



## JAVA STARS 2006 - Sun Microsystems Award

Projekttitlel:

Mathematik-Trainer – Bestimmung von Ableitungs- und  
Stammfunktionen

# Übersicht

Bundesland	NRW
------------	-----

Teamnummer	241
------------	-----

Schulnummer	194025
-------------	--------

Schulname	Städtisches Gymnasium Delbrück
-----------	--------------------------------

Schulform	Gymnasium – Sekundarstufe II
-----------	------------------------------

Name des Teams	Delbruecker
----------------	-------------

Projektname	Mathematik-Trainer – Bestimmung von Ableitungs- und Stammfunktionen
-------------	---

Projektkurzbeschreibung (max. 4 Zeilen)	Mit dem Mathetrainer lassen sich Ableitungs- und Stammfunktionen zu nahezu beliebigen Funktionen bestimmen. Zudem werden computergenerierte Aufgaben zu normalen Polynomfunktionen und zu spezifischen Regelanwendungen gestellt und müssen vom Anwender gelöst werden.
---	---

Unterrichtsfach	Informatik – Grundkurs 12
-----------------	---------------------------

Gruppengröße	fünf
--------------	------

Alter der Schüler/innen	17-19 Jahre
-------------------------	-------------

teilnehmende Mädchen	keine
----------------------	-------

# Inhaltsverzeichnis

<b>1</b>	<b>ALLGEMEINE FRAGEN ZUM WETTBEWERB.....</b>	<b>4</b>
1.1	WIE HABEN SIE/HABT IHR VON DEM WETTBEWERB "JAVA STARS 2006" ERFAHREN?.....	4
1.2	HABT IHR AN ANDEREN WETTBEWERBEN TEILGENOMMEN?.....	4
1.3	BEI WELCHEN WETTBEWERBEN HABT IHR BISHER DAS PROGRAMM SCHON VORGELEGT? .....	4
<b>2</b>	<b>PROJEKTDESCHEIBUNG .....</b>	<b>5</b>
2.1	PROJEKTDESCHEIBUNG .....	5
2.2	PROJEKTIDEE .....	5
2.3	THEMA.....	5
2.4	UNTERRICHTSFACH.....	5
2.5	NUTZEN FÜR DEN UNTERRICHT.....	5
2.6	ZEITAUFWAND .....	5
2.7	ARBEITSTEILUNG .....	5
2.8	ARBEITSUMGEBUNG .....	6
2.9	TESTUMGEBUNG .....	6
2.10	INSTALLATION / START DES PROGRAMMS .....	6
<b>3</b>	<b>LÖSUNGSKONZEPT.....</b>	<b>7</b>
3.1	AUFBAU DER LÖSUNG.....	7
3.2	EINGESETZTE VERFAHREN .....	7
<b>4</b>	<b>PROGRAMM-ARCHITEKTUR .....</b>	<b>8</b>
4.1	ÜBERSICHT .....	8
4.2	GUI: .....	8
4.2.1	<i>Methoden</i> .....	8
4.3	ABLEITHILFEN: .....	9
4.3.1	<i>Methoden</i> .....	9
4.4	POLYNOM: .....	11
4.4.1	<i>Methoden</i> .....	11
4.5	KLAMMERPOLYNOM:.....	12
4.5.1	<i>Methoden</i> .....	12
4.6	BRUCHPOLYNOM, MULTIPLIKATIONSPOLYNOM UND TRIGONOMETRISCHESPOLYNOM: .....	12
4.7	FUNKTIONSERSTELLER: .....	12
4.7.1	<i>Zustände</i> .....	12
4.7.2	<i>Methoden</i> .....	12
<b>5</b>	<b>BENUTZERSCHNITTSTELLE .....</b>	<b>15</b>
5.1	EINFÜHRUNG: .....	15
5.2	FUNKTIONSRECHNER: .....	16
5.2.1	ABLEITEN VON FUNKTIONEN: .....	16
5.2.2	BILDEN EINER STAMMFUNKTION VON FUNKTIONEN: .....	17
5.3	TRAININGSBEREICH: .....	17
5.3.1	ÜBUNGSBEREICH: .....	18
5.3.2	TESTBEREICH: .....	20
<b>6</b>	<b>ANLAGEN.....</b>	<b>25</b>
6.1	DOKUMENTIERTER QUELLKODE .....	25
6.2	EIDESSTATTLICHE ERKLÄRUNG .....	25
6.3	EINVERSTÄNDNISERKLÄRUNG DER ERZIEHUNGSBERECHTIGTEN .....	25

# **1 Allgemeine Fragen zum Wettbewerb**

---

## **1.1 *Wie haben Sie/habt ihr von dem Wettbewerb "JAVA STARS 2006" erfahren?***

Unser Informatiklehrer Herr Kempe erzählte uns von dem Wettbewerb und ermutigte uns daran teilzunehmen. Weitere Informationen bekamen wir von der Internetseite des Projekts

## **1.2 *Habt Ihr an anderen Wettbewerben teilgenommen?***

Unsere Gruppe hat bisher an keinem anderen Wettbewerb teilgenommen. Dies ist unsere erste Teilnahme.

## **1.3 *Bei welchen Wettbewerben habt Ihr bisher das Programm schon vorgelegt?***

Das Programm ist eine vollständige Neuentwicklung und wurde speziell für den Wettbewerb entwickelt und programmiert.

## 2 Projektbeschreibung

---

### 2.1 Projektbeschreibung

Das Programm kann die Ableitungs- bzw. Stammfunktion vieler Funktionen bestimmen. Fest integriert ist ein Funktionsgenerator, der Funktionen zu unterschiedlichen Bereichen erstellt. Darunter fallen zum einen ganz normale Polynomfunktionen, aber auch Funktionen in denen bestimmte Ableitungsregeln (Kettenregel, Produktregel, Quotientenregel) angewandt werden müssen. Zudem gibt es die Möglichkeit vermischte Funktionen erstellen zu lassen um verschiedene Regeln gleichzeitig zu üben. Die Lösung wird natürlich kontrolliert und bewertet, ferner ist ein Test mit einer festen Aufgabenanzahl möglich.

### 2.2 Projektidee

Die Projektidee ist uns nach der letzten Matheklausur gekommen. Einige Lehrer merkten an, dass viele Schüler Nachholungsbedarf beim Bilden von Ableitungen und Stammfunktionen hätten. Damit war die Grundlage für unser Projekt geschaffen.

### 2.3 Thema

Das Programm hat die Bestimmung von Ableitungs- bzw. Stammfunktionen zur Themengrundlage.

### 2.4 Unterrichtsfach

Das Programm findet Anwendung im Mathematik-Unterricht. Es werden grundlegende Arbeitsweisen der Analysis geübt.

### 2.5 Nutzen für den Unterricht

Das Programm bietet den Vorteil, dass es immer wieder zufällige und somit vollständig unterschiedliche Funktionen generiert, die dann gelöst werden müssen. Zudem können auch spezielle Regelanwendungen trainiert werden.

### 2.6 Zeitaufwand

Das Team traf sich zweimal wöchentlich im Informatikunterricht. Zusätzlich konnte man sich in den Unterrichtspausen über Probleme etc. austauschen. Vor den Weihnachtsferien wurde größtenteils die Planung des Programms betrieben. Anfängliche Klassen wurden programmiert. Die Arbeit wurde in den Weihnachtsferien intensiviert und lief auch nach den Ferien effektiv weiter.

