# Base

1. PathNode: class, Control
   * Description: virtual string {get; set;}
   * Source: virtual string {get; set;}
   * Destination: virtual string {get; set;}
   * Equals (Object): override bool
   * GetHashCode (): override int
2. DirectoryExtension: static class
   * CopyTo (this DirectoryInfo source, string destinationPath): void public static

# Configurator: Control

1. PathBox: Control
   * PathField: TextBox private
   * ChangePath: Button private
   * PathLabel: Label private
   * PathDialog: FolderBrowserDialog private
   * Label: string {public get; public set;}
   * Path: string {public get; public set;}
2. NodePreviewBox: PathNode
   * DescriptionLabel: Label private
   * SourcePathLabel: Label private
   * DescriptionPathLabel: Label private
   * DescriptionField: TextBox readonly private
   * SourcePathField: TextBox readonly private
   * DestinationPathField: TextBox readonly private
   * DownArrow: Button private
   * Description: string {public get; public set;}
   * Source: string {public get; public set;}
   * Destination: string {public get; public set;}
   * DescriptionLabelText: string {public get; public set;}
   * PathLabelText: string {public get; public set;}
   * DownArrowClick: event public
3. NodeBox: PathNode
   * SourceBox: PathBox private
   * DestinationBox: PathBox private
   * UpArrow: Button private
   * Source: string {public get; public set;}
   * Destination: string {public get; public set;}
   * UpArrowClick: event public
4. NodeCollectionBox: Control, Ienumerable (реализовал просто ужасно)
   * AddNode: Button private
   * DeleteNode: Button private
   * Nodes: List < PathNode > private // Содержит Контролы NodePreviewBox и NodeBox
   * OnMouseEnter (object sender, EventArgs e): void private
   * OnMouseLeave (object sender, EventArgs e): void private
   * OnDownArrowClick (object sender, EventArgs e): void private
   * OnUpArrowClick (object sender, EventArgs e): void private
   * AddPathNode (string sourcePath, string DestinationPath, string Description): Bool public
   * AddPathNode (PathNode pathNode): Bool public
   * AddPathNodeList (List< PathNode > nodes): Bool public
   * Clear (): void public // Удаляет все PathNode
   * GetPathNodeList: List< PathNode > public
5. Configurator: Control
   * NodeCollection: NodeCollectionBox private
   * Save: Button private
   * Back: Button private
   * SaveConfig (string path): void private
   * LoadConfig (string path): List< PathNode > public

# SelectionMenu: Control

1. NodeCheckerBox: PathNode
   * Checker: CheckBox private
   * StatusLabel: Label private
   * Check (): void private
   * ConnectionStatus: enum public // Established, Failed, FatalError
   * Status: ConnectionStatus public // Зависит от существования Source и Destination
   * ChangeStatus (ConnectionStatus status): void public
   * Checked: bool public
   * Description: string {public get; public set;}
   * Source: string {public get; public set;}
   * Destination: string {public get; public set;}
2. NodeCheckerCollectionBox: Control, IEnumerable
   * Nodes: List< PathNode > private
   * AddPathNode (string sourcePath, string DestinationPath, string Description): Bool public
   * AddPathNode (PathNode pathNode): Bool public
   * AddPathNodeList (List< PathNode > nodes): Bool public
   * Clear (): void public // Удаляет все PathNode
   * GetPathNodeList: List< PathNode > public // Возвращает NodeCheckerBox’ы, у которых статус не FatalError и Checked == true

# UpdateMenu: Control

* + Nodes: List< PathNode > private
  + FilesCount: int ptivate
  + Counter: int private
  + Updater: Timer private
  + OnTick (object sender, EventArgs e): void private
  + ProgressBox: ProgressBar private
  + GetFilesCountByDirectoryPath (string Path): int private
  + SetPathNodeList (List< PathNode > nodeList): void public
  + AddPathNode (PathNode pathNode): void public
  + Clear (): void public

# MainForm: Form

1. States: enum private // Configureator, SelectionMenu, UpdaterMenu
2. State: States private
3. Nodes: List< PathNode > private // Для загрузки листа использовать Config.Load()
4. UpdateButton: Button private
5. ConfigButton: Button private
6. ExitButton: Button private
7. ChangeState (States state): void private