

УТВЕРЖДАЮ:

\_\_\_\_\_ Галкин В. А.

«\_\_» \_\_\_\_\_ 2017 г.

Описание программы  
к курсовой работе  
«Локальная безадаптерная сеть»  
(вариант № 26а)  
по курсу «Сетевые технологии в АСОИУ»

ИСПОЛНИТЕЛИ:

\_\_\_\_\_ Лецев А. О., ИУ5-64

\_\_\_\_\_ Мельников К. И., ИУ5-64

«\_\_» \_\_\_\_\_ 2017 г.

Москва — 2017 г.

---

# Содержание

1. Общие сведения .....	3
2. Назначение разработки .....	3
3. Описание логической структуры .....	3
3.1. Алгоритмы передачи файла .....	3
3.1.1. Алгоритм выбора папки .....	3
3.1.2. Алгоритм начала отправки файла.....	4
3.1.3. Алгоритм начала получения файла .....	5
3.1.4. Алгоритм отправки части файла.....	5
3.1.5. Алгоритм получения части файла.....	6
3.1.6. Алгоритм завершения передачи файла .....	6
4. Используемые технические средства.....	6
5. Входные и выходные данные .....	7
5.1. Входные данные .....	7
5.2. Выходные данные.....	7
6. Спецификация данных.....	7
6.1. Формат кадра .....	7
6.2. Форматы пакетов .....	7
6.2.1. Пакет данных о файле.....	7
6.2.2. Пакет неготовности к получению файла .....	7
6.2.3. Пакет запроса части файла .....	7
6.2.4. Пакет части файла .....	8
6.2.5. Пакет квитанции о получении файла .....	8
6.2.6. Пакет квитанции о завершении передачи .....	8
6.2.7. Пакет разрыва соединения.....	8
6.2.8. Пакет квитанции о разрыве соединения .....	8
7. Спецификация функций и классов .....	8

# 1. Общие сведения

Наименование программы: «Локальная безадаптерная сеть». Основанием для разработки является учебный план кафедры ИУ5 «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана на 6 семестр. Исполнителями являются студенты МГТУ им. Н.Э. Баумана группы ИУ5-64 Лещев Артем Олегович и Мельников Константин Игоревич. Программа выполнена на языке программирования C# и работает под управлением операционной системы Microsoft Windows версии 10 или выше.

# 2. Назначение разработки

Программа позволяет передавать файлы по локальной сети с возможностью докачки после восстановления прерванной связи, состоящей из двух персональных компьютеров, соединённых через интерфейс RS-232C нуль-модемным кабелем.

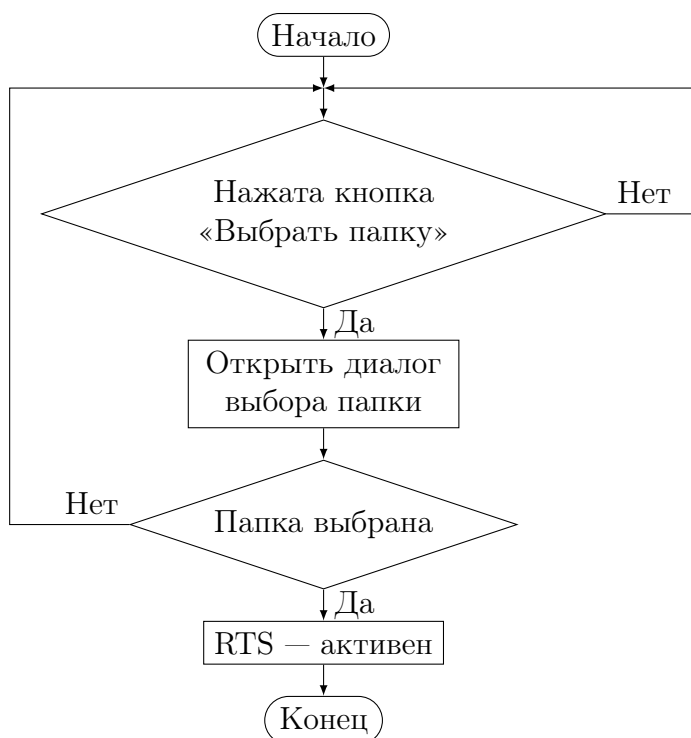
# 3. Описание логической структуры

Логически программа состоит из трёх уровней:

