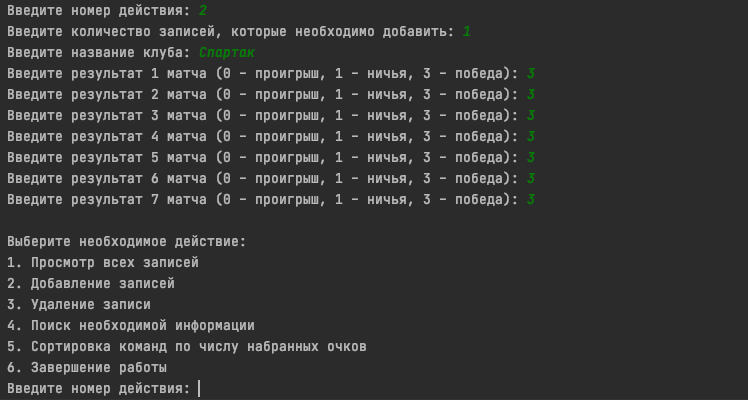


Рисунок 5 - Выполнение 2-го действия

Результат верного ввода новых данных показан на рисунке 6.

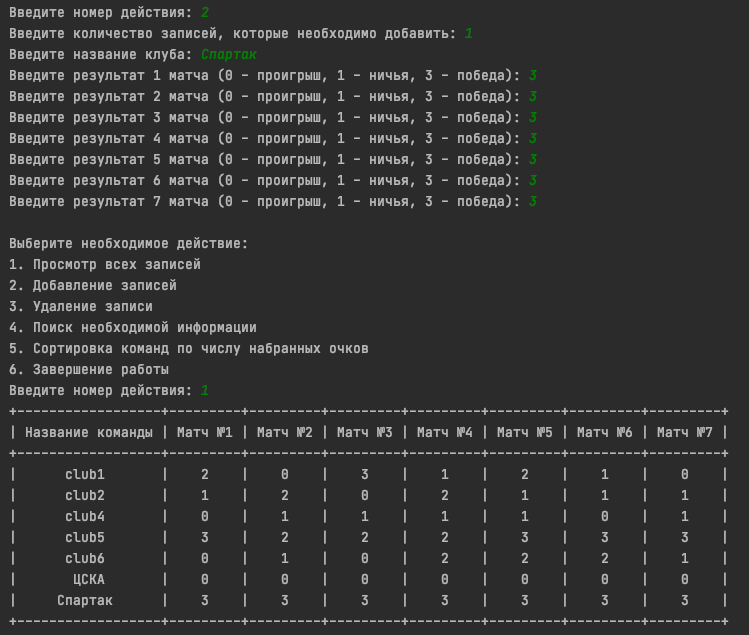


Рисунок 6 - Добавление новой записи и проверка действия

При вводе неверного значения в поле "количество "записей", команда сообщит, что значение не является верным и вернется снова к вводу числа (Рисунок 7).

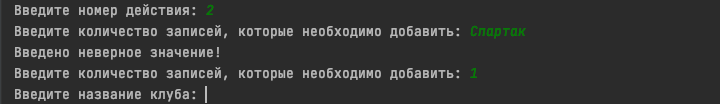


Рисунок 7 - Неверный ввод во втором действии

При выполнении третьего действия программа предложит ввести название команды, которое необходимо удалить. Название команды является ключом в словаре (Рисунок 8).

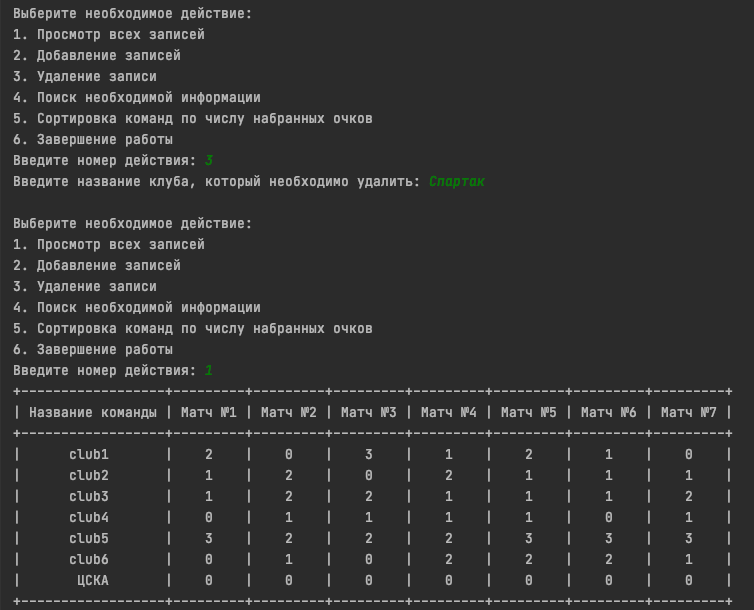


Рисунок 8 - Выполнение 3-го действии

При неверном вводе программа отобразит сообщение об ошибке и вернется к вводу снова (аналогично второму действию). Результат выполнения представлен на рисунке 9.

