

# Projektdokumentation zur Anwendung der linearen Optimierung

---

## Projekt Nr. 8

### *Werbebudgetoptimierung 1.1*

#### *Datum*

26.01.2016

#### *Teilnehmerin*

Ries, Melissa

Matrikelnummer: 288 973

Altikulac, Zarife

Matrikelnummer: 287 418

## Inhaltsverzeichnis

<b>Inhaltsverzeichnis .....</b>	
<b>Abbildungsverzeichnis .....</b>	
<b>1    Einleitung.....</b>	<b>1</b>
<b>2    Werbebudgetoptimierung 1.1.....</b>	<b>2</b>
2.1    Funktionen der Methode.....	2
2.2    Ausgangssituation der Methode .....	2
2.3    Anforderungen vor dem Projektstart .....	2
2.4    Situationslage nach der Realisierung.....	3

## Abbildungsverzeichnis

Abbildung 1: Korrektur des Rechtschreibfehlers .....	3
Abbildung 2: Änderung in der Klasse "Indexfeld" .....	3
Abbildung 2: Ausschnitt der UniCode-Verbesserung.....	4
Abbildung 3: Implementierte Hilfefunktion .....	5
Abbildung 5: Code zur Implementierung.....	5
Abbildung 6: Hilfe-Methode im Code Teil 1 .....	6
Abbildung 7: Hilfe-Methode im Code Teil 2 .....	6
Abbildung 8: Hilfe-Methode im Code Teil 3 .....	7
Abbildung 9: Inhalt der Hilfestellung .....	7

## 1 Einleitung

Die Veranstaltung „Anwendung der linearen Optimierung“ umfasst zwei Bereiche. Zum einen werden von einzelnen Teilnehmern drei Methoden aus OR ALPHA in Bezug auf Schwächen und Verbesserungspotenzialen analysiert und mittels Präsentationen vorgetragen. Zum anderen sind die bestehenden Fehler in Methoden bzw. Solvern, von zweier Teams in Form von Projekten zu beheben. Dabei wird der Source-Code angepasst und am Ende des Semesters, das Resultat durch Abschlusspräsentation mitgeteilt.

Vor der Zuteilung der Projekte, wird darauf achtgegeben, dass Teilnehmer bereits mit den Methoden bzw. den Solvern vertraut sind. Denn durch die Einarbeitung in die Methoden im ersten Teil, ist es angemessen auch im Projekt die entsprechenden Fehlerbehebungen vorzunehmen.

Die endgültige Abgabe erfolgt dennoch, durch die Projektdokumentation, dem Analysebericht, sowie dem neuen Source-Code.

In unserem Projekt sind die Methoden Werbebudgetoptimierung 1.1, Zuordnungsplanung 2.0 und Iterator 2.0 anzupassen. In der folgenden Dokumentation sind die verschiedenen Methoden in Abschnitten gegliedert und nach realisierenden Korrekturen erläutert. Die zugehörigen Source-Codes werden separat zum Abgabetermin abgegeben.

## 2 Werbebudgetoptimierung 1.1

Die Methode beabsichtigt eine optimale Planung von Werbemitteln in unterschiedlichen Medien wie Fernsehen, Radio, Zeitung und Sonstige Medien bspw. Werbebroschüren.

### 2.1 Funktionen der Methode

Das Kriterium kann entweder als Minimierung des Budgets oder Maximierung der potenziellen Käufer festgelegt werden. Dabei können die genannten Medien nach Kostenkategorien differenziert und zugleich die Mindest- bzw. Maximalanzahl der Werbespots bestimmt werden.

### 2.2 Ausgangssituation der Methode

- Daten können über das GUI eingegeben werden
- Der LP-Ansatz wird erzeugt (mit Hilfe des LPSolves)
- Das Ergebnis der Berechnung wird ausgegeben
- Die Funktion „Neu“ setzt alle Daten zurück
- Die Funktion „Speichern“ ermöglicht die Berechnung zu sichern
- Die Funktion „Öffnen“ ermöglicht eine hinterlegte Datei anzuzeigen

### 2.3 Anforderungen vor dem Projektstart

Die Anforderung zur Werbebudgetoptimierung war ursprünglich nur eine Hilfsfunktion auszuarbeiten und diese zu implementieren. Nach der Analyse der Methode war dennoch festzustellen, dass es mehrere Fehler gab, die es zu beheben galt.