### 2.7 Arbeitsteilung

Anfänglich wurde folgende Einteilung vorgenommen:

- Benutzerschnittstelle: Marco Rossi
- Bestimmen der Ableitungsfunktion und sonstige Funktionen: Andreas Lücke
- Bestimmen der Stammfunktion: Marius Dransfeld

- Vergleich der Benutzereingabe mit der Lösung: Michael Keimeier
- Generieren von Funktionen zu den unterschiedlichen Typen: Simon Mehringskötter

Bei Fertigstellung der jeweiligen Aufgabe wurde dann meistens getestet und die gefunden Fehler der Gruppe mitgeteilt.

## 2.8 *Arbeitsumgebung*

Wir nutzen Windows-Computer mit installiertem BlueJ oder Eclipse.

## 2.9 *Testumgebung*

Wir testen das Programm sowohl auf Windows-Computern in der Schule als auch zu Hause. Dabei nutzen wir die Java SDK 1.4.2 und 1.5.0

## 2.10 *Installation / Start des Programms*

Um das Programm starten zu können, benötigt man eine installierte Java-Version 1.4.2 oder höher. Dann genügt das Ausführen der Datei „Mathematik-Trainer.jar“

## 3 Lösungskonzept

---

### 3.1 Aufbau der Lösung

Als Anwender erhalten Sie eine Oberfläche, die leicht zu verstehen ist. Sie starten in einem Hauptmenü, welches Sie mit je einem Klick in die Teilbereiche der Software leitet. Zum einen gibt es einen Funktionsrechner, der es Ihnen ermöglicht die von Ihnen gestellt Funktionen abzuleiten bzw. eine mögliche Stammfunktion zu bilden. Der zweite Bereich dient zur Übung der Ableitungsregeln und Bildung von Stammfunktionen. Es gibt keine aufwendig verschachtelten Menüs. Sie gelangen schnell von einer Menüseite zur anderen. Um Sie stets daran zu erinnern, was bei den Eingaben beachtet werden muss steht in kurzen Stichpunkten alles Wichtige über dem Eingabefeld. Als Anwender haben Sie somit eine komfortable Navigation in unserer Swing Oberfläche und finden sich immer und überall zu Recht, da es immer einen Weg zurück gibt und die Buttons so beschriftet sind, dass Ihnen direkt klar ist, was bei einer Betätigung geschieht. Eine intuitive und einfach gehaltene Benutzeroberfläche ist die Grundlage des Projektes, um den Fokus auf die Funktionen und das Training zu legen. Zudem soll auch jeder unerfahrene Schüler das Programm einfach nutzen können.

### 3.2 Eingesetzte Verfahren

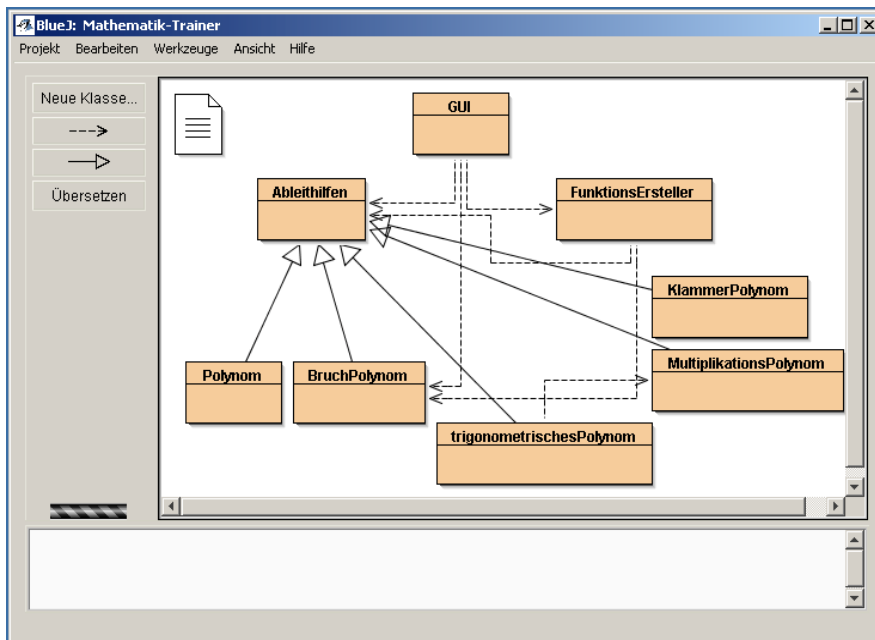
Zur Erarbeitung der Lösung haben wir auf die Ableitungsregeln und die Regeln, die zur Bildung einer Stammfunktion notwendig sind, zurückgegriffen. Diese sind uns aus dem Unterricht bekannt und sind auch in Formelsammlungen wieder zu finden. Gemäß den Berechnungsformeln werden die Funktionen berechnet und mit ihnen gearbeitet. Die vielen Klassen bringen eine grobe Struktur in das Programm, die dafür sorgt, dass wir den Durchblick behalten können. Deshalb sind alle Klassen mit der Klasse „Ableithilfen“ verknüpft, da in dieser die wichtigsten Methoden zur Bildung der Stamm- und Ableitungsfunktion zu finden sind. Die GUI (Graphical User Interface) ist die Klasse, die das ganze Projekt startet. In ihr wird das Anzeigefenster erzeugt, über welches die Benutzereingaben entgegengenommen und gesteuert werden. Diese Klasse holt sich die Funktionen, sorgt dafür, dass sie abgeleitet werden bzw. die Stammfunktion gebildet wird, überprüft die Eingabe mit der Lösung und gibt die für den Anwender relevanten Informationen aus. Uns war es wichtig, dass wir eine Klasse haben, die alles steuert und dafür sorgt, dass das, was der Anwender möchte, auch in die Tat umgesetzt wird.

### 3.3 Anmerkung

Es kann vorkommen, dass die Ermittlung der Richtigkeit der eigenen Ableitung fehlerhaft ist, da dieser Teil nicht vollkommen fehlerfrei ist. Aus zeitlichen Gründen konnten wir diese Fehler leider nicht mehr entfernen.

# 4 Programm-Architektur

## 4.1 Übersicht



## 4.2 GUI:

Die Klasse „GUI“ ist die Benutzeroberfläche für den Anwender. Sie verwaltet alle Elemente, die zur Darstellung benötigt werden und nutzt die Methoden der anderen Klassen um die oben beschriebenen Funktionen des Mathetrainers ausführen zu können.

### 4.2.1 Methoden

Die Klasse stellt folgende Methoden zu Verfügung:

- `public static void main(String[] args)`

Diese Methode wird beim Aufruf der „.jar“-Datei benötigt. Sie erzeugt die Klasse und damit das Fenster der Benutzeroberfläche

- `private void setMainWindow(int newWindow)`

Der Inhalt des Ausgabefensters wird geändert und so mit einem neuen Bildschirm gefüllt

- `private void setScreen_1()`

Es wird der Inhalt eines Fensters erstellt, in dem man zwischen der Funktion des automatischen Bestimmens von Ableitungs- bzw. Stammfunktionen gewählt werden kann.

- `private void setScreen_11()`

Es wird der Inhalt eines Fensters erstellt, in dem man von nahezu beliebiger Funktion die Ableitungs- bzw. Stammfunktion bilden lässt.

