**4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ**

В этом разделе рассматривается схема алгоритма, а также описываются другие используемые в программе алгоритмы

**4.1 Схема алгоритма**

Схема алгоритма составления кодировок представлена в Приложении В.

**4.2 Алгоритмы по шагам**

**4.2.1 Алгоритм записи в архив**

Для алгоритма по шагам рассмотрен метод getNextCodeBuffer класса Coder

1. Начало
2. Входные данные: QString src, содержащая информацию из исходного файла, словарь кодировок QMap<char, QString> dictionary, где содержится битовая кодировка каждого встречающегося в исходном файле байта, FileWriter fw, объект класса, который производит запись буферов в файл, переменная bool leftPrev, которая указывает, остался ли незаполненный байт с предыдущего вызова метода, short prevSize, который содержит длину битовой записи в незаконченном байте, char prev, где содержится незаконченный байт
3. Начинается итерация по строке src
4. Проверяется переменная leftPrev, если она true, то для работы берется незаконченный байт prev, если false, то новый
5. Итерация по строке с информацией src, каждый символ из нее ставится в соответствие со словарем кодировок dictionary, строка с кодировкой из словаря передается методу QString encode(QString codeStr), который вернет строку закодированных байт
6. Байты, полученные из метода encode, дополняют переменную prev, которая затем попадает в буффер, если выставлен флаг leftPrev, иначе сразу записываются в выходной буфер QString buffer и, если число бит кодировки байта не было кратно 8, то выставляется флаг leftPrev, размер незаконченного байта записывается в prevSize, а кодировка в переменную prev
7. Переход к шагу 6 пока не закончатся символы в строке src
8. Когда все байты исходной строки src закодированы, buffer передается в метод void write(QString buffer) класса FileWriter
9. Конец