Module Class Diargam Module - name: string - type: string + Module(name: string, type: string) {virtual } + ~Module() {virtual} + GetName(): {virtual} + GetType() + inputConnections: int {virtual }+Process(input:string):string **EchoModule** ReverseModule NoopModule Delay - friend ModuleCreatorFactory - friend ModuleCreatorFactory - friend ModuleCreatorFactory - friend ModuleCr - previousInput : : + EchoModule(name: string, type: + DelayModule(n + ReverseModule(name: string, type + NoopModule(name: string, type: string) string) string) string) {virtual final} + Pr {virtual final} + Process(input:string) {virtual final} + Process(input:string) {virtual final} + Process(input:string) : string: : string: : string: : string:

Process Class <u>Diargam</u>

Process

{static} +ProcessWords(createdModules,
inputs) : vector<string>

{static} +ProcessRemainingDelayWords (createdModules, outputs) :void

Free Fuction in Process.cpp

FindOuputs(outputs, module, input): void
 FindDelyedModules(delayedModules, modules): void
 FindStartingModules(createdModules):vector<shared_ptr>

Connect Class Diargam

Connect

{static} +ConnectModules(existingModule moduleNames): void

Factory Class Diargam

ModuleCreatorFactory

{static} +CreateModule(name, type): unique_ptr<Module>

Module

eatorFactory string

ame: string, type:

ocess(input:string)