- *private void setScreen\_12()*

Es wird der Inhalt eines Fensters erstellt, in dem man die Art der Übung wählen kann. Zur Auswahl stehen unbegrenzt ausführbare Übungen und anzahlbegrenzte Tests

- *private void setScreen\_121(String pregel, boolean ptyp, boolean pcounter, int ptotalnumber)*

Es wird der Inhalt eines Fensters erstellt, in dem man eine generierte Funktion gestellt bekommt und dann die Lösung bestimmen muss. Die Eingabe wird auf Richtigkeit überprüft und dementsprechend wird eine Ausgabe erstellt. Dieser Screen wird für die Übungen als auch für die Tests benutzt. Je nachdem welches von beiden ausgeführt wird, wird die Aufteilung des Fensters leicht verändert.

- *private void setScreen\_122(int pcorrectnumber, int ptotalnumber)*

Der Auswertungsbildschirm mit Daten über die Anzahl der korrekt gelösten Aufgaben.

- *private String function\_facelift(String function)*

Die Funktion wird für Ausgabe mit „html“ konvertiert. Das ".0" nach Zahlen wird entfernt. Exponenten werden nicht mehr durch "^" angezeigt, sondern erscheinen als wirklich Hochzahl wie z.B.  $x^2$

- *private boolean vergleichePolynom(Polynom p1, Polynom p2)*

Diese Funktion gibt zurück, ob die beiden Polynome p1 und p2 gleich sind.

- *private boolean ArrayListIstGleich(ArrayList a1, ArrayList a2)*

Diese Funktion gibt zurück, ob die beiden ArrayLists a1 und a2 gleich sind.

### 4.3 Ableithilfen:

Die Klasse Ableithilfen, ist eine Methodensammlung, für andere Klassen um leichter aus einem String eine Funktion zu gewinnen, eine Funktion ab- bzw. aufzuleiten oder eine bestehende Funktion für die Ausgabe zu einen String zu konvertieren.

#### 4.3.1 Methoden

- *public BigDecimal potenz(BigDecimal pBasis, int pExponent)*

Diese Methode liefert die Potenz einer Dezimalzahl

- *public String zum\_String(ArrayList Funktion)*

Diese Methode macht eine Funktion zu einem String für die Ausgabe

- *public ArrayList Funktion\_ableiten(ArrayList Funktion)*

Diese Methode bekommt eine Funktion als Parameter, die sie ableitet und zurückgibt

- *public ArrayList Funktionenpotenz(ArrayList Funktion, String potenz)*

Diese Methode potenziert eine Funktion. Liegt ein gerader Exponent vor, wird versucht auszumultiplizieren, ansonsten wird alles in einem Klammerpolynom gespeichert.

- *public ArrayList Funktionenmultiplikation(ArrayList Funktion, Object Pol)*  
Multipliziert eine Funktion mit einem Polynom und liefert das Ergebnis als ArrayList zurück.
- *public ArrayList Funktionenmultiplikation(ArrayList Funktion, ArrayList Funktion2)*  
Multipliziert eine Funktion mit einer anderen und liefert das Ergebnis als ArrayList zurück. Es wird bei geeigneten Polynomen versucht auszumultiplizieren, ansonsten wird ein MultiplikationsPolynom erstellt.
- *public ArrayList Funktionsubtraktion(ArrayList Funktion1, ArrayList Funktion2)*  
Diese Funktion zieht von der Funktion1 die Funktion2 ab und gibt das Ergebnis zurück. Dafür wird die 2.Funktion mit (-1) mal genommen und dann Funktionaddition verwendet.
- *public ArrayList Funktionendivision(ArrayList Funktion1, ArrayList Funktion2)*  
Diese Funktion dividiert Funktion1 durch Funktion2. Es wird versucht zu kürzen, ansonsten wird ein BruchPolynom erstellt.
- *public ArrayList Funktionaddition(ArrayList Funktion1, ArrayList Funktion2)*  
Diese Methode addiert zu Funktion1 Funktion2 und gibt das Ergebnis zurück. Wenn möglich werden gleiche Exponenten zusammengefasst.
- *public ArrayList Teilfunktionen\_zusammensetzen(String text)*  
Teilfunktion\_bilden --> Teilfunktion\_auswerten --> !!! Teilfunktion\_zusammensetzen !!!  
Liest aus einem String die eingegebene Funktion aus.
- *public ArrayList Teilfunktion\_auswerten(String text)*  
Teilfunktion\_bilden --> !!! Teilfunktion\_auswerten !!! --> Teilfunktionen\_zusammensetzen  
Diese Methode nimmt einen String entgegen, der keine Klammern besitzt, und macht daraus eine Funktion, die als ArrayList zurückgegeben wird.
- *public ArrayList Teilfunktion\_bilden(String text)*  
!!! Teilfunktion\_bilden !!! --> Teilfunktion\_auswerten --> Teilfunktion\_zusammensetzen  
Diese Methode dient dazu einzelne, kleine Polynome auszulesen und als Funktion zurückzugeben.
- *public int nächstes\_Rechenzeichen(String text, int position)*  
Diese Methode geht den String durch und schaut nach Zeichen, die keine Zahl, kein Punkt, x, oder Hochzeichen sind. Zurückgegeben wird die Position des Zeichens
- *public int Klammerpaar(String text, int index)*  
Diese Methode findet zu einer angegebenen Klammer den passenden Partner und liefert dessen Position zurück.

- *public String Stringvorbereiten(String text)*  
Diese Methode dient dazu den String vorzubereiten, damit z.B. -- zu + wird und somit nicht für Teilfunktionen\_zusammensetzen ungültig erscheint. Zurückgegeben wird der modifizierte String.
- *public boolean Rechenzeichenprüfer(String text)*  
Diese Methode prüft, ob die Rechenzeichen z.B. richtig angewendet worden sind. Ist alles korrekt, wird true zurückgegeben, ansonsten false.
- *public boolean Klammerprüfer(String text)*  
Diese Methode prüft, ob es für jede offene Klammer auch eine geschlossene gibt. Ist die Eingabe gültig, wird true zurückgegeben, ansonsten false.
- *public long parsen(String text)*  
Wenn „2.0“ eingegeben wurde, empfindet das Integer.parseInt als ungültig, diese Methode umgeht dieses Problem und kann auch so dafür eingesetzt werden zu prüfen, ob es sich um eine Ganzzahl handelt.
- *public String Bruch(BigDecimal DezimalZahl)*  
Diese Methode dient dazu eine Dezimalzahl in einen Bruch umzuwandeln. Findet sie keinen angemessenen Bruch, wird die Dezimalzahl gekürzt als String zurückgegeben. Teile dieser Methode sind von der folgenden Website von Basic nach Java umprogrammiert worden:  
<http://www.arstechnica.de/index.html?name=http://www.arstechnica.de/computer/msoffice/vba/vba0033.html>
- *public boolean aufleitbar(ArrayList Funktion)*  
Diese Methode testet ob eine übergebene Funktion aufleitbar ist. Zur Zeit nur möglich mit einfachen Polynomfunktionen.
- *public ArrayList Funktion\_aufleiten(ArrayList)*  
Diese Methode bestimmt zu einer übergebenen Funktion eine mögliche Stammfunktion . Vorher muss mit *aufleitbar(Funktion)* getestet werden, ob die Funktion auch aufleitbar ist.

#### 4.4 Polynom:.

Die Klasse Polynom dient zur Abspeicherung eines einfachen Polynoms der Form:  $ax^b$

Es gibt einmal den Koeffizienten und Exponenten des Polynoms vor der Ableitung als Eigenschaftsvariable und einmal ein Polynom für danach

##### 4.4.1 Methoden

Außer der Methode Ableiten(), die keinen Rückgabe Wert hat und nur dazu dient die Eigenschaftsvariable für nach der Ableitung zu füllen, hat diese Klasse nur get- und set-Methoden

- *public float getErgebnis(int basis)*

Diese Methode liefert das Produkt aus Koeffizient und Potenz, welche sich aus der Basis „basis“ und dem Exponenten zusammensetzt.