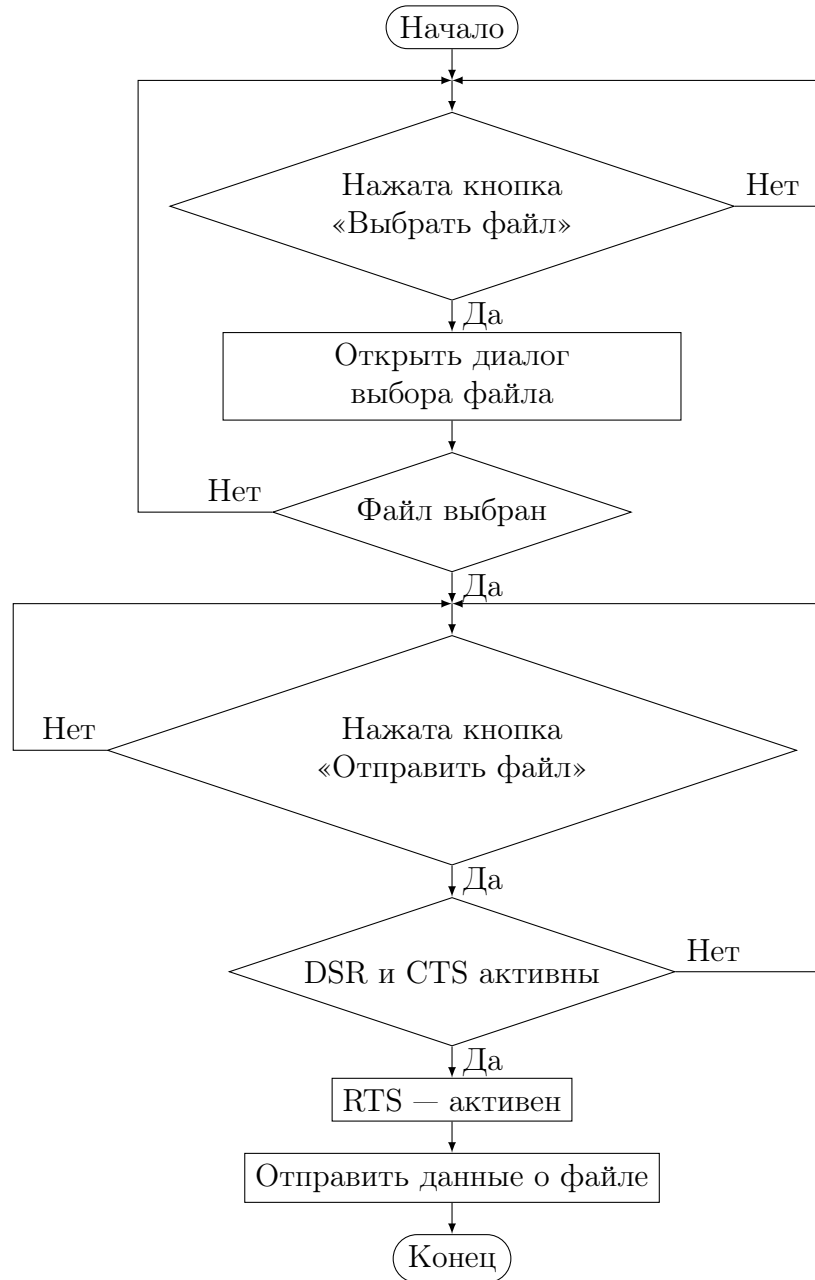
- пользовательский уровень, реализующий логику передачи файлов;
- канальный уровень, составляющий кадр для передачи по локальной сети;
- физический уровень, передающий байты через интерфейс RS-232C.