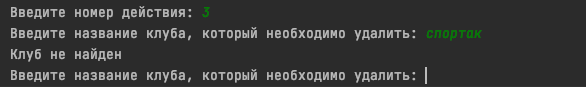


Рисунок 9 - Неверный ввод в 3-ем действии

При выполнении действия №4 Программа предложит ввести название клуба, о котором нужно узнать статистику. Аналогично предыдущим пунктам при неверном вводе команда предложит ввести данные заново. Все возможные варианты представлены на рисунке 10.

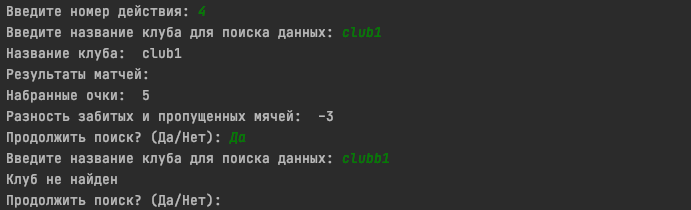


Рисунок 10 - Выполнение 4-го действия

При выполнении 5-го действия от пользователя не требуется ввода данных. Все результаты представляются сразу (Рисунок 11).

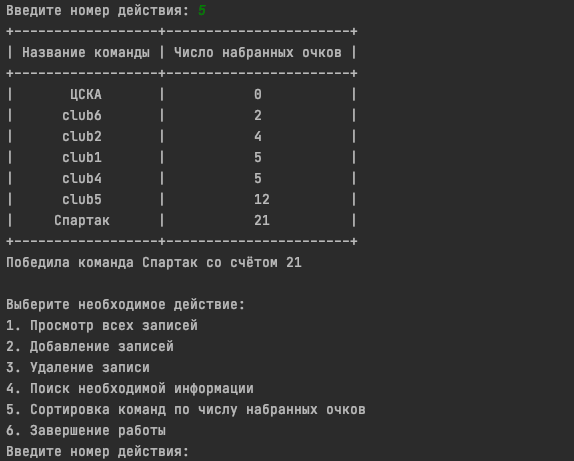


Рисунок 11 - Результат выполнения 5-го действия

При выполнении 6-го действия пользователю будет выведен текст с прощанием. После этого скрипт завершит свою работу (Рисунок 12).

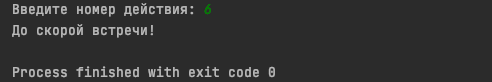


Рисунок 12 - Завершение работы

## Результаты отладки

При использовании встроенной в PyCharm функции *Debug* были получены результаты отладки, которая была пройдена успешно. Результаты отладки показаны на рисунке 13.



Рисунок 13 - Результаты отладки



Рисунок 14 - Окно проблем проекта

## Решение контрольных примеров и проверка правильности функционирования программы по результатам контрольных просчётов

Решением контрольных примеров является выполнение всех действий, предложенных в программе. Результат выполнения и проверка функционирования представлены ниже.

