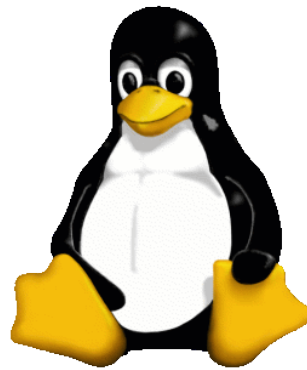


Programmierung eines NewsTickers in Java

(Applet & Applikation)



Projektarbeit Programmiertechnik:Java

NTA FH Isny
12. Info

David Mayr / Andreas Krug

Inhalt

- 1 Anforderungsprofil
- 2 Konzeption
- 3 Besonderheiten Applet/Applikation
- 4 Bedienung
- 5 Einschränkungen
- 6 Schnittstellen
- 7 Quellenangaben

1 Anforderungsprofil

Die Aufgabenstellung ist, einen NewsTicker zur Anzeige von Nachrichten-Überschriften in Form einer Laufschrift mit klickbaren Links zu den jeweiligen Internetseiten, die dann in einem Browser geöffnet werden sollen, zu realisieren.

Dabei soll der NewsTicker sowohl als Applet zum Einsatz auf einer Internet-Seite, als auch mit nur geringfügig anderer Funktionalität als eigenständige Applikation implementiert werden.

Im folgenden die aus der Aufgabenstellung ersichtlichen Unterschiede in einer Gegenüberstellung:

	Applet	Applikation
Einlesen von Nachrichten-Überschriften und dazugehöriger Internet-Links ...	aus den Parametern der HTML-Datei	aus den Daten des Projekts "Applikation zum Erfassen von IT-News"
Übergabe von Konfigurations-Parameter (Laufrichtung, Geschwindigkeit)	aus den Parametern der HTML-Datei	"geeignete Methode" (verwendet: Kommandozeilen-Argumente)
Anpassen der Schrift-Attribute (Schrift, Farbe, Grösse, Stil)	durch die Parameter der HTML-Datei	"geeignete Methode" (verwendet: Kommandozeilen-Argumente)

2 Konzeption

Da der NewsTicker sowohl als Applet als auch als Applikation implementiert werden sollte, wurde je eine Klasse mit den jeweils Applet- bzw. Applikations-spezifischen Funktionen erstellt.

Im Falle des **Applets** müssen die Daten¹ aus den HTML-Parametern des Applets ausgelesen werden. Dies wurde in einer eigenen Klasse mit dem Namen **HTMLParser** realisiert, aus dessen Instanz dann mit entsprechenden get...()-Methoden die Daten im Applet-Objekt zur Verfügung gestellt werden können.

Im Falle der **Applikation** müssen die Daten aus zwei Quellen ermittelt werden: den Kommandozeilen-Argumenten² und aus einer XML-Datei (Nachrichten mit Links). Dafür wurden die beiden Klassen **ArgsParser** und **XMLParser** erstellt, die wie der HTMLParser auch die Daten über entsprechende get...()-Methoden zurückgeben können. Ausserdem muss bei der Applikation erst noch ein Frame-Objekt – das Programmfenster – erstellt und konfiguriert werden.

In allen Fällen, in denen Werte eingelesen werden die optional sind³, wurden in der Klasse **Standard** Vorgabewerte eingeführt, die dann eingesetzt werden.