## 3.1. Алгоритмы передачи файла

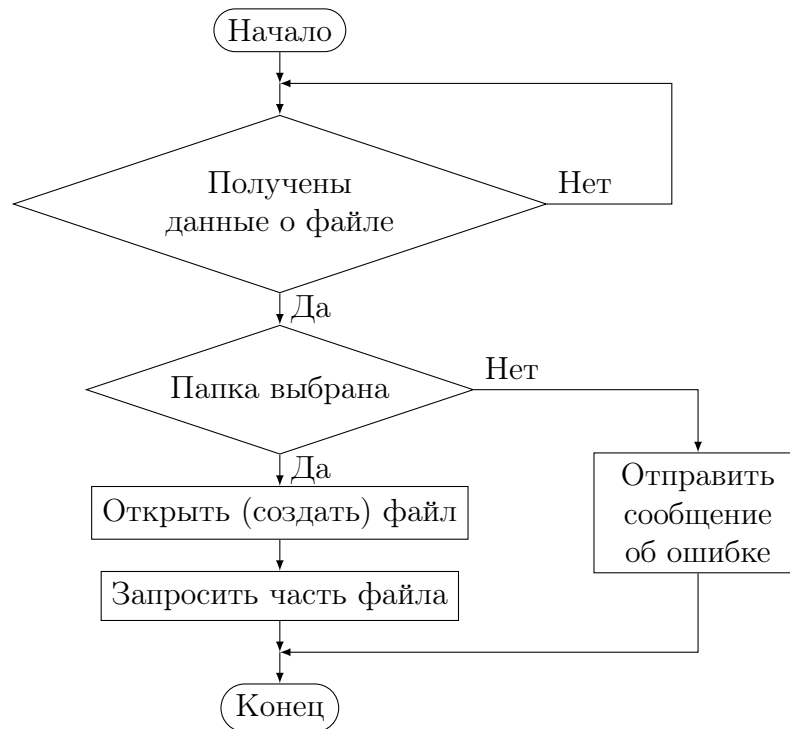
### 3.1.1. Алгоритм выбора папки



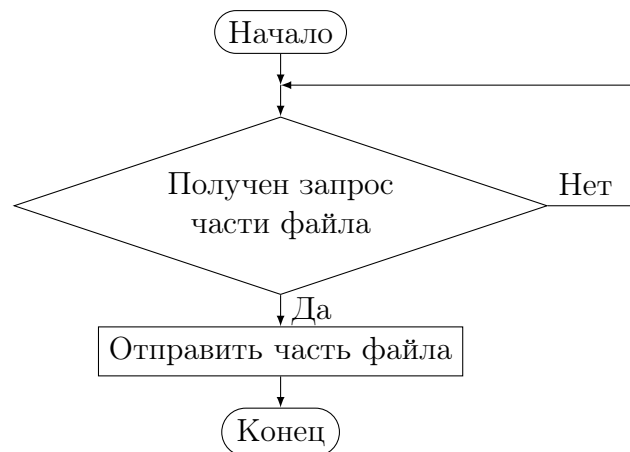
### 3.1.2. Алгоритм начала отправки файла



