|  |
| --- |
| Melanie Göbel , Tobias Perny |
| JMS |
| Chatapplikation mit Java Message Service |

|  |
| --- |
|  |

Inhalt

[Aufgabenstellung 2](#_Toc403741867)

[Benotungskriterien 2](#_Toc403741868)

[Arbeitsaufteilung 3](#_Toc403741869)

[Aufwandsabschätzung 3](#_Toc403741870)

[Endzeitaufteilung 3](#_Toc403741871)

[Arbeitsdurchführung 4](#_Toc403741872)

[Testbericht 4](#_Toc403741873)

[Quellenangaben 4](#_Toc403741874)

# Aufgabenstellung

Implementieren Sie eine Chatapplikation mit Hilfe des Java Message Service. Verwenden Sie Apache ActiveMQ ([http://activemq.apache.org](http://activemq.apache.org/)) als Message Broker Ihrer Applikation. Das Programm soll folgende Funktionen beinhalten:

Benutzer meldet sich mit einem Benutzernamen und dem Namen des Chatrooms an.   
Beispiel für einen Aufruf:   
  
*vsdbchat <ip\_message\_broker> <benutzername> <chatroom>*

Der Benutzer kann in dem Chatroom (JMS Topic) Nachrichten an alle Teilnehmer eine Nachricht senden und empfangen.   
Die Nachricht erscheint in folgendem Format:  
  
*<benutzername> [<ip\_des\_benutzers>]: <Nachricht>*

Zusätzlich zu dem Chatroom kann jedem Benutzer eine Nachricht in einem persönlichen Postfach (JMS Queue) hinterlassen werden. Der Name des Postfachs ist die IP Adresse des Benutzers (Eindeutigkeit).  
  
Nachricht an das Postfach senden:   
*MAIL <ip\_des\_benutzers> <nachricht>*  
Eignes Postfach abfragen:   
*MAILBOX*

Der Chatraum wird mit den Schlüsselwort EXIT verlassen. Der Benutzer verlaesst den Chatraum, die anderen Teilnehmer sind davon nicht betroffen.

# Benotungskriterien

o 2 Punkte: Installation Message Broker Apache ActiveMQ  
o 8 Punkte: Implementierung des Chatraums (JMS Topic)  
o 6 Punkte: Implementierung der Postfach-Funktionalität (JMS Queue)

Abnahmen, die nur auf localhost basieren sind unzulässig und werden mit 6 Minuspunkten benotet!

# Arbeitsaufteilung

## Aufwandsabschätzung

|  |  |  |
| --- | --- | --- |
| Aufgabe | Person(en) | Zeitschätzung in Minuten |
| Installation von Apache ActiveMQ | Göbel, Perny | 5 |
| Implementierung Chatroom | Göbel | 120 |
| Implementierung Postfach | Perny | 120 |
| Testen | Göbel,Perny | 30 |
| Protokoll | Göbel | 40 |
| Bugfixing | Perny | 60 |
| Gesamt | Göbel | 195 = 3 Stunden 15 Min |
| Gesamt | Perny | 215 = 3 Stunden 35 Min |
| Gesamt | Gesamt | 410 |

## Endzeitaufteilung

|  |  |  |
| --- | --- | --- |
| Aufgabe | Person | Zeit |
| Installation von Apache ActiveMQ |  | 5 |
| Implementierung Chatroom |  |  |
| Implementierung Postfach |  |  |
| Testen |  |
| Protokoll |  |  |
| Bugfixing |  |  |
| Gesamt | Göbel |  |
| Gesamt | Perny |  |
| Gesamt | Gesamt |  |

# Theorie

## Java Message Service

Ist eine Java-API die Nachrichten via eine Message Oriented Middleware (MOM) Nachrichten zwischen Sender und Receiver vermittelt. Die Message Oriented Middleware verwaltet die Nachrichten mit 2 verschiedenen Konzepten

Queue (Point-to-Point)

Jede Nachricht bekommt einen bestimmten Nachrichtkanal und wird zur Abholung bereitgestellt. Nachrichten können nur einmal abgeholt werden, das System folgt nach den *First Come First Serve* Prinzip.

Topic (Publisher/Subscriber)

Ein Absender schickt eine Nachricht an mehrere Empfänger (Subscriber), die Nachricht wird automatisch ausgeteilt. Wenn sich ein Empfänger abmeldet, hat es keine Auswirkung auf andere Empfänger.

# Designüberlegung

# Arbeitsdurchführung

# Testbericht

# Quellenangaben