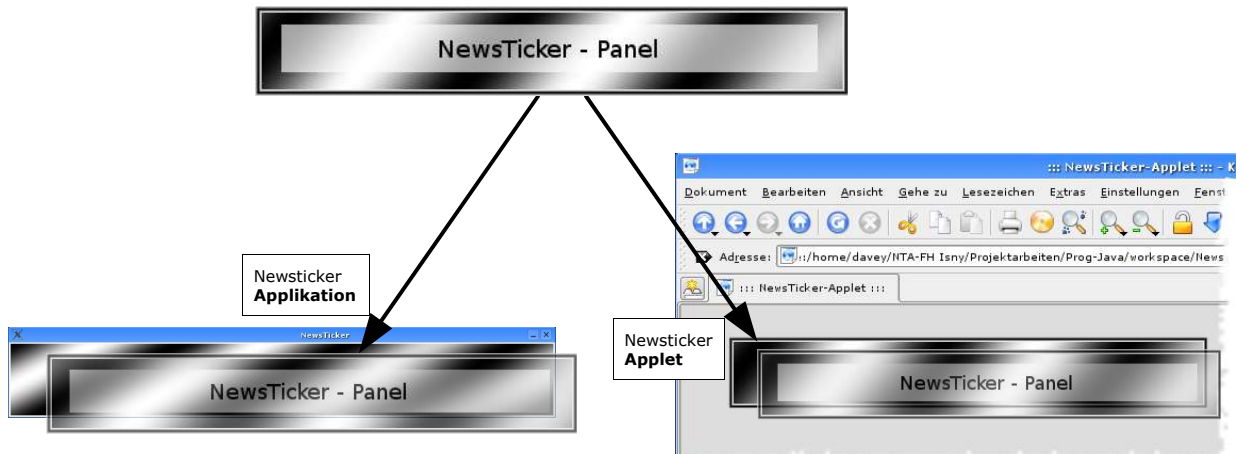
Die eigentliche Newsticker-Funktionalität ist in der Klasse **NewsTicker** enthalten, die von der AWT-Klasse **Panel** abgeleitet ist.

In beiden Fällen - sowohl beim Applet als auch in Applikation - wird eine Instanz dieses allgemeinen **NewsTicker**-Panels gebildet und die eingelesenen Daten an dessen Konstruktor übergeben. Das NewsTicker-Panel wird dann dem Applet bzw. dem Frame einfach hinzugefügt und erledigt den Rest.

1 Daten bedeutet hier: alle Nachrichten mit den dazugehörigen Links und evtl. pro-Nachricht Schrift-Einstellungen. Ausserdem die allgemeine Konfiguration des Newstickers wie Laufrichtung und Geschwindigkeit.

2 darunter auch der Dateiname einer XML-Datei mit den Nachrichten im RDF- / RSS-Format

3 Im Prinzip sind beim Applet alle Parameter optional – bei der Applikation muss nur die XML-Datei mit dem richtigen Namen vorhanden sein, wenn kein Dateiname als Argument übergeben wurde.



Alle Nachrichten sind Instanzen der Klasse **Nachricht**, die für jedes Nachrichtenobjekt Attribute wie u.a. den Titel, den Link, die Position und Ausmaße speichert und über entsprechende Methoden kapselt – so 'weiß' jedes Nachrichtenobjekt selbst, wo und wie es angezeigt werden soll.

Nach der Anfangs-Initialisierung wird jedem Nachrichtenobjekt mit der Methode **resetPositionen()** eine Ausgangsposition in Abhängigkeit der Laufrichtung und Position/Breite der vorherigen Nachricht zugewiesen – ausserhalb des sichtbaren Bereichs. Anhand folgender Graphik sollte das leichter verständlich werden:

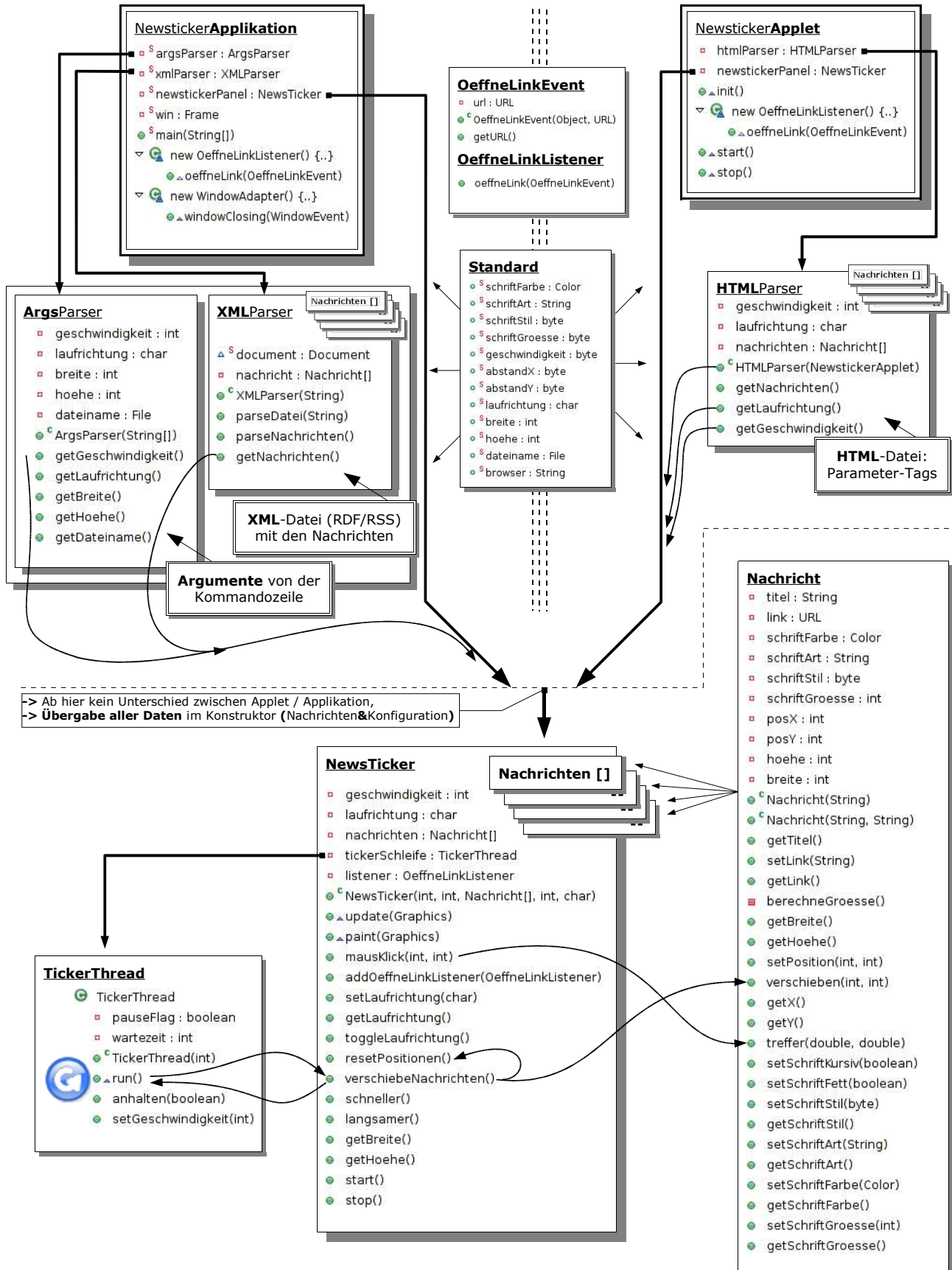


Jedes Nachrichten-Objekt kann über die Methode **verschieben(int,int)** im Prinzip in alle Richtungen verschoben werden, indem einfach die zu verschiebende Pixel-Anzahl in x- und y-Richtung übergeben werden.

Ausserdem kann mit der Methode **treffer(int, int)** jedes Nachrichten-Objekt 'gefragt' werden, ob es von der Maus an den übergebenen Koordianten "getroffen" wurde um dann evtl. ein OeffneLinkEvent zu erzeugen, der den Link der Nachricht in einem Browser öffnet.

Über eine Instanz der Unterklasse **TickerThread** von NewsTickerView, welche von **Thread** abgeleitet ist, wird in regelmässigen Abständen die NewsTickerView-Methode **verschiebeNachrichten()** aufgerufen, die je nach Laufrichtung alle Nachrichten verschiebt bzw. deren Positions-Attribute anpasst. Nach jedem Verschieben aller Nachrichten muss das NewsTickerView-Panel neu gezeichnet werden. Dann wird geprüft, ob die letzte Nachricht bereits links aus dem Panel verschwunden ist – ist das der Fall, werden wieder alle Positionen zurückgesetzt.