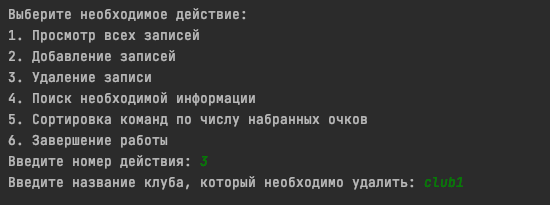


Рисунок 15 - Удаление записи

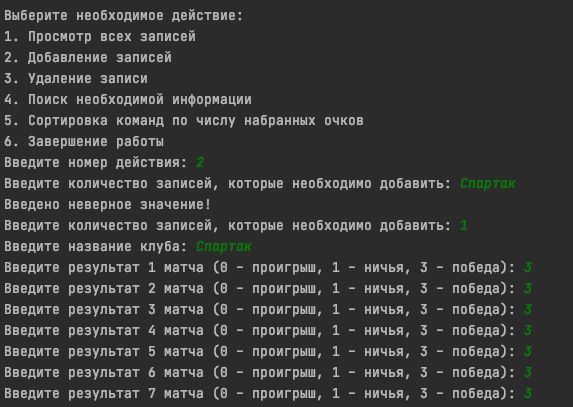


Рисунок 16 - Добавление записи

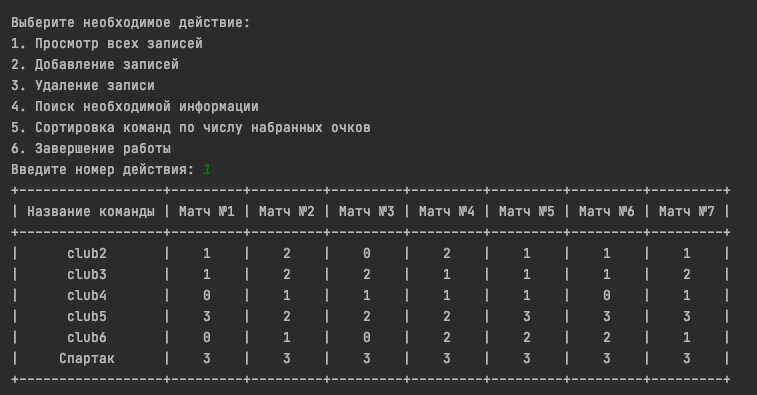


Рисунок 17 - Просмотр записей

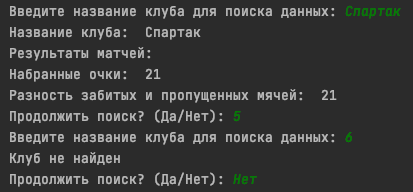


Рисунок 18 - Поиск информации

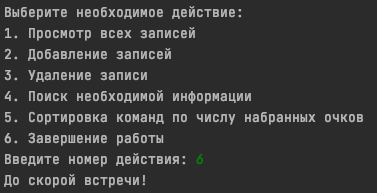


Рисунок 19 - Завершение работы

# Заключение

За время выполнении данной курсовой работы были пройдены все этапы разработки программного обеспечения:

1. Постановка задачи;
2. Формализация задачи, определение входной и выходной информации;
3. Сбор необходимых исходных данных, используемых в программе;
4. Определение необходимых требований к техническим и программным средствам для функционирования приложения;
5. Составление логической структуры решения задачи и программы;
6. Устранение ошибок;
7. Составление инструкции пользователя.

