

Softwarearchitektur HAL, SWE1, WS2011/12 –

Felix Wagner (if10b048)

NS Client:

- Program: Instanz der Klasse "Client" erstellen und deren Methode "Connect" aufrufen
- Client: Verbindung zum Server herstellen, Sätze schicken und Antworten empfangen

NS Interface:

- IPlugin: "Vertrag" für Plugins festlegen
- Word: "Vertrag", wie ein Wort aussehen muss

NS Server:

- Program: Instanz der Klasse "Server" erstellen, deren Methode "ListenForClients" aufrufen und allgemeine Exceptions handeln, die weiter oben in der Architektur geworfen werden.
- Server: Plugin-Ordner erstellen, auf Client-Verbindungen warten und pro Verbindung einen neuen Thread starten, der mit der Methode ClientComm.WelcomeClient fortsetzt.
- ClientComm: Instanz des Pluginmanagers erstellen, Nachricht des Clients empfangen, Instanz des Textparsers erstellen, diesem den Clientsatz zur Verarbeitung übergeben, gesplittete Wortliste List<Word> an Pluginmanager weiterleiten, Exceptions handeln für den Fall, dass der Client unerwartet die Verbindung unterbricht. Antwort des Pluginmanagers zum Client zurückschicken. Die Navi-Liste löschen, wenn Server beendet wird (Ressourcen!)
- TextParser: Aus einem string eine Auflistung von Wörtern erstellen, Worttyp bestimmen.
- PluginManager: Plugin-dll's laden. Die aufbereitete Wortliste an Plugins schicken. Antwort des Plugins mit der höchsten Priorität zurück an ClientComm leiten.
- InvalidSentenceException: Wird an ClientComm geworfen, wenn Satz nicht mit einem Satzzeichen endet.

NS PluginMath:

- PluginMath: Verarbeitete Liste entgegennehmen, Priorität und ggf. Ergebnis einfacher mathematischer Operation berechnen.

NS PluginNav:

- PluginNavi: Verarbeitete Liste entgegennehmen, Priorität und ggf. Ort suchen, in dem eine gewünschte Straße liegt. Ergebnis zurückgeben an PluginManager.

NS PluginPsych:

- PluginPsycho: Verarbeitete Liste entgegennehmen, Priorität und ggf. passende Antwort auf den Satz suchen.

NS PluginTellDay:

- PluginTellBirthday: Verarbeitete Liste entgegennehmen, Priorität und ggf. Wochentag suchen, der einem bestimmten Datum entspricht.

NUNIT TESTS (ACHTUNG: Pfadnamen sind ggf. anzupassen, da das Programm mit relativen Pfad arbeitet, die Unit tests aber in einem anderen Ordner liegen!):

NS PluginManagerTest:

- ReadPluginsTest: Versucht Plugins aus dem Pluginfolder auszulesen.
- ReceivePluginAnswerTest: Testet, ob überhaupt Antworten zurückkommen. Testet, ob Frage an den Mathematiker korrekt beantwortet wird.

NS PluginNavTest, PluginPsychTest, PluginTellBirthdayTest (ohne GetPriorityTest):

- GetPriorityTest: Prüft die Prioritätsbedingungen des Plugins
- CalculateSentenceTest: Prüft die Korrektheit der Antwort des Plugins

NS TextParserTest:

- SplitsCorrectlyTest: Prüft, ob Sätze richtig aufgesplittet werden und richtigen Typ besitzen.

Architektur

