**Отчет по лабораторной работе: Реализация кодирования Шеннона-Фано и Хаффмана**

**Класс Form1**

**Описание:** Основной класс формы, реализует пользовательский интерфейс и связывает его с функциональностью алгоритмов.

| **Метод** | **Описание** | **Аргументы** | **Возвращаемое значение** |
| --- | --- | --- | --- |
| Form1() | Конструктор, инициализирует компоненты формы и добавляет столбцы в DataGridView. | Нет | Нет |
| button1\_Click | Реализует логику обработки текста: вычисление вероятностей, кодирование (Шеннон-Фано, Хаффман), вывод результатов в UI. | object sender, EventArgs e | Нет |
| button2\_Click | Загружает текст из файла и обновляет текстовое поле. | object sender, EventArgs e | Нет |
| f | Вычисляет энтропию для заданной вероятности. | double x | double (результат) |

**Класс ShannonFanoEncoder**

**Описание:** Реализует алгоритм кодирования Шеннона-Фано.

| **Метод** | **Описание** | **Аргументы** | **Возвращаемое значение** |
| --- | --- | --- | --- |
| ShannonFanoEncoder(double[] probabilities) | Конструктор, принимает вероятности и инициализирует внутренние структуры. | double[] probabilities | Нет |
| Encode() | Выполняет кодирование последовательности по алгоритму Шеннона-Фано. | Нет | string[] (коды символов) |
| SplitSequence(int L, int R) | Находит точку разделения последовательности вероятностей для минимизации разницы. | int L, int R | int (индекс разделения) |
| Fano(int L, int R) | Рекурсивно строит коды Шеннона-Фано, добавляя биты. | int L, int R | Нет |

**Класс HuffmanTree**

**Описание:** Реализует алгоритм кодирования Хаффмана.

| **Метод** | **Описание** | **Аргументы** | **Возвращаемое значение** |
| --- | --- | --- | --- |
| HuffmanTree() | Конструктор, инициализирует структуру дерева. | Нет | Нет |
| Build(Dictionary<char, double> frequencies) | Строит дерево Хаффмана на основе частот символов. | Dictionary<char, double> frequencies | Нет |
| ReturnAlphabet() | Возвращает коды символов, сгенерированные деревом Хаффмана. | Нет | List<string> (коды символов) |

**Пример использования методов**

1. **Алгоритм Шеннона-Фано:**
2. ShannonFanoEncoder shannonFanoEncoder = new ShannonFanoEncoder(probabilities.Values.ToArray());
3. string[] shannonCodes = shannonFanoEncoder.Encode();
4. **Алгоритм Хаффмана:**
5. HuffmanTree huffmanTree = new HuffmanTree();
6. huffmanTree.Build(probabilities);
7. List<string> huffmanCodes = huffmanTree.ReturnAlphabet();
8. **Расчет энтропии:**
9. double entropy = probabilities.Values.Sum(p => f(p));

**Общий вывод**

Программа успешно реализует два алгоритма кодирования (Шеннона-Фано и Хаффмана) и вычисляет их характеристики, такие как средняя длина кода, энтропия и избыточность. Код модульный и хорошо структурирован, что позволяет легко добавлять новые функции и изменять существующие.