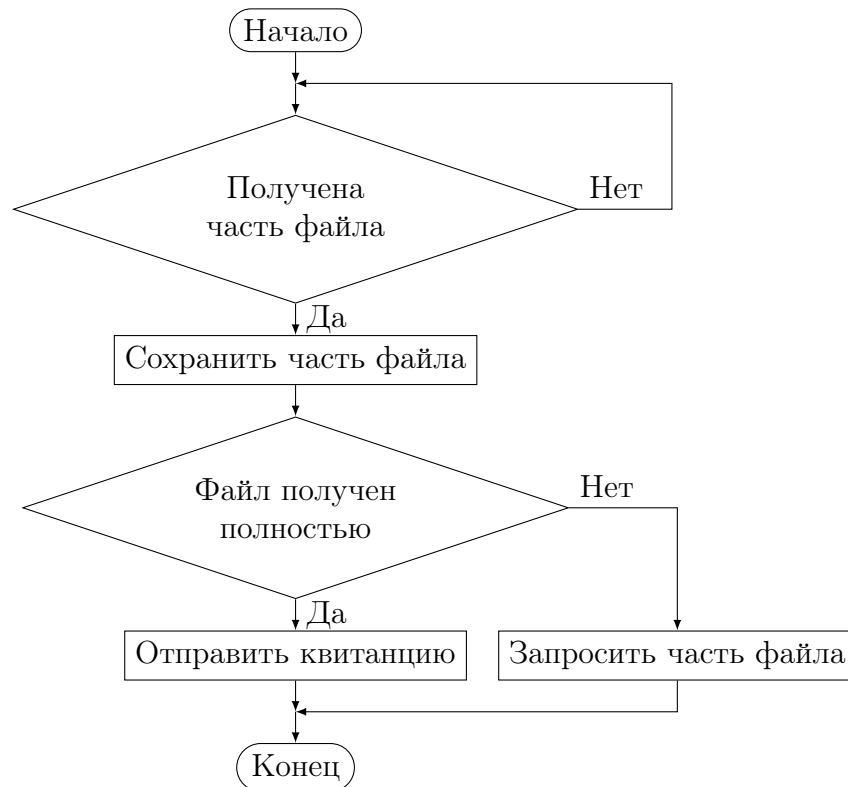
### 3.1.3. Алгоритм начала получения файла



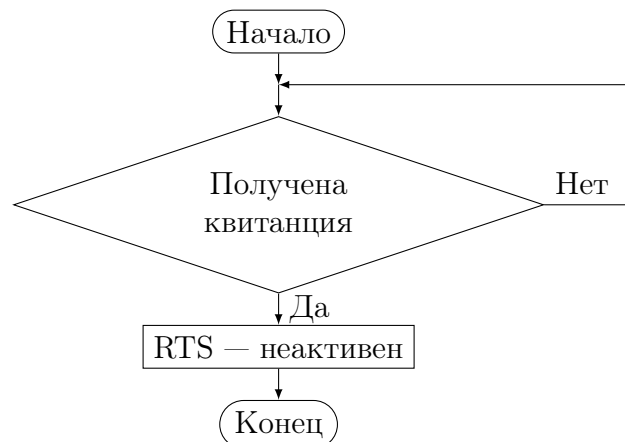
### 3.1.4. Алгоритм отправки части файла



### 3.1.5. Алгоритм получения части файла



### 3.1.6. Алгоритм завершения передачи файла



## 4. Используемые технические средства

Программа должна работать на IBM-совместимом персональном компьютере следующей конфигурации:

- Центральный процессор с частотой 1 ГГц или быстрее;
- Объем оперативной памяти от 4 ГБ;
- Жёсткий диск объемом от 20 ГБ и свободным пространством минимум 100 МБ;
- Графическая карта, поддерживающая DirectX 9 или новее с драйвером WDDM 1.0;
- Монитор разрешением от 800 × 600 пикселей;

- Операционная система Microsoft Windows версии 10 или новее.

Для работы программы требуются два персональных компьютера данной конфигурации, соединенных нуль-модемным кабелем через интерфейс RS-232C, либо (для тестирования) один персональный компьютер данной конфигурации с настроенным виртуальным нуль-модемным кабелем.

## 5. Входные и выходные данные

### 5.1. Входные данные

Входными данными является двоичный файл на передающем персональном компьютере.

### 5.2. Выходные данные

Выходными данными является двоичный файл в заданной папке на принимающем персональном компьютере.

