

Hinweis:

Sie haben die beiden Klassen `Socket` und `ServerSocket` erhalten. Dies sind nicht die gleichnamigen Java-JDK-Standardklassen. Sie wurden speziell für die Abiturprüfung in Hessen geschrieben und stehen dort als Klassendiagramm mit Dokumentation zur Verfügung. In Klausuren/Tests werden diese Klassen ebenfalls zur Verfügung gestellt. Sowohl in der Abiturprüfung als auch in Klausuren/Tests/Abgaben/Hausaufgaben ist die Verwendung dieser Klassen verbindlich. Dies gilt ab sofort. Damit der Compiler diese Klassen und nicht die Standardklassen verwendet, müssen sie im selben Verzeichnis wie das zu erstellende Programm liegen.

1. Aufgabe

Erstellen Sie einen Server, welcher die nachfolgend beschriebenen Anforderungen erfüllt. Zum Testen des Servers verwenden Sie den bereits bekannten Telnet-Client.

Beschreibung der Aufgaben Servers:

Verbindet sich der Client mit dem Server, so schickt der Server folgende Aufforderung an den Client: "Bitte Passwort eingeben." Der Client sendet anschließend das Passwort (String, ohne Verschlüsselung), welches vorher mittels Tastatur einzugeben ist. Ist das Passwort korrekt, so sendet der Server eine kurze Nachricht (z. B. "Treff morgen 12:00 bei Marie Niert."), welche der Client auf der Konsole ausgibt. Danach beendet der Server die Verbindung und gibt eine entsprechende Meldung auf der Konsole aus. Anschließend wartet der Server auf die nächste Verbindungsanfrage eines Clients. Der Server soll den jeweils aktuellen Status auf der Konsole ausgeben. Die Konsolenausgaben für den oben beschriebenen Ablauf sollen etwa so aussehen:

Server:

```
C:\WINDOWS\system32\cmd.exe
*** SERVER ***
Server wartet auf Verbindungsanfrage...
Server ist mit Client verbunden...
Server hat Verbindung zum Client beendet
Server wartet auf Verbindungsanfrage...
```

Telnet-Client:

```
C:\WINDOWS\system32\cmd.exe
Passwort eingeben: horst
Nachricht: Treff morgen 12:00 bei Marie Niert
Verbindung zu Host verloren.
C:\Dokumente und Einstellungen\frank>
```

Ist das Passwort falsch, so sendet der Server eine entsprechende Meldung und beendet ebenfalls die Verbindung. Für diesen Vorgang sollen die Konsolenausgaben etwa so aussehen:

Server:

```
C:\WINDOWS\system32\cmd.exe
*** SERVER ***
Server wartet auf Verbindungsanfrage...
Server ist mit Client verbunden...
Server hat Verbindung zum Client beendet
Server wartet auf Verbindungsanfrage...
```

Telnet-Client:

```
C:\WINDOWS\system32\cmd.exe
Passwort eingeben: bielefeld
Falsches Passwort!!!
Verbindung zu Host verloren.
C:\Dokumente und Einstellungen\frank>
```

Verhalten des Servers bei Fehlern:

Sollte beim Start des Servers der `ServerSocket` nicht erzeugt werden können (weil z.B. der Port bereits von einem anderen Server genutzt wird), so wird die folgende Meldung ausgegeben und das Programm wird beendet:

```
C:\WINDOWS\system32\cmd.exe
Fehler: ServerSocket kann nicht erzeugt werden
Drücken Sie eine beliebige Taste . . .
```

Sollte der Client die Verbindung abrupt trennen (wenn z.B. der Client vor Eingabe des Passwortes durch Schließung des Konsolenfensters beendet wird), so darf der Server nicht abstürzen. Er gibt eine eindeutige Fehlermeldung aus und muss weiterhin auf Verbindungsanfragen von Clients warten können.

Die Konsolenausgabe für den beschriebenen Ablauf soll etwa so aussehen:

```
C:\WINDOWS\system32\cmd.exe
*** SERVER ***
Server wartet auf Verbindungsanfrage...
Server ist mit Client verbunden...
Fehler: Verbindung zum Client wurde unterbrochen
Server wartet auf Verbindungsanfrage...
```