## 4.5 *KlammerPolynom*:

Diese Klasse dient dazu Konstrukte der Art  $a(\text{Funktion})^b$  entgegenzunehmen und zu speichern!

Auch hier werden der Exponent, die Funktion und der Koeffizient als Eigenschaftsvariable gehalten, auf die über die `get-` und `setMethoden` zugegriffen werden kann.

### 4.5.1 *Methoden*

Auch hier gibt es die Methode `Ableiten()`, die den gleichen Zweck hat wie bei `Polynom`

## 4.6 *BruchPolynom, MultiplikationsPolynom und trigonometrischesPolynom*:

`BruchPolynom`, `MultiplikationsPolynom` und `trigonometrischesPolynom` sind vom Grundsatz her genauso aufgebaut wie `Polynom` und `Klammerpolynom`: Es gibt die Eigenschaftsvariablen für die aktuellen Daten und Eigenschaftsvariablen für nach der Ableitung, die durch `Ableitung()` gefüllt werden.

## 4.7 *FunktionsErsteller*:

Die Klasse stellt verschiedene Methoden zur Verfügung, mit der die Funktionen für das Programm generiert werden.

### 4.7.1 *Zustände*

- *private BruchPolynom eBruchFunktion*  
beinhaltet die Bruchfunktion für die Quotientenregel
- *private MultiplikationsPolynom eMultiplikationsFunktion*  
beinhaltet die Multiplikationsfunktion für die Produktregel
- *private KlammerPolynom eKlammerFunktion*  
beinhaltet die Klammerfunktion für die Kettenregel
- *private trigonometrischesPolynom eTrigonometrischesPolynom*  
beinhaltet die Trigonometrischefunktion für die Kettenregel

### 4.7.2 *Methoden*

- *public FunktionsErsteller()*  
Dies ist der Konstruktor, der dafür sorgt, dass ein Objekt der Klasse `FunktionsErsteller` erzeugt wird. In ihm werden alle Eigenschaftsvariablen auf null gesetzt.
- *public ArrayListFunktionErstellen(String pRegel)*  
Dies ist die Hauptfunktion der Klasse. Man übergibt im Parameter eine Regel. Die Möglichkeiten lauten „Kettenregel“, „Produktregel“, „Quotientenregel“, „Normalespolynom“, „Gemischt“. Je nach Parameter werden Funktionen generiert. Hierbei werden die restlichen Funktionen der Klasse beansprucht.
- *public void BruchFunktionErstellen(int temp1, int temp2)*

Diese Methode generiert eine Bruchfunktion. Hierzu werden mit den Methoden „PolynomFunktion1Erstellen(int pAnzSum)“ und „PolynomFunktion2Erstellen(int pAnzSum)“ erst mal zwei Polynomfunktionen erstellt. Die erste ist dann der Zähler, die zweite ist der Nenner. Das Objekt der Klasse „BruchPolynom“ wird dann in der Eigenschaftsvariable „eBruchFunktion“ gespeichert. Man benötigt die Brüche für die Quotientenregel.

- *public void MultiplikationsFunktionErstellen(int temp1, int temp2)*

Diese Methode generiert eine Multiplikationsfunktion. Hierzu werden mit den Methoden „PolynomFunktion1Erstellen(int pAnzSum)“ und „Polynomfunktion2Erstellen(int pAnzSum)“ erst mal zwei Polynomfunktionen erstellt. Die erste ist dann die erste Funktion, die dann mit der zweiten multipliziert wird. Das Objekt der Klasse „MultiplikationsPolynom“ wird dann in der Eigenschaftsvariable „eMultiplikationsFunktion“ gespeichert. Dies wird für die Produktregel benötigt.

- *public void KlammerFunktionErstellen(float pExp, int temp1)*

Diese Methode generiert eine Klammerfunktion. Diese Klammerfunktion hat einen Vorfaktor, eine innere Funktion, die durch die Methode „PolynomFunktion1Erstellen(int pAnzSum)“ erstellt wird und einen Exponenten für die Klammer. Das Objekt der Klasse „KlammerPolynom“ wird in der Eigenschaftsvariable „eKlammerFunktion“ gespeichert. Dies wird für die Kettenregel benötigt.

- *public void TrigonometrischeFunktionErstellen(int temp1)*

Diese Methode generiert eine trigonometrische Funktion. Hier gibt es auch einen Vorfaktor, eine innere Funktion, die durch die Methode „PolynomFunktion1Erstellen(int pAnzSum)“ erstellt wird. Diese Funktion ist dann entweder in der Klammer vom Sinus oder vom Kosinus. Das Objekt der Klasse „trigonometrischesPolynom“ wird in der Eigenschaftsvariable „eTrigonometrischeFunktion“ gespeichert. Dies wird für die Kettenregel benötigt.

- *public ArrayList PolynomFunkton1Erstellen(int pAnzSum)*

Diese Methode generiert die Funktionen, die in allen Objekten verarbeitet werden. Die Funktionen, die hier heraus kommen, werden in der Bruchfunktion, Multiplikationsfunktion, Trigonometrischenfunktion, Klammerfunktion und für die normalen Polynome. Der Parameter, der in anderen Methoden oft durch „temp1“ und „temp2“ gekennzeichnet ist gibt die Anzahl der Summanden des Polynoms an.