## 6. Спецификация данных

### 6.1. Формат кадра

Наименование поля	Тип поля	Размер поля (в байтах)	Значение
Стартовый байт	byte	1	0xFF
Контрольная сумма	uint	4	$\sum_{i=1}^m \text{newPacket}[i]$
Пакет	byte[]	$m$	newPacket
Стоповый байт	byte	1	0xFF

### 6.2. Форматы пакетов

#### 6.2.1. Пакет данных о файле

Наименование поля	Тип поля	Размер поля (в байтах)	Значение
Тип пакета	byte	1	0x00
Имя файла	string	$k + \lfloor \log_{128} k \rfloor + 1$	fileDialog.SafeFileName
Длина файла	long	8	fileStream.Length
Контрольная сумма	byte[]	64	SHA512(fileStream)

#### 6.2.2. Пакет неготовности к получению файла

Наименование поля	Тип поля	Размер поля (в байтах)	Значение
Тип пакета	byte	1	0x01

#### 6.2.3. Пакет запроса части файла

Наименование поля	Тип поля	Размер поля (в байтах)	Значение
Тип пакета	byte	1	0x02
Позиция	long	8	fileStream.Length

#### 6.2.4. Пакет части файла

Наименование поля	Тип поля	Размер поля (в байтах)	Значение
Тип пакета	byte	1	0x03
Позиция	long	8	Позиция в файле
Размер части	short	2	$n$
Часть файла	byte[]	$n$	Данные

#### 6.2.5. Пакет квитанции о получении файла

Наименование поля	Тип поля	Размер поля (в байтах)	Значение
Тип пакета	byte	1	0x04

#### 6.2.6. Пакет квитанции о завершении передачи

Наименование поля	Тип поля	Размер поля (в байтах)	Значение
Тип пакета	byte	1	0x05

#### 6.2.7. Пакет разрыва соединения

Наименование поля	Тип поля	Размер поля (в байтах)	Значение
Тип пакета	byte	1	0x06

#### 6.2.8. Пакет квитанции о разрыве соединения

Наименование поля	Тип поля	Размер поля (в байтах)	Значение
Тип пакета	byte	1	0x07

### 7. Спецификация функций и классов

1. `T:iu5nt.App` — основной класс приложения, запускает окно программы.
2. `M:iu5nt.App.InitializeComponent` — стандартный метод инициализации графического интерфейса.
3. `M:iu5nt.App.Main` — стандартный метод входа в программу.
4. `T:iu5nt.DataLink` — класс, реализующий канальный уровень.
5. `F:iu5nt.DataLink.currentPacket` — текущий передаваемый пакет.
6. `F:iu5nt.DataLink.checkSumm` — контрольная сумма передаваемого пакета.
7. `F:iu5nt.DataLink.length` — длина передаваемого пакета.
8. `F:iu5nt.DataLink.recievedPacket` — получаемый пакет.
9. `F:iu5nt.DataLink.recievedPacketBuffer` — биты получаемого пакета.
10. `F:iu5nt.DataLink.debugBuffer` — отладочный буфер получаемых битов.
11. `F:iu5nt.DataLink.firstTrigger` — найден ли стартовый байт.
12. `F:iu5nt.DataLink.secondTrigger` — найден ли стоповый байт.
13. `F:iu5nt.DataLink.screenTrigger` — был ли удалён экранирующий байт.
14. `F:iu5nt.DataLink.firstTPosition` — длина полученного пакета без стартового и стопового байтов.
15. `F:iu5nt.DataLink.cleanerTimer` — таймер очистки буфера получаемого пакета.
16. `T:iu5nt.DataLink.RecieveMETHod` — тип обработчика приёма пакета.
17. `E:iu5nt.DataLink.OnRecieve` — событие приёма пакета.
18. `M:iu5nt.DataLink.#cctor` — статический конструктор класса, реализующего канальный уровень.