Ausgenommen der fehlenden Hilfsfunktion war Rechtschreibfehler zu erkennen. Zugleich wurden Umlaute wie ä, ö und ü entweder ersetzt mit den Buchstaben ae, ue, oe oder diese wurden vollständig entfernt. Sowohl die Verständlichkeit bestimmter Begrifflichkeiten als auch die Ästhetik der Methode war somit nachteilig. Eine weitere Zusatzaufgabe der Werbebudgetoptimierung war die Daten in den Kostenkategorien nicht erneut eingeben zu müssen, wenn das Einsehen der Budgeteingrenzung in den einzelnen Medien erwünscht wird. Denn die Eingaben wurden teilweise automatisch gelöscht.

## 2.4 Situationslage nach der Realisierung

- Der Rechtschreibfehler in der Klasse „Indexfeld“ (Zeile 45) wurde behoben. ✓

**Zuvor:** Fernrsehanstalten

**Danach:** Fernsehanstalten

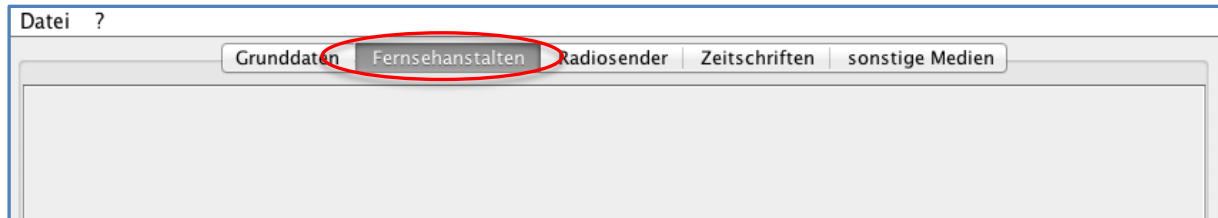


Abbildung 1: Korrektur des Rechtschreibfehlers

```

44 tp.addTab("Grunddaten", grunddaten.getGrunddaten());
45 tp.addTab("Fernsehanstalten", fernsehanstalten.getMedium());

```

Abbildung 2: Änderung in der Klasse "Indexfeld"

- Unicode vertritt die internationalen Schriftzeichen und deckt alle bekannten Textzeichen in jeweiligen Zeichensätzen ab. Zusätzlich zu den lateinischen Buchstaben werden griechisches, kyrillisches, arabisches, hebräisches, thailändisches, japanisches, chinesisches und koreanisches Alphabet ebenso als Zeichensätze bereitgestellt.<sup>1</sup>

Die Umlaute wurden in folgenden Zeilen mittels Unicode-Zeichensätzen eingefügt. ✓

### Klasse „Grunddaten“:

Zeile 92: „Maximal zur Verfgung stehendes Budget“ durch \u00fc ≙ ü

Zeile 108: „maennlich“ durch \u00e4 ≙ ä

Zeile 125: „Maximierung der potenziellen Kufer“ durch \u00e4 ≙ ä

Zeile 448: „Kein bzw. fehlerhafter Wert fr Budget eingegeben“ durch \u00fc ≙ ü

### Klasse „Medium“:

Zeile 81: „Max. zu verwendetes Budget fr dieses Medium“ durch \u00fc ≙ ü

### Klasse „SolverDaten“:

Zeile 1311: „Kann Solver bat. nicht ausfhren. Exit 1.“ durch \u00fc ≙ ü

<sup>1</sup> [http://unicode.e-workers.de/about\\_unicode.php](http://unicode.e-workers.de/about_unicode.php)

**Klasse „Tabelle“:**

Zeile 40: „mnnliche“ durch \u00e4  $\triangleq$  ä

**Klasse „WerbeBudgetApplication“:**

Zeile 234: „Loesung“ durch \u00f6  $\triangleq$  ö

Zeile 253: „Das eingegebene Problem ist unlsbar“ durch \u00f6  $\triangleq$  ö

Zeile 281: „Minimierung des Budgets ergab folgende Loesung“ durch \u00f6  $\triangleq$  ö