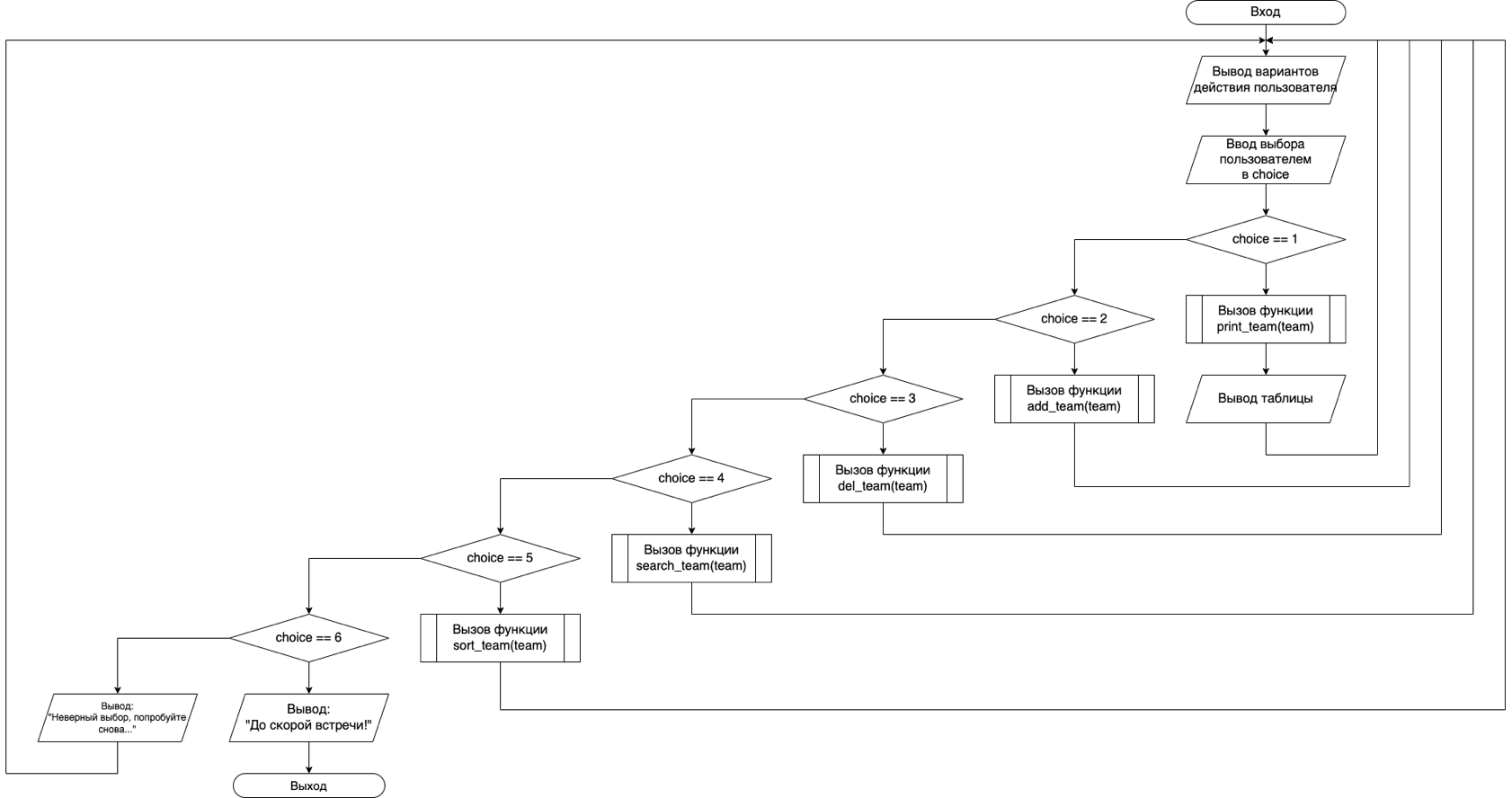
При работе над курсовой работой были освоены принципы разработки алгоритмов решения задач, правила составления блок-схем алгоритмов, закреплены процессы отладки и выполнения программ с использованием среды программирования PyCharm CE.

Цель курсовой работы была достигнута. Реализованное программное приложение полностью соответствует поставленным задачам, определенным в задании на курсовую работу.

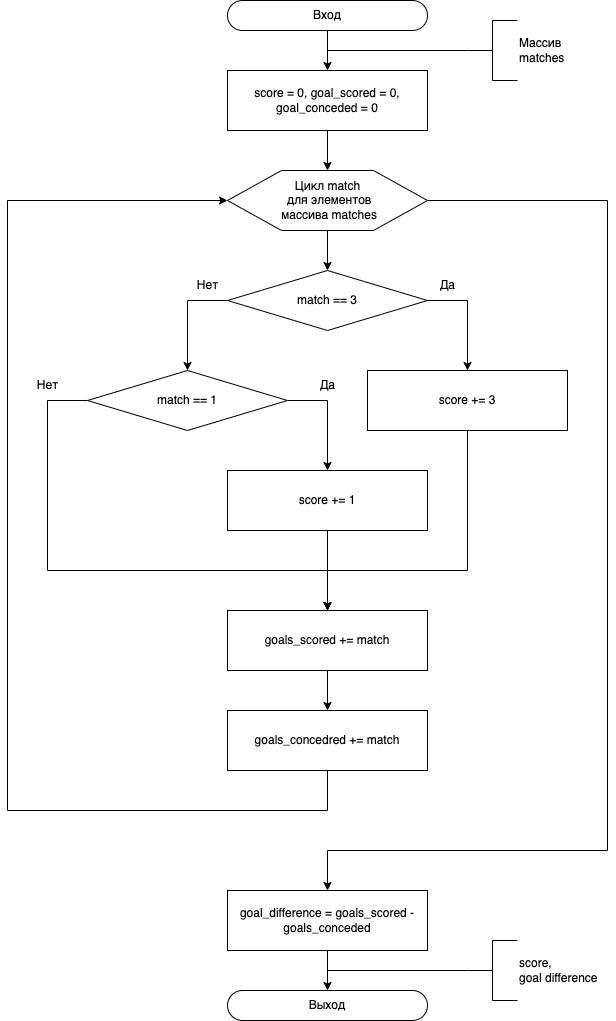
# Библиографический список

1. Примеры и задачи по программированию на Паскале и Питоне. Фонд оценочных средств для промежуточных аттестаций: учеб. пособие / О.А. Москвитина, А.Н. Пылькин – М.: КУРС, 2018. – 192 с.
2. *Лутц М.* Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
3. З*латопольский Д.М.* Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.***Федоров, Д. Ю.* Программирование на языке высокого уровня Python** : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2019. – 161 с. – (Бакалавр. Прикладной курс). – ISBN 978-5-534-10971-9. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: <https://urait.ru/bcode/437489> (дата обращения: 13.02.2020).

