# Aufgabe 3: Objekt benutzen

## Ziel

Du lernst in deinem Code mit bereits bestehenden Objekten umzugehen. Du kennst die Begriffe *Konstruktor,* *Eigenschaften* und *Methoden.* Du weisst wie man eine kleine GUI Applikation mit Java erstellt.

## Aufgabentyp

Einzelarbeit

## Zeitaufwand

120 Minuten

## Ausgangslage

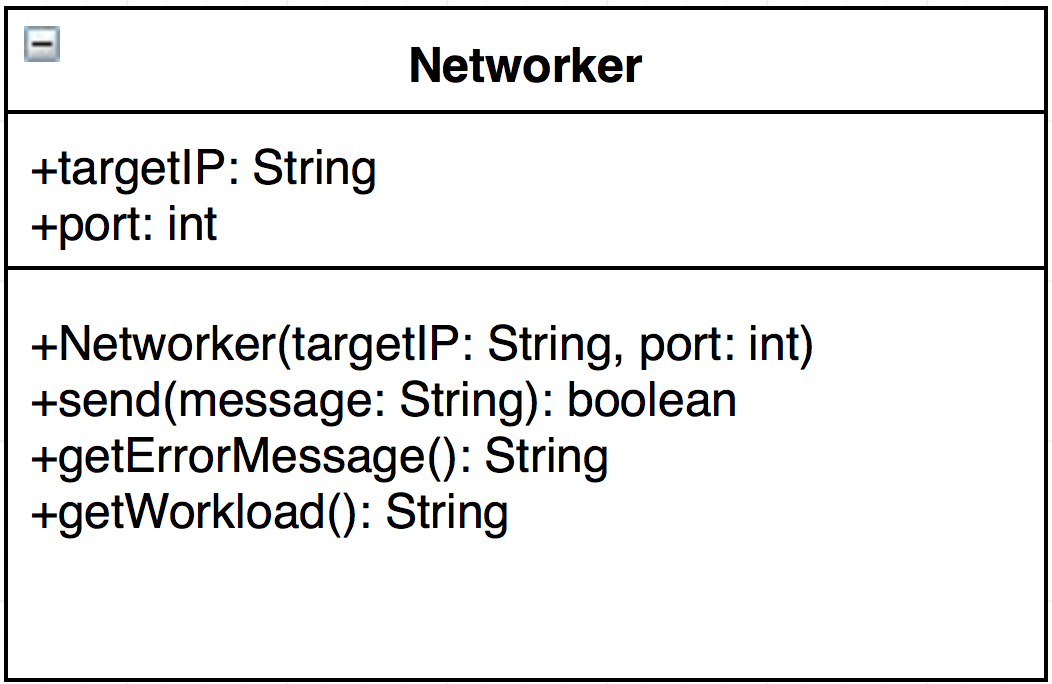
An einer grossen Messe im Hallenstadion sollen die Besucher Nachrichten von Computerstationen via WiFi verschicken können. Diese Nachrichten werden auf einem Server gesammelt und auf einer grossen Leinwand projiziert, so dass alle Besucher diese Nachrichten lesen können. Du sollst für diesen Anlass die Applikation programmieren, welche auf diesen Stationen anschliessend installiert werden soll.

## Aufgabenstellung

Erstelle eine GUI Applikation mit einem Textfeld zur Eingabe der Nachricht und einen Senden Button. Um diese Nachricht zu übertragen kannst du die bereits bestehende Klasse *Networker* benutzen.

## Vorgehensweise

1. Erstelle ein neues Java Projekt (*new->Java Project...*). Füge anschliessend ein Application Window ein (*Rechtsklick auf src Ordner->new->other...->WindowBuilder->SwingDesigner->Application Window*). Wechsle anschliessend in den Design Modus.
2. Das GUI braucht JTextField für die Eingabe der Nachricht. Benenne dieses Textfeld *txtNachricht.* Zusätzlich erstellst du einen JButton namens *btnSenden.* Mit einem Doppelklick auf den Button fügst du einen ActionListener hinzu.
3. Kopiere nun die Datei *Networker.java* in deinen *src* Ordner. Mithilfe dieser Klasse kannst du ein Objekt vom Typ *Networker* erstellen und anschliessend benutzen. Dazu brauchst du eine *Instanz* vom *Networker*. Diese erstellst du mit folgender Syntax: *Objekttyp variablenname = new ObjektTyp();* Nenne deine Instanz *net.* Folgendes Klassendiagramm soll dir einen Überblick über die Möglichkeiten des *Networker* geben (oben Eigenschaften unten Methoden):



1. Nun kannst du deinem Objekt die IP-Adresse des Servers mitteilen. Diese wird dir mitgeteilt. Es wird über den Port 49876 kommuniziert. Teile auch den Port deinem Objekt mit.
2. Versuche nun mit der send() Methode deinen Namen zu übertagen. Erscheint dein Name auf der Leinwand hast du schon ein erstes Erfolgserlebnis :-)
3. Die Methode getErrorMessage() liefert dir eine Fehlermeldung zurück, falls etwas mit der Übertragung nicht klappte. Diese kannst z.B. via *System.out.println()* ausgeben.
4. Baue den Code nun so um, dass du den Text aus dem Textfeld übernimmst. (Hinweis: *getText()* Methode vom *txtNachricht* aufrufen)
5. Zusatzaufgabe: Nach jeder Übertragung gibt der *Networker* via die Methode *getWorkload()* eine Zahl zwischen 0 und 10 zurück. Diese sagt aus, wie stark der Server im Moment belastet ist. 0-4 bedeutet: grün, kannst schicken! 5-9 bedeutet: orange, es wird knapp! 10 bedeutet: rot, bitte keine weiteren Nachrichten mehr. Du kannst die Fensterfarbe des GUIs entsprechend einfärben und eine Fehlermeldung ausgeben, falls die Workload des Servers 10 erreicht. Versuche mit Hilfe des Internets herauszufinden, wie man eine Fehlermeldung ausgibt (Stichwort: *JOptionPane, showMessage*).

## Ergebnis

Eine einfache grafische Benutzeroberfläche über welche einen Text eingegeben und an den Server übermittelt werden kann.