Zeile 287: „Maximierung der zu erreichenden Kunden ergab folgende Loesung“ durch \u00f6  $\triangleq$  ö

Zeile 295: „Maximal zur Verfuegung stehendes Budget zur Verteilung der Werbeeinheiten: “ durch \u00fc  $\triangleq$  ü

Zeile 349: „Daten fuer optimiertes Budget“ durch \u00fc  $\triangleq$  ü

Zeile 641: „Bitte einen gueltigen Dateinamen eingeben!“ durch u00fc  $\triangleq$  ü

Zeile 653: „Datei oeffnen“ durch \u00f6  $\triangleq$  ö

Zeile 675: „Bitte einen gueltigen Dateinamen eingeben!“ durch u00fc  $\triangleq$  ü

The screenshot shows a Java Swing window titled "WerbeBudgetApplication". It contains two main panels. The left panel, titled "Budget und Werbemedien", has a text field "Maximal zur Verfügung stehendes Budget:" with a red circle around "Verfügung". Below it are four text fields for "Anzahl Fernsehanstalten:", "Anzahl Radiosender:", "Anzahl Zeitschriften:", and "Sonstige Medien:", followed by a "Felder leeren" button. The right panel, titled "Optimierungskriterium", has two radio buttons: "Minimierung des Budgets" (selected) and "Maximierung der potentiellen Käufer" (with a red circle around "Käufer"). Below these is a "weiter >>" button. At the bottom left, a "Kundendaten" panel has a text field "Anzahl Kunden die erreicht werden sollen:", a "davon" label, and two radio buttons: "männlich" (selected, with a red circle) and "weiblich".

Abbildung 3: Ausschnitt der UniCode-Verbesserung

- Eine Hilfefunktion wurde ausgearbeitet und in die Menüleiste der Methode implementiert. ✓

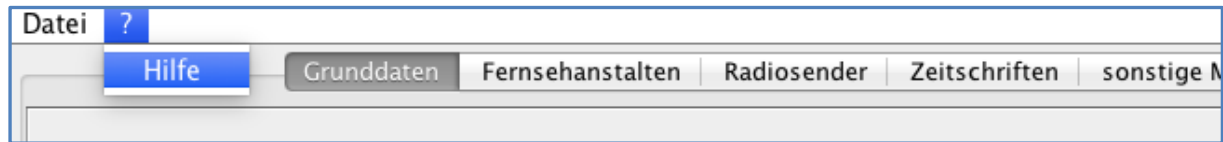


Abbildung 4: Implementierte Hilfefunktion

In der Klasse „Menue“ (Zeile 150 bis 157) wurde das „Fragezeichen“ als Symbol in der Menüleiste erstellt, welches als Option zur Hilfefunktion dient und die Nutzer bei Schwierigkeiten unterstützen soll. Außerdem besteht eine Unterleiste als „Hilfe“. Nach dem Anwählen der Hilfe öffnet sich der ausgearbeitete Text. Dieser ist in der Abbildung 9 ersichtlich. Der angezeigte Inhalt wurde von den Projektmitgliedern sowie alle anderen Komponente zur Hilfefunktion selbst verfasst.

Die Benutzeroberfläche wurde bereits mit JAVA Swing programmiert. Aus diesem Grund wurde die Hilfefunktion ebenso auf dieser Weise erarbeitet und implementiert.

```
150 // Menüelemente erzeugen
151 JMenu hilfe = new JMenu("?");
152 //Untermenüelemente erzeugen
153 hilfe.add(item = new JMenuItem("Hilfe"));
154 item.setActionCommand(String.valueOf(HILFE));
155 item.addActionListener(this);
156
157 menuBar.add(hilfe);
```

Abbildung 5: Code zur Implementierung

In der Klasse „WerbeBudgetApplication“ (Zeile 699 bis 807) wurde die hilfe-Methode programmiert. Zunächst wurde ein Fenster (Dialog) mit entsprechendem Titel „Hilfefunktion zur Werbebudgetoptimierung 1.1“ erzeugt und die Größe hierzu festgelegt. Im Anschluss wurde die Überschrift „Bedienungsanleitung“ hinzugefügt. Nach dem Erzeugen des erforderlichen Gerüsts wurde der Textinhalt durch ein JLabel verfasst, das den Vorteil hat mit HTML zu programmieren, um den Text optisch schöner zu gestalten. Auch die Position des Labels musste festgesetzt werden. Erforderlich war



eine zusätzliche vertikale und horizontale Scroll-Funktion, da der Textinhalt größer als des Dialoges war.