Um das Zusammenspiel der einzelnen Klassen besser verstehen zu können, sollte folgende Graphik hilfreich sein:



3 Besonderheiten

Browser starten: Applet

Aus einem Applet heraus, das bereits in einem Browser läuft, lässt sich ein weiterer Browser mit folgendem Aufruf starten:

```
getAppletContext().showDocument( "http://www.blah.com" , "_blank" );
```

Dieser Aufruf wurde in die Anonyme **OeffneLinkListener**-Klasse in die überschriebene Methode **OeffneLink** innerhalb der Applet-**init()** Methode platziert, um immer dann ausgeführt zu werden, wenn der OeffneLinkEvent erzeugt wird.

Browser starten: Applikation

Aus einer Applikation heraus lautet die Anweisung ein externes Programm zu starten wie folgt:

```
Runtime.getRuntime().exec( new String[] { "programmname" , "argumente" } );
```

Dieser Aufruf wurde in die Anonyme **OeffneLinkListener**-Klasse in die überschriebene Methode **OeffneLink** innerhalb der **main()**-Methode der Applikation platziert, um immer dann ausgeführt zu werden, wenn der OeffneLinkEvent erzeugt wird.

4 Bedienung

Wird eine Nachrichten-Überschrift angeklickt und getroffen, wird der damit verbundene Link in einem Internet-Browser geöffnet. Bei einem Mausklick am linken Rand wird die Laufgeschwindigkeit verringert, am rechten Rand erhöht und im übrigen Bereich die Laufrichtung gewechselt.

5 Einschränkungen

Läuft der Newsticker als Applikation, versucht diese beim klicken einer Nachrichten-Überschrift das Kommando "firefox"⁴ mit der aufzurufenden URL als Argument zu starten. Dieses Kommando muss sich im Pfad der ausführbaren Programme befinden⁵ - ist das nicht der Fall, kann der Browser nicht gestartet werden. Beim Applet gilt diese Einschränkung nicht - diesem ist es innerhalb eines Browsers leicht möglich eine zweite Instanz des Browsers mit dem Link zu öffnen.

4 Auch das Browser-Kommando kann in der Klasse "Standard" geändert werden.

5 Unter Linux ist das bei installiertem Firefox-Browser in der Regel gegeben.

6 Schnittstellen

Die erforderliche Schnittstelle zum Projekt "Java-Applikation zum Erfassen von verschiedenen IT-Nachrichten" wurden einfach dadurch realisiert, dass bei unserer Newsticker-Applikation der Dateiname der Nachrichten-Datei⁶ sowie die Breite, Höhe, Geschwindigkeit und Laufrichtung des Newstickers als Kommandozeilen-Argumente übergeben werden können.

Die dafür zuständige Klasse "ArgsParser" sorgt ausserdem dafür, dass bei Aufruf der Applikation mit dem Argument "--help" oder "-h" folgenden Hilfe-Text zur Verwendung der möglichen Argumente ausgibt:

```
NewsTicker. Moegliche Argumente:  
--geschwindigkeit [1-9]  
--laufrichtung    [h/v]  
--breite         [breite in pixel]  
--hoehe         [hoehe in pixel]  
--datei         [xml-dateiname(rss/rdf)]  
--schriftgroesse [in punkten]  
--help         diese Hilfe.
```

Damit ist es der Projektgruppe, die diesen Newsticker in ihr Projekt einbinden soll, einfach möglich ihn als eigenes Programm mit den entsprechenden Argumenten aus ihrem Programm heraus zu starten und die erzeugten XML-Dateien anzeigen zu lassen.

7 Quellenangaben

Die Java API-Dokumentation

Diverse Internet-Seiten und Java-Tutorials

⁶ Im XML-Format (RSS/RDF), erzeugt/heruntergeladen von der Projektarbeit unserer Kommilitonen.