- *public ArrayList PolynomFunktion2Erstellen(int pAnzSum)*

siehe *public ArrayList PolynomFunktion1Erstellen(int pAnzSum)*

- *public BruchPolynom getBruchFunktion()*

gibt die Eigenschaftsvariable „eBruchFunktion“ zurück

- *public MultiplikationsPolynom getMultiplikationsFunktion()*

gibt die Eigenschaftsvariable „eMultiplikationsFunktion“ zurück

- *public KlammerPolynom getKlammerFunktion()*

gibt die Eigenschaftsvariable „eKlammerFunktion“ zurück

- *public trigonometrischesPolynom getTrigonometrischeFunktion()*

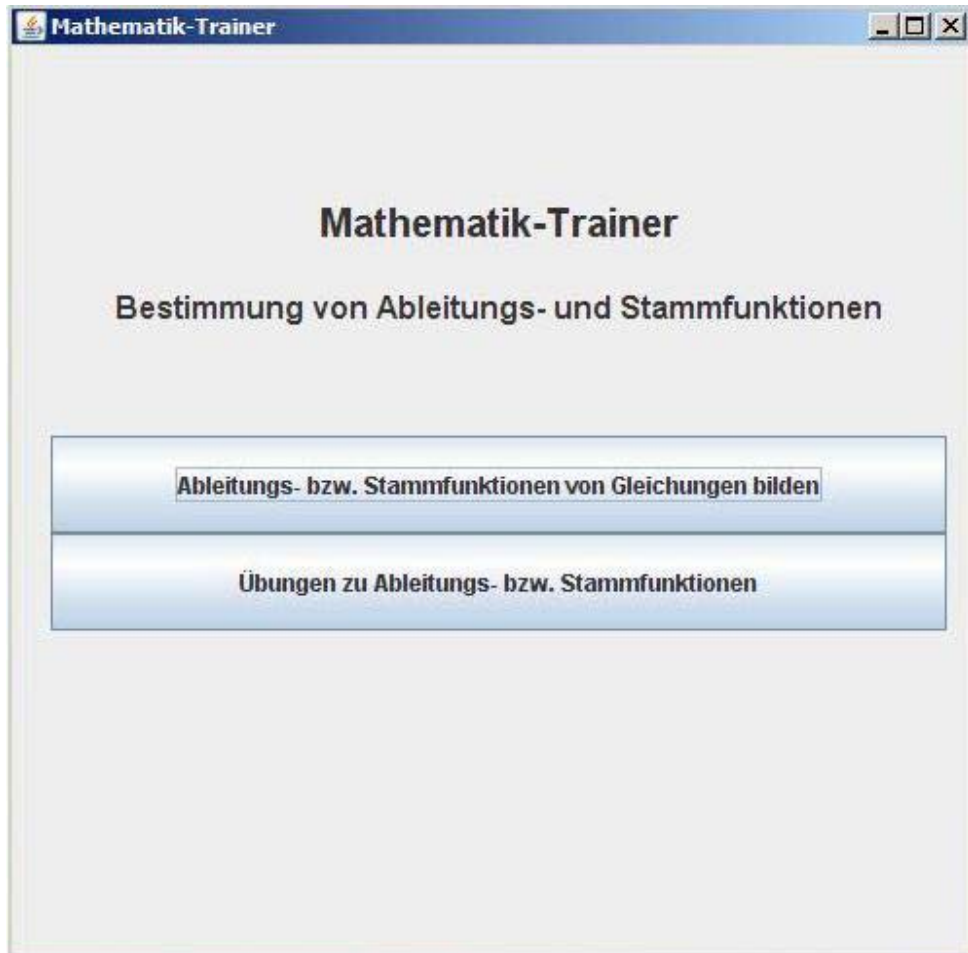
gibt die Eigenschaftsvariable „eTrigonometrischesPolynom“ zurück

## 5 Benutzerschnittstelle

---

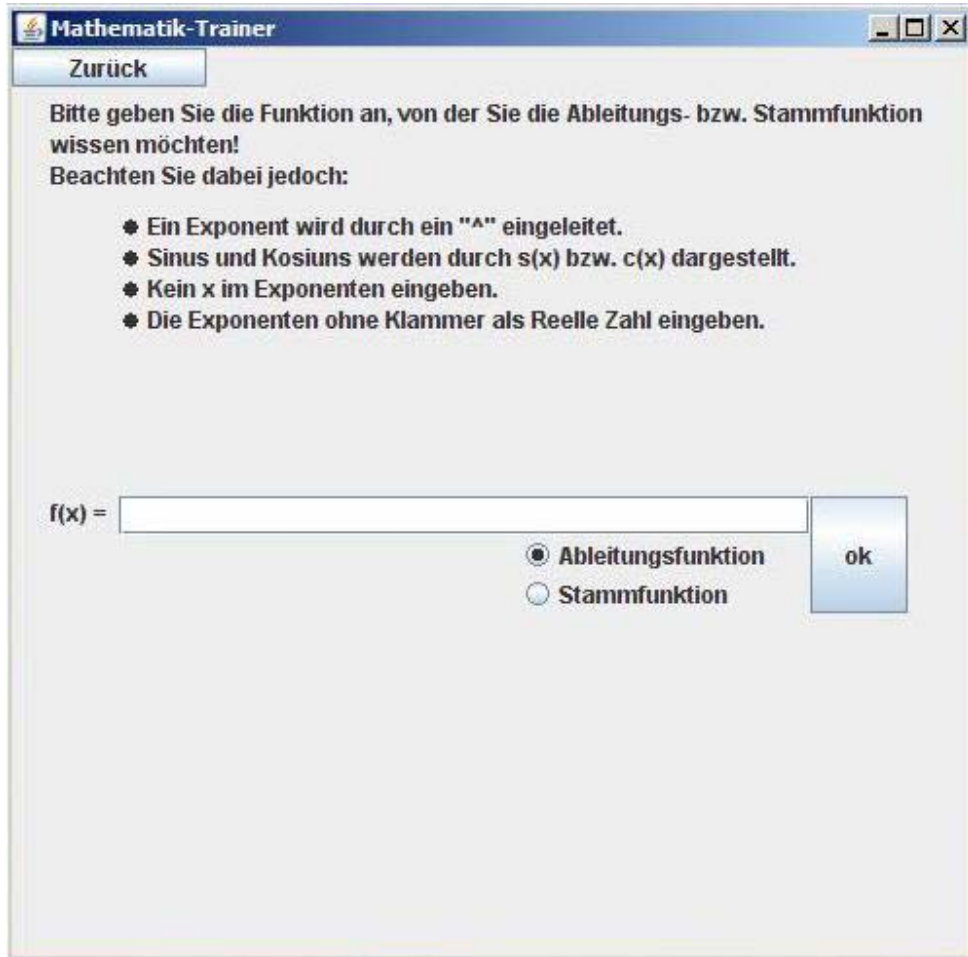
### 5.1 Einführung:

Nach dem Start der Lernsoftware gelangen Sie in das Hauptmenü des Programms.



Sie haben nun die Möglichkeit einen der beiden Hauptteile der Software auszuwählen. Über die erste Schaltfläche kommen Sie in einen „Funktionsrechner“. Mit diesem Funktionsrechner können Sie Funktionen ableiten bzw. eine mögliche Stammfunktion bilden lassen. Genaueres können Sie im 2. Kapitel „Funktionsrechner“ nachlesen. Die zweite Schaltfläche führt in den Trainingsbereich. Dort können sie Tests machen, bei denen Sie anschließend eine Auswertung erhalten, oder Sie versuchen sich an einigen zufällig generierten Funktionen. Weitere Informationen zum Trainingsbereich erhalten Sie im 3. Kapitel „Trainingsbereich“.

## 5.2 Funktionsrechner:



Der Funktionsrechner bietet Ihnen die Möglichkeit Funktionen abzuleiten oder eine mögliche Stammfunktion zu bilden. Dies hat den Vorteil, dass mühsames Ableiten nach Ketten-, Produkt-, und/oder Quotientenregel nicht mehr nötig ist, da dieses vom Programm erledigt wird. Auch eine mögliche Stammfunktion ist auf leichte Weise zu finden, wobei dieses Feature auf „normale Polynome“ vom Typ  $f(x) = ax^n \pm bx^{(n-1)} \pm cx^{(n-2)} \pm dx^{(n-3)} \dots$ ) beschränkt ist. Mit dem Schaltfläche „Zurück“ gelangen Sie wieder in das Hauptmenü.