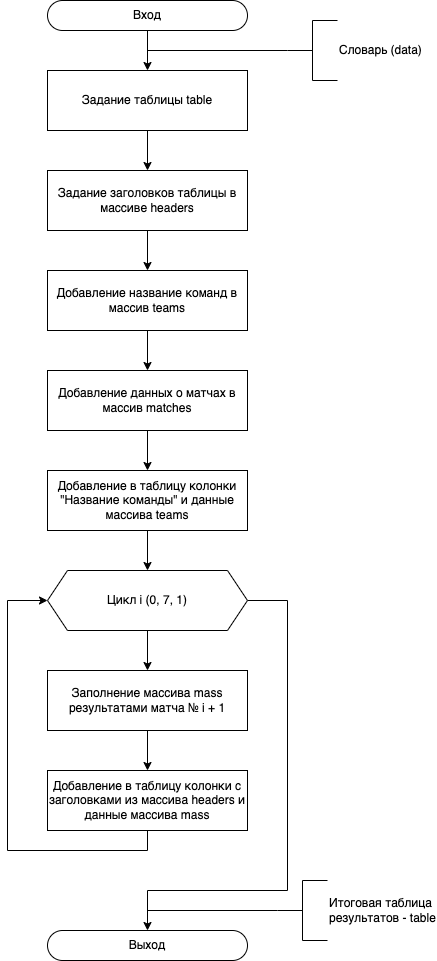
# Приложение А. Детализация блока «Основное меню»



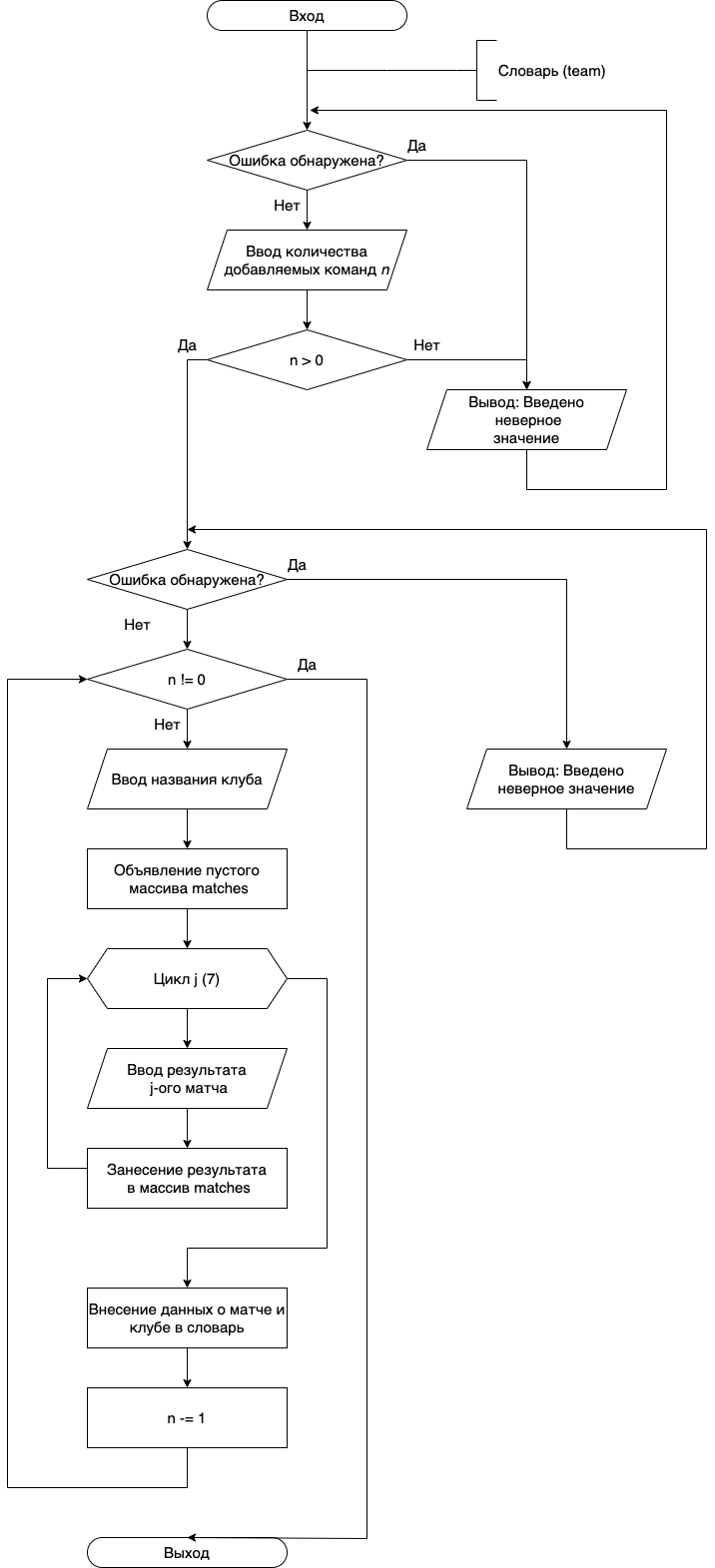
# Приложение Б. Детализация блока «Подсчёт очков»