19. `M:iu5nt.DataLink.TimerListener(System.Object, System.Timers.ElapsedEventArgs)` — метод очистки буфера получаемого пакета.
20. `M:iu5nt.DataLink.RecievePacket(System.Collections.BitArray)` — обработчик принятых из порта битов.
21. `M:iu5nt.DataLink.SendPacket(System.Byte[])` — метод отправки пакета.
22. `T:iu5nt.Physical` — класс, реализующий физический уровень.
23. `F:iu5nt.Physical._serialPort` — объект COM-порта.
24. `F:iu5nt.Physical.connected` — открыт ли COM-порт.
25. `F:iu5nt.Physical.failList` — список синдромов ошибки циклического кода.
26. `T:iu5nt.Physical.PortListener` — тип обработчика изменения состояния контактов COM-порта.
27. `E:iu5nt.Physical.OnCheck` — событие изменения состояния контактов COM-порта.
28. `E:iu5nt.Physical.UICheck` — событие внешнего изменения состояния контактов COM-порта.
29. `M:iu5nt.Physical.SetRts(System.Boolean)` — метод изменения состояния RTS.
30. `M:iu5nt.Physical.StatusCheck(System.Object, System.IO.Ports.SerialPinChangedEventArgs)` — обработчик внешнего изменения состояния контактов COM-порта.
31. `M:iu5nt.Physical.Connect(System.String)` — метод открытия COM-порта.
32. `M:iu5nt.Physical.Disconnect` — метод закрытия COM-порта.
33. `M:iu5nt.Physical.DataReceivedHandler(System.Object, System.IO.Ports.SerialDataReceivedEventArgs)` — обработчик получения данных из COM-порта.
34. `M:iu5nt.Physical.Learning` — метод создания списка синдромов ошибки циклического кода.
35. `M:iu5nt.Physical.DeCycle(System.Byte[])` — метод декодирования циклического кода.
36. `M:iu5nt.Physical.Send(System.Collections.BitArray)` — метод отправки битов в порт.
37. `M:iu5nt.Physical.BitArrayToByteArray(System.Collections.BitArray)` — метод преобразования массива битов в массив байтов.
38. `M:iu5nt.Physical.GetCycled(System.Collections.BitArray)` — метод кодирования циклическим кодом.
39. `T:iu5nt.MainWindow` — класс окна программы.
40. `F:iu5nt.MainWindow.ConnectionBox` — группа компонентов управления соединением.
41. `F:iu5nt.MainWindow.PortsList` — список портов.
42. `F:iu5nt.MainWindow.OpenButton` — кнопка открытия порта.
43. `F:iu5nt.MainWindow.CloseButton` — кнопка закрытия порта.
44. `F:iu5nt.MainWindow.DtrIndicator` — индикатор состояния DTR.
45. `F:iu5nt.MainWindow.DsrIndicator` — индикатор состояния DSR.
46. `F:iu5nt.MainWindow.RtsIndicator` — индикатор состояния RTS.
47. `F:iu5nt.MainWindow.CtsIndicator` — индикатор состояния CTS.
48. `F:iu5nt.MainWindow.FileBox` — группа компонентов отправки файла.
49. `F:iu5nt.MainWindow.FileName` — текстовое поле для отображения названия выбранного файла.
50. `F:iu5nt.MainWindow.SelectFile` — кнопка выбора файла.
51. `F:iu5nt.MainWindow.SendFile` — кнопка отправки файла.