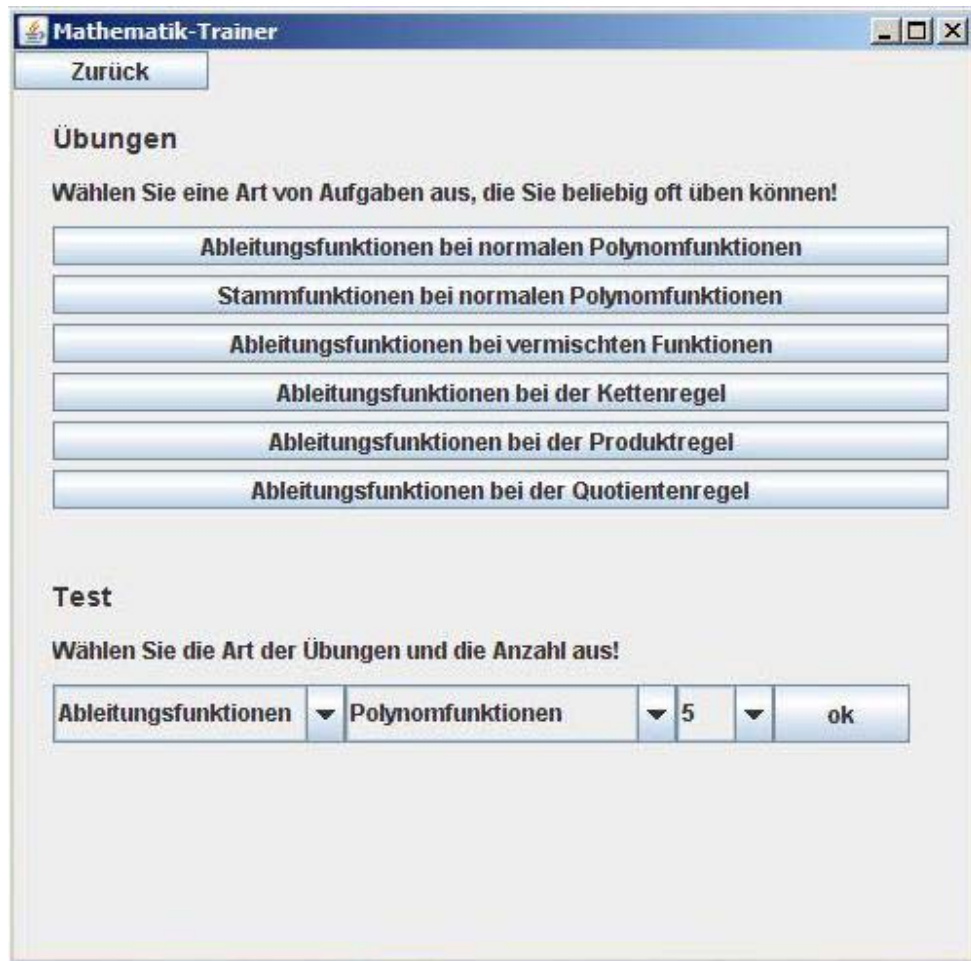
### 5.2.1 Ableiten von Funktionen:

Um eine Funktion abzuleiten müssen Sie einfach die Funktion in das Feld eingeben. Eine mögliche Eingabe wäre z.B.  $2/3x^3+2x^2-4x+1$ . Durch einen Klick auf die Schaltfläche „ok“ wird Ihnen die Funktion abgeleitet unten angezeigt. Die Regeln, die Sie bei der Eingabe beachten müssen sind über dem Eingabefeld notiert.

### 5.2.2 Bilden einer Stammfunktion von Funktionen:

Stammfunktionen bilden Sie ähnlich wie die Ableitungsfunktionen. Sie müssen lediglich den Punkt von „Stammfunktion“ auswählen. Durch einem Klick auf die Schaltfläche „ok“ erhalten Sie eine mögliche Stammfunktion zu ihrer eingegebenen Funktion.

### 5.3 Trainingsbereich:



Im Trainingsbereich haben Sie die Möglichkeit Ihr Können durch Übungen oder Tests zu verbessern. Deshalb ist dieser Bereich auch in zwei grundlegende Teile aufgeteilt. Es gibt zum einen den Übungsteil, in dem Sie zu verschiedenen Regeln Funktionen ableiten können, bzw. eine mögliche Stammfunktion bestimmen sollen. Andererseits gibt es den Testbereich. Dort können Sie verschiedene Funktionen hintereinander lösen und erhalten hinterher eine Auswertung. Dies bietet Ihnen die Chance herauszufinden an welchen Stellen Ihre Probleme liegen, um diese dann im Übungsbereich zu verkleinern.

### 5.3.1 Übungsbereich:

Zu erst müssen Sie eines der Übungsgebiete auswählen. Zur Auswahl stehen:

- Ableitungsfunktionen bei normalen Polynomfunktionen
- Stammfunktionen bei normalen Polynomfunktionen
- Ableitungsfunktionen bei vermischten Funktionen
- Ableitungsfunktionen bei der Kettenregel
- Ableitungsfunktionen bei der Produktregel
- Ableitungsfunktionen bei der Quotientenregel

Nach einem Klick auf einen der Auswahlbuttons gelangen Sie in den passenden Themenbereich.

Mathematik-Trainer

Zurück Neue Aufgabe erstellen

Beachten Sie bei der Eingabe!

- Ein Exponent wird durch ein "^" eingeleitet.
- Sinus und Kosinus werden durch s(x) bzw. c(x) dargestellt.
- Kein x im Exponenten eingeben.
- Die Exponenten ohne Klammer als Reelle Zahl eingeben.

Bestimmen Sie die erste Ableitung folgender Funktion:

$f(x) = -(16/5)x^2$

$f'(x) =$   ok

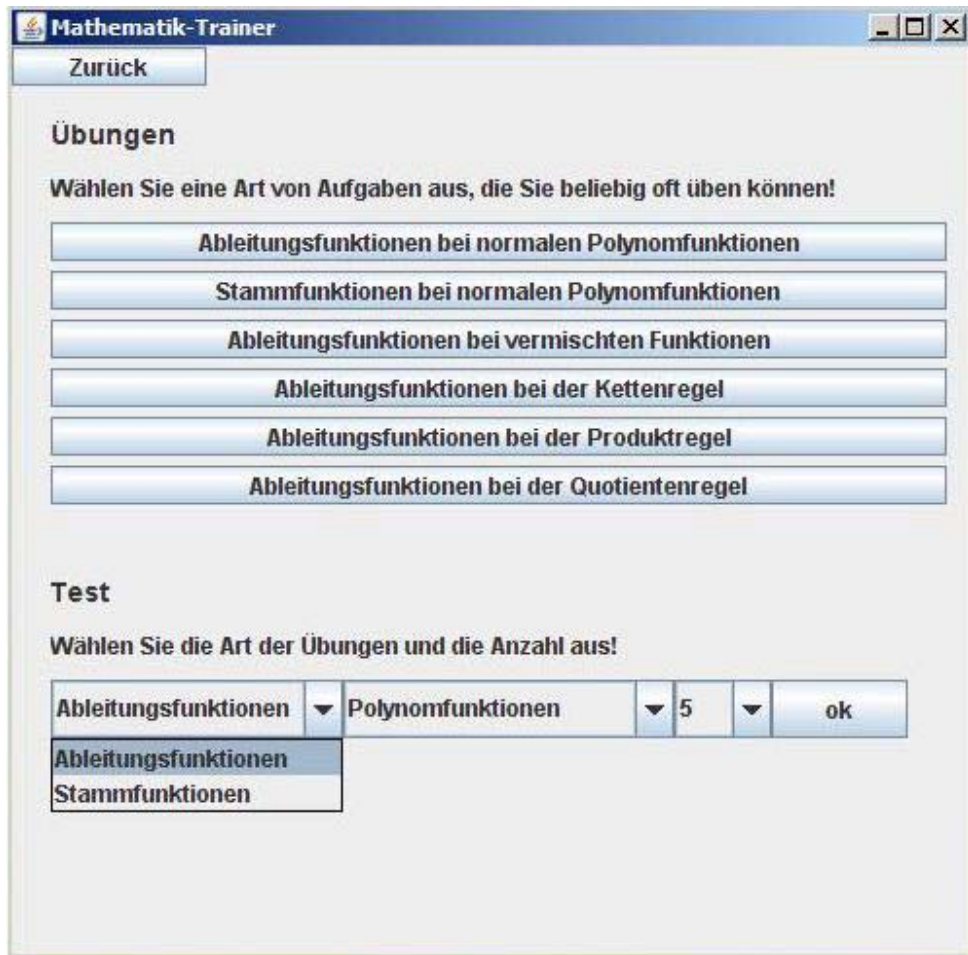
Hier sehen Sie die Funktion  $f(x) = -(19/5)x^8$ , welche Sie ableiten sollen. Das Ergebnis schreiben Sie einfach in das Eingabefeld. Hierbei müssen Sie die darüber stehenden Anmerkungen beachten. Sie müssten also  $-(152/5)x^7$  eingeben. Auch  $-30,4x^7$  wäre eine mögliche Eingabe. Mit einem Klick auf „ok“ bestätigen Sie Ihre Eingabe und erfahren nun ob Ihr Ergebnis richtig oder falsch ist.