[illegible]

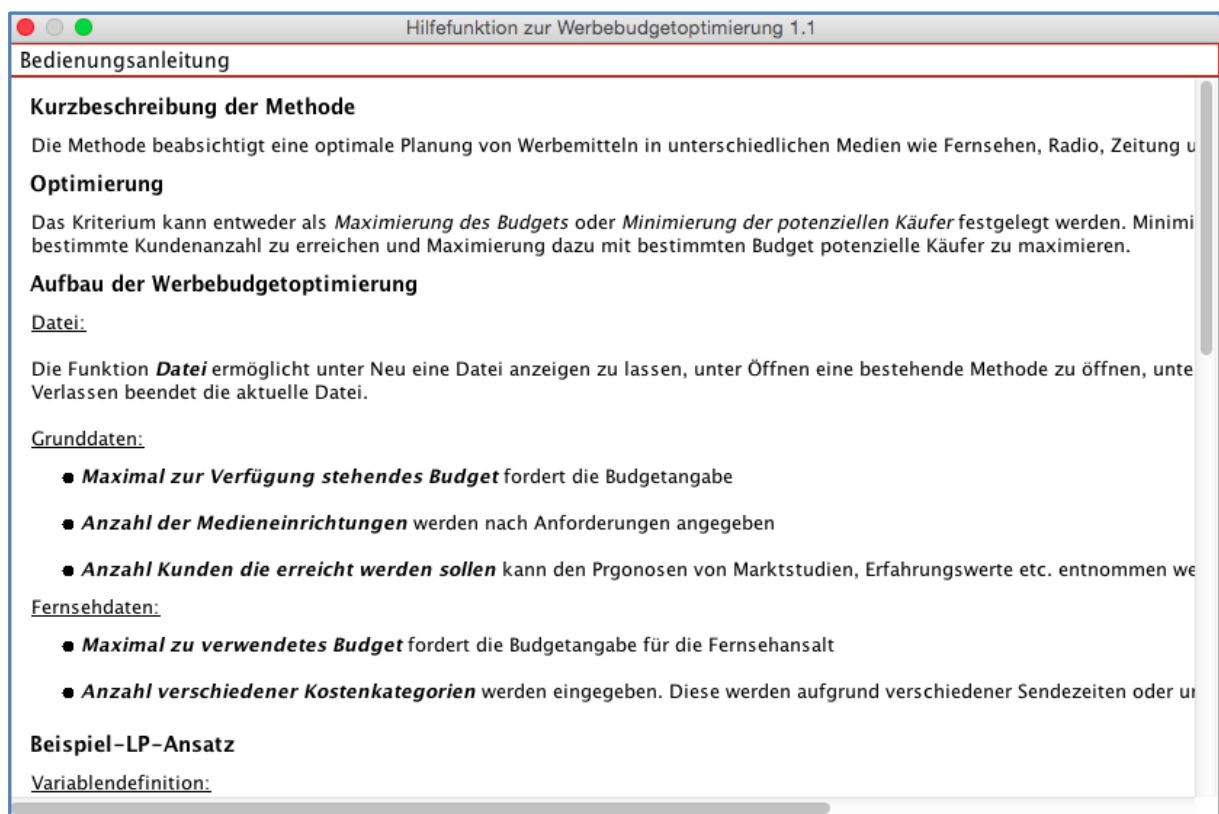
### Abbildung 6: Hilfe-Methode im Code Teil 1

[illegible]

### Abbildung 7: Hilfe-Methode im Code Teil 2

[illegible]

### Abbildung 8: Hilfe-Methode im Code Teil 3



### Abbildung 9: Inhalt der Hilfestellung

- Die Daten in den Kostenkategorien zu speichern, so dass beim Einsehen der Budgeteingrenzung in den Medien nicht verloren gehen, ergab größere Schwierigkeiten. Dies wurde im Analysebericht präzise erläutert.