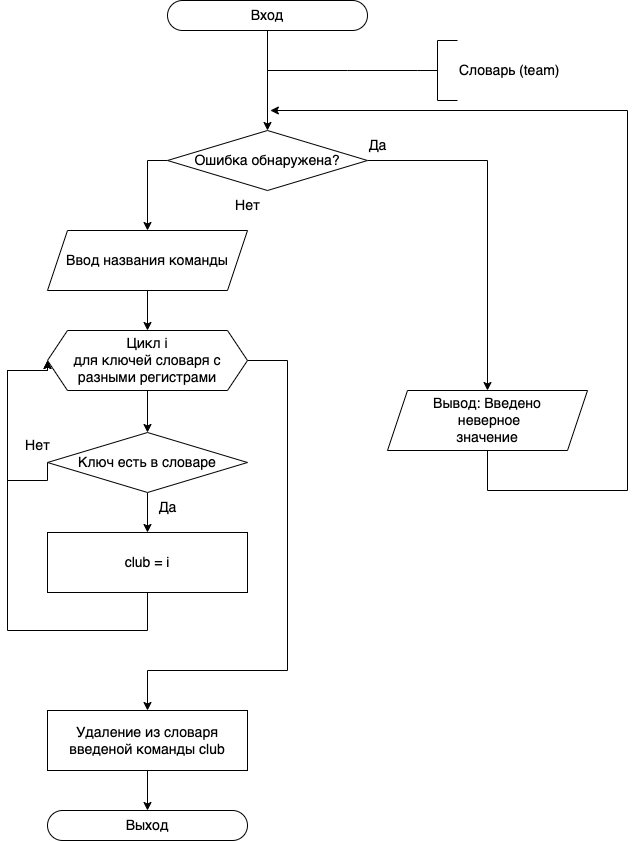
# Приложение В. Детализация блока «Вывод таблицы»



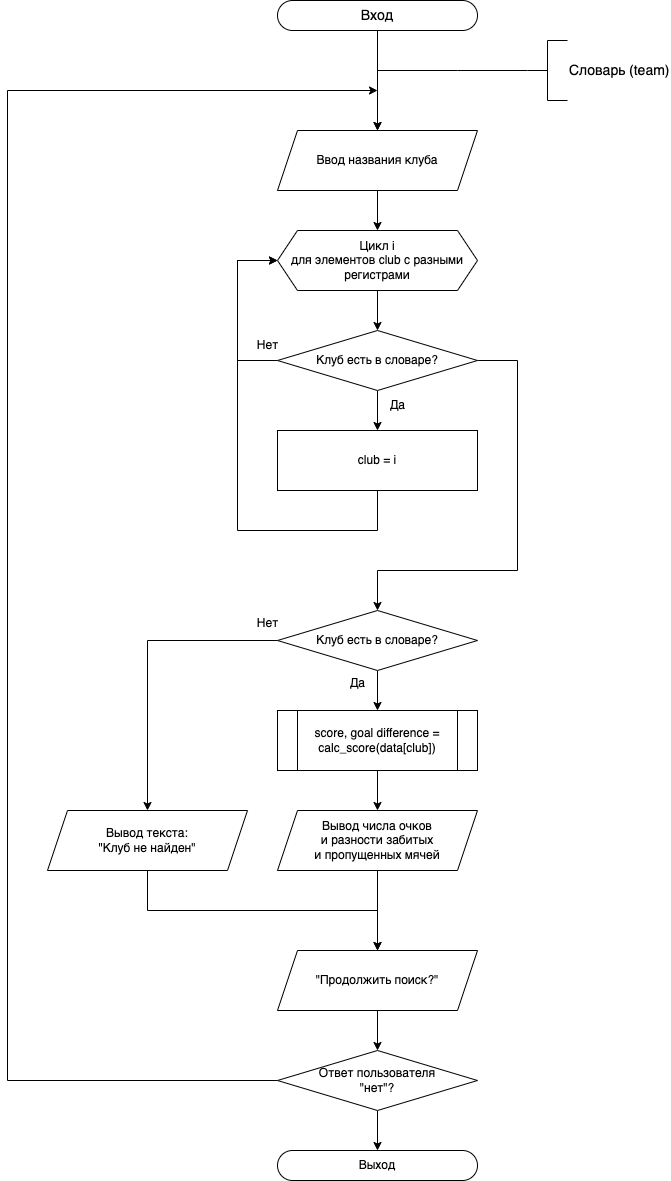
# Приложение Г. Детализация блока «Добавление команд»