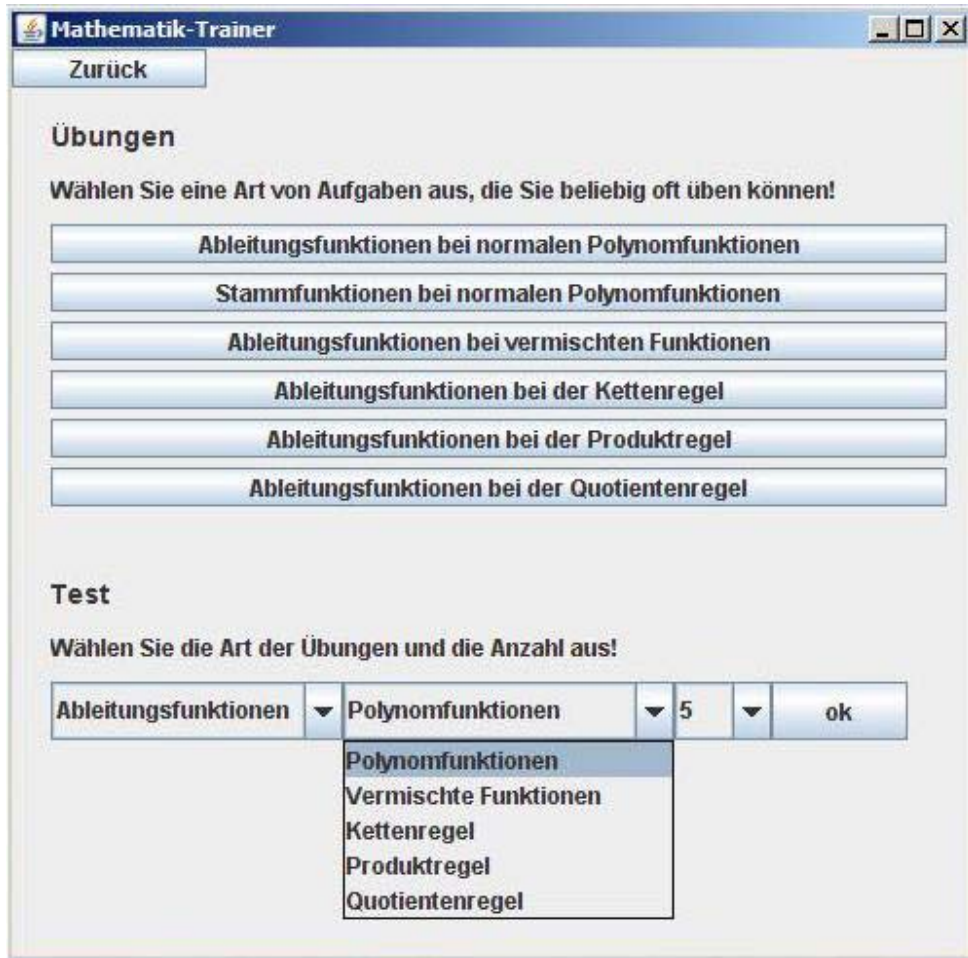
Nach einem Klick auf den Button „Neue Aufgabe erstellen“ erhalten Sie eine neue Aufgabe. Über den Knopf „Zurück“ gelangen Sie in das vorherige Menü.

### 5.3.2 Testbereich:

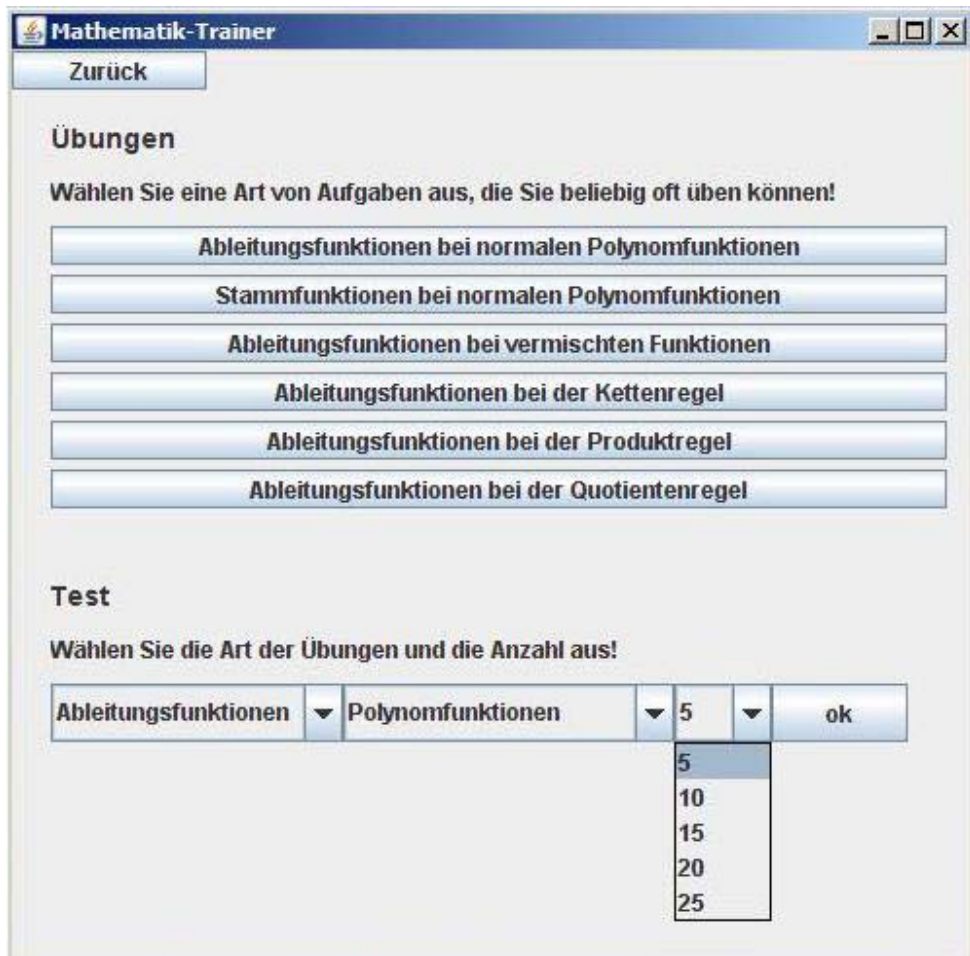
Um einen Test zu starten müssen Sie unten verschiedene Einstellungen vornehmen. Über das erste Auswahlfeld haben Sie die Möglichkeit fest zu legen ob sie Ableitungsfunktionen oder Stammfunktionen bilden möchten.



