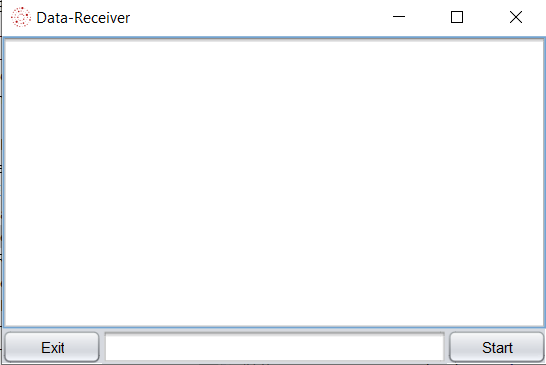
## Einleitung

Wir haben ein Fenster zur Eingabe von Fahrradinformationen für ein Verwaltungsprogramm für einen Fahrradverleih erstellt.

Für die Serialisierung über eine Internetverbindung benötigen wir auch einen Empfänger.

Dieser Empfänger kann einfach gehalten werden: Mit ‚Start‘ starten wir den Empfang, mit ‚Exit‘ beenden wir das Programm und im Hauptbereich können wir die empfangenen Daten protokollieren.

Das Programm selbst benötigt neben der main mit dem grafischen Fensterteil nur eine Methode, die den Empfang startet und Anmeldungen von einem Client entgegennimmt.

**/\* Auszug aus der GUI-Erstellung \*/**

JButton btnStart = **new** JButton("Start");

btnStart.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

**if** (running)

**return**;

startReceiver();

}

});

**/\* Empfangsmethode \*/**

**void** startReceiver() {

textField.setText("Receiver startet");

running = **true**;

java.net.ServerSocket serverSocket;

**try** {

serverSocket = **new** java.net.ServerSocket(11111);

**while** (**true**) {

// blockiert, bis sich ein Client angemeldet hat

java.net.Socket client = serverSocket.accept();

...

}

} **catch** (IOException e1) {

e1.printStackTrace();

}

}

Starten wir das Programm und starten wir den Client, können wir Daten übermitteln, wir stellen aber fest, dass das Receiver-Fenster blockiert, also auf keine Eingaben reagiert.

Zu den Hintergründen dient die Lektüre des Infoblatts (BPE12\_L4 1.2 Infoblatt Threads).