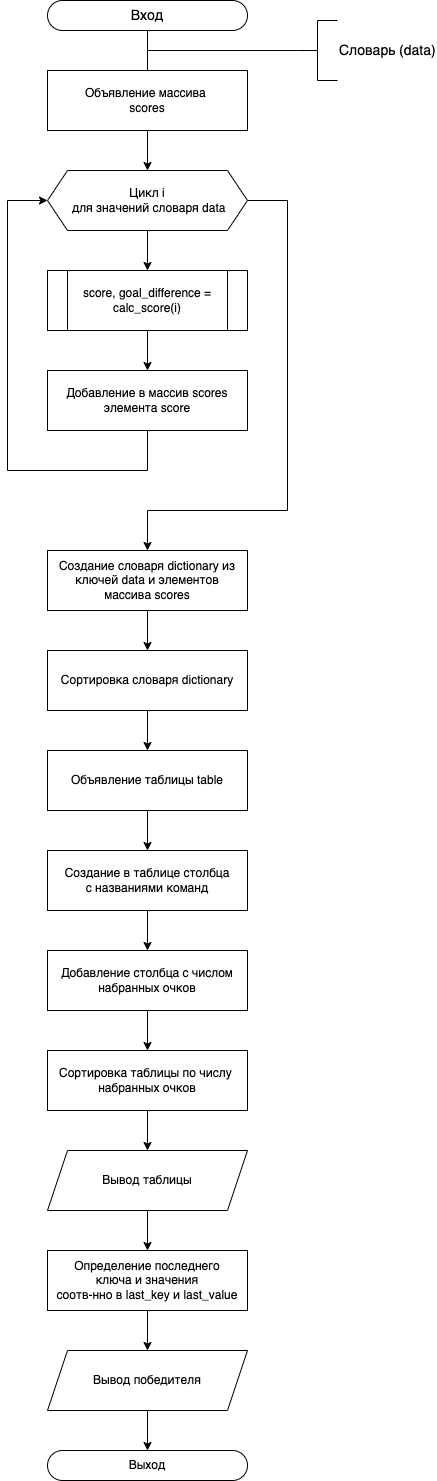
# Приложение Д. Детализация блока «Удаление команд».



# Приложение Е. Детализация блока «Поиск информации»



# Приложение Ж. Детализация блока «Сортировка команд»



# Приложение З. Текст программы

import prettytable as pt

team = {'club1': [2, 0, 3, 1, 2, 1, 0],

'club2': [1, 2, 0, 2, 1, 1, 1],

'club3': [1, 2, 2, 1, 1, 1, 2],

'club4': [0, 1, 1, 1, 1, 0, 1],

'club5': [3, 2, 2, 2, 3, 3, 3],

'club6': [0, 1, 0, 2, 2, 2, 1]}

def calc\_score(matches):

score = 0

goals\_scored = 0

goals\_conceded = 0

for match in matches:

if match == 3:

score += 3

elif match == 1:

score += 1

goals\_scored += match

goals\_conceded += 3 - match

goal\_difference = goals\_scored - goals\_conceded

return score, goal\_difference

def print\_team(data):

table = pt.PrettyTable()

headers = ['Матч №1', 'Матч №2', 'Матч №3', 'Матч №4', 'Матч №5', 'Матч №6', 'Матч №7']

teams = [i for i in data.keys()]

matches = [i for i in data.values()]

table.add\_column('Название команды', teams)

for i in range(0, 7):

mass = [matches[j][i] for j in range(0, len(matches))]

table.add\_column(headers[i], mass)

return table

def add\_team(team):

while True:

try:

n = int(input('Введите количество записей, которые необходимо добавить: '))

if n > 0:

break

except ValueError:

print('Введено неверное значение!')