- 52. `F:iu5nt.MainWindow.DirectoryBox` — группа компонентов выбора папки.
- 53. `F:iu5nt.MainWindow.DirectoryName` — текстовое поле для отображения названия выбранной папки.
- 54. `F:iu5nt.MainWindow.SelectDirectory` — кнопка выбора папки.
- 55. `F:iu5nt.MainWindow.StatusText` — текстовое поле для отображения текущего состояния программы.
- 56. `F:iu5nt.MainWindow.ProgressBar` — индикатор передачи файла.
- 57. `F:iu5nt.MainWindow.DisconnectButton` — кнопка разрыва логического соединения.
- 58. `F:iu5nt.MainWindow.fileDialog` — диалоговое окно выбора файла.
- 59. `F:iu5nt.MainWindow.folderDialog` — диалоговое окно выбора папки.
- 60. `F:iu5nt.MainWindow.folderReady` — выбрана ли папка.
- 61. `F:iu5nt.MainWindow.sending` — режим передачи.
- 62. `F:iu5nt.MainWindow.fileStream` — открытый файл.
- 63. `F:iu5nt.MainWindow.fileName` — имя файла.
- 64. `F:iu5nt.MainWindow.hashName` — контрольная сумма файла.
- 65. `F:iu5nt.MainWindow.filePath` — путь к файлу.
- 66. `F:iu5nt.MainWindow.tempPath` — путь к временному файлу.
- 67. `F:iu5nt.MainWindow.length` — длина файла.
- 68. `F:iu5nt.MainWindow.chunkSize` — максимальный размер части файла.
- 69. `F:iu5nt.MainWindow.timer` — таймер ожидания ответа.
- 70. `F:iu5nt.MainWindow.retries` — количество попыток отправки пакета.
- 71. `F:iu5nt.MainWindow.maxRetries` — максимальное количество попыток отправки пакета.
- 72. `F:iu5nt.MainWindow.lastPacket` — последний отправленный пакет.
- 73. `M:iu5nt.MainWindow.#ctor` — конструктор окна программы.
- 74. `M:iu5nt.MainWindow.OpenButton_Click(System.Object, System.Windows.RoutedEventArgs)` — обработчик нажатия на кнопку открытия порта.
- 75. `M:iu5nt.MainWindow.CloseButton_Click(System.Object, System.Windows.RoutedEventArgs)` — обработчик нажатия на кнопку закрытия порта.
- 76. `M:iu5nt.MainWindow.SelectFile_Click(System.Object, System.Windows.RoutedEventArgs)` — обработчик нажатия на кнопку выбора файла.
- 77. `M:iu5nt.MainWindow.SelectDirectory_Click(System.Object, System.Windows.RoutedEventArgs)` — обработчик нажатия на кнопку выбора папки.
- 78. `M:iu5nt.MainWindow.SendFile_Click(System.Object, System.Windows.RoutedEventArgs)` — обработчик нажатия на кнопку отправки файла.
- 79. `M:iu5nt.MainWindow.DisconnectButton_Click(System.Object, System.Windows.RoutedEventArgs)` — обработчик нажатия на кнопку разрыва логического соединения.
- 80. `M:iu5nt.MainWindow.InvokeHandler(System.Byte[], System.Boolean)` — промежуточный обработчик приёма пакета.
- 81. `M:iu5nt.MainWindow.ReceiveMessage(System.Byte[], System.Boolean)` — обработчик приёма пакета.
- 82. `M:iu5nt.MainWindow.ParseFileName(System.IO.BinaryReader)` — обработчик пакета данных о файле.

- 83. `M:iu5nt.MainWindow.RequestFileChunk` — метод для запроса следующей части файла.
- 84. `M:iu5nt.MainWindow.SendFileChunk(System.IO.BinaryReader)` — обработчик пакета запроса части файла.
- 85. `M:iu5nt.MainWindow.SaveFileChunk(System.IO.BinaryReader)` — обработчик пакета части файла.
- 86. `M:iu5nt.MainWindow.SendPacket(System.Byte[])` — метод отправки пакета.
- 87. `M:iu5nt.MainWindow.ResendPacket(System.Object, System.EventArgs)` — обработчик таймера повторной отправки пакета.
- 88. `M:iu5nt.MainWindow.PortCheck(System.Boolean, System.Boolean, System.Boolean, System.Boolean)` — прокси-обработчик изменения состояния контактов COM-порта.
- 89. `M:iu5nt.MainWindow.RealPortCheck(System.Boolean, System.Boolean, System.Boolean, System.Boolean)` — обработчик изменения состояния контактов COM-порта.
- 90. `M:iu5nt.MainWindow.ExceptionHandler(System.Object, System.Windows.Threading.DispatcherUnhandledExceptionEventArgs)` — обработчик неперехваченных исключений.
- 91. `M:iu5nt.MainWindow.Dispose` — обработчик закрытия окна для очистки ресурсов.
- 92. `T:iu5nt.MainWindow.MessageType` — перечисление возможных типов пакетов.
- 93. `M:iu5nt.MainWindow.InitializeComponent` — стандартный метод инициализации компонентов.