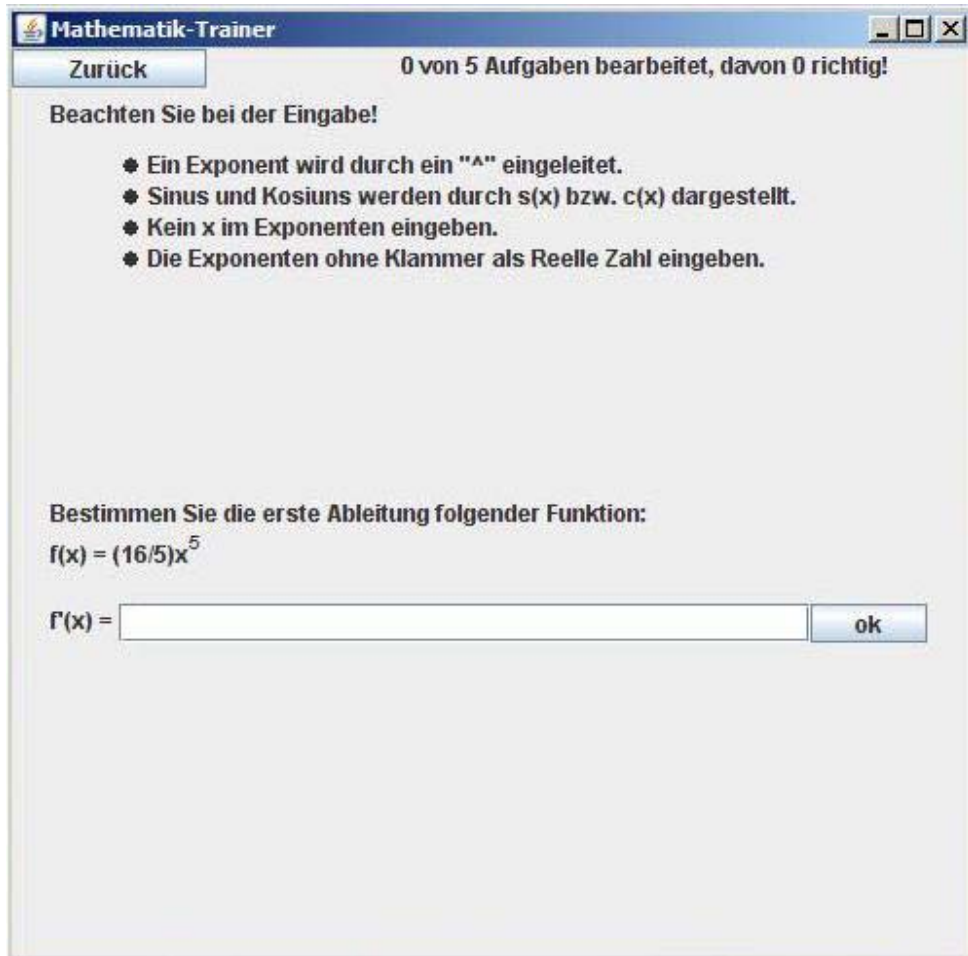
Nun müssen Sie im zweiten Feld die Regel wählen, in der Sie getestet werden möchten.



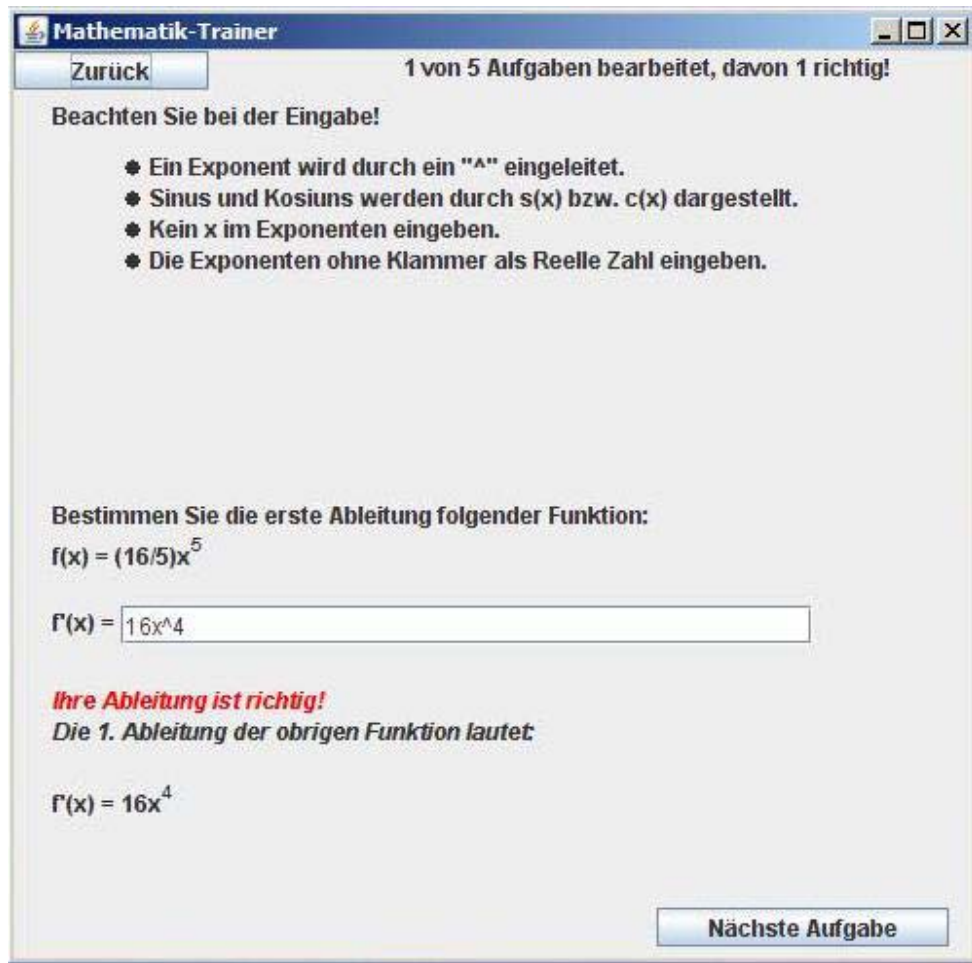
Anschließend müssen Sie die Anzahl der Fragen bestimmen.



Nach einem Klick auf den Knopf „ok“ wird der Test gestartet und Sie erhalten wieder eine Funktion, deren Ableitung Sie bestimmen sollen.



Die Funktion  $f(x) = -(45/2)x^5 - (5/2)x^2$  soll in unserem Beispiel abgeleitet werden. Die Ableitung müssen Sie nun in das Eingabefeld eintragen. dabei ist es wichtig, dass Sie die über dem Feld notierten Anmerkungen beachten. In Unserem Fall würde die benötigte Eingabe  $-112,5x^4-5x$  lauten, auch  $-225/2x^4-10/2x$  wäre eine mögliche Eingabe. Mit einem Klick auf „ok“ bestätigen Sie ihr Ergebnis und erfahren nun, ob sie richtig gelegen haben oder nicht.



Oben haben Sie eine Übersicht, die Ihnen sagt wie viele Aufgaben Sie von der Ihnen gewählten Anzahl schon erledigt sind. Auch die momentane Anzahl an richtigen Lösungen wird oben aufgelistet. Sie können jetzt über einen Klick auf den Knopf mit der Aufschrift „Nächste Aufgabe“ zur nächsten Funktion gehen und diese dann lösen. Zum Schluss erhalten Sie eine Auswertung, die Ihnen sagt wie viele Fragen Sie richtig und falsch beantwortet haben.

## **6 Anlagen**

---

6.1 *Dokumentierter Quellcode*

6.2 *Eidesstattliche Erklärung*

6.3 *Einverständniserklärung der Erziehungsberechtigten*