while True and n != 0:

try:

club = input('Введите название клуба: ')

matches = []

for j in range(7):

match = int(input('Введите результат {} матча (0 - проигрыш, 1 - ничья, 3 - победа): '.format(j+1)))

matches.append(match)

team[club] = matches

n -= 1

except ValueError:

print('Введено неверное значение\nПожалуйста, повторите ввод...')

def del\_team(team):

while True:

try:

club = input('Введите название клуба, который необходимо удалить: ')

for i in club.lower(), club.upper(), club:

if i in team:

club = i

del team[club]

break

except KeyError:

print('Клуб не найден')

def search\_team(team):

while True:

club = input('Введите название клуба для поиска данных: ')

for i in club.lower(), club.upper(), club:

if i in team:

club = i

if club in team:

score, goal\_difference = calc\_score(team[club])

print('Название клуба: ', club,

'\nРезультаты матчей: ',

'\nНабранные очки: ', score,

'\nРазность забитых и пропущенных мячей: ', goal\_difference)

else:

print('Клуб не найден')

choice = input('Продолжить поиск? (Да/Нет): ')

if choice.lower() == 'нет':

break

def sort\_team(data):

scores = []

for i in data.values():

score, goal\_difference = calc\_score(i)

scores.append(score)

dictonary = dict(zip(data.keys(), scores))

sorted\_dict = dict(sorted(dictonary.items(), key=lambda item: item[1]))

table = pt.PrettyTable()

table.add\_column('Название команды', [i for i in data.keys()])

table.add\_column('Число набранных очков', scores)

table.sortby = 'Число набранных очков'

print(table)

last\_key = list(sorted\_dict.keys())[-1]

last\_value = list(sorted\_dict.values())[-1]

print(f'Победила команда {last\_key} со счётом {last\_value}')

def main():

while True:

print(

'\nВыберите необходимое действие:\n'

'1. Просмотр всех записей\n'

'2. Добавление записей\n'

'3. Удаление записи\n'

'4. Поиск необходимой информации\n'

'5. Сортировка команд по числу набранных очков\n'

'6. Завершение работы'

)

choice = int(input('Введите номер действия: '))

if choice == 1:

print(print\_team(team))

elif choice == 2:

add\_team(team)

elif choice == 3:

del\_team(team)

elif choice == 4:

search\_team(team)

elif choice == 5:

sort\_team(team)

elif choice == 6:

print('До скорой встречи!')

break

else:

print('Неверный выбор, попробуйте еще раз')

if \_\_name\_\_ == '\_\_main\_\_':

main()

# Приложение И. Содержимое Словаря

teams = {

'club1': [2, 0, 3, 1, 2, 1, 0],

'club2': [1, 3, 3, 2, 1, 1, 1],

'club3': [1, 2, 2, 1, 1, 1, 2],

'club4': [0, 1, 1, 1, 1, 0, 1],

'club5': [3, 2, 2, 2, 3, 3, 3],

'club6': [0, 3, 0, 2, 2, 2, 1],

'club7': [1, 2, 0, 2, 1, 1, 1],

'club8': [1, 2, 2, 1, 1, 1, 2],

'club9': [0, 1, 1, 1, 1, 0, 1],

'club10': [3, 2, 2, 2, 3, 3, 3],

'club11': [1, 2, 0, 2, 0, 0, 1],

'club12': [1, 0, 3, 1, 1, 1, 2],

'club13': [0, 1, 1, 1, 1, 0, 1],

'club14': [3, 2, 2, 2, 3, 3, 3],

'club15': [0, 1, 0, 2, 2, 0, 1],

'club16': [1, 2, 0, 2, 1, 1, 1],

'club17': [0, 1, 1, 1, 1, 0, 1],

'club18': [3, 2, 2, 2, 3, 3, 3],

'club19': [0, 1, 0, 2, 2, 2, 1],

'club20': [3, 3, 3, 3, 3, 3, 3],

}