

Softwarepraktikum 2003

Buchführungssystem

Projektdokumentation
(Teilbeleg 2)

Teamleiter: Dominique Petersen

Mitglieder des Projektteams:

Annette Keusch
Nadine Pollmer
Dirk Brückner
Marco Döring

Praktikumsbetreuer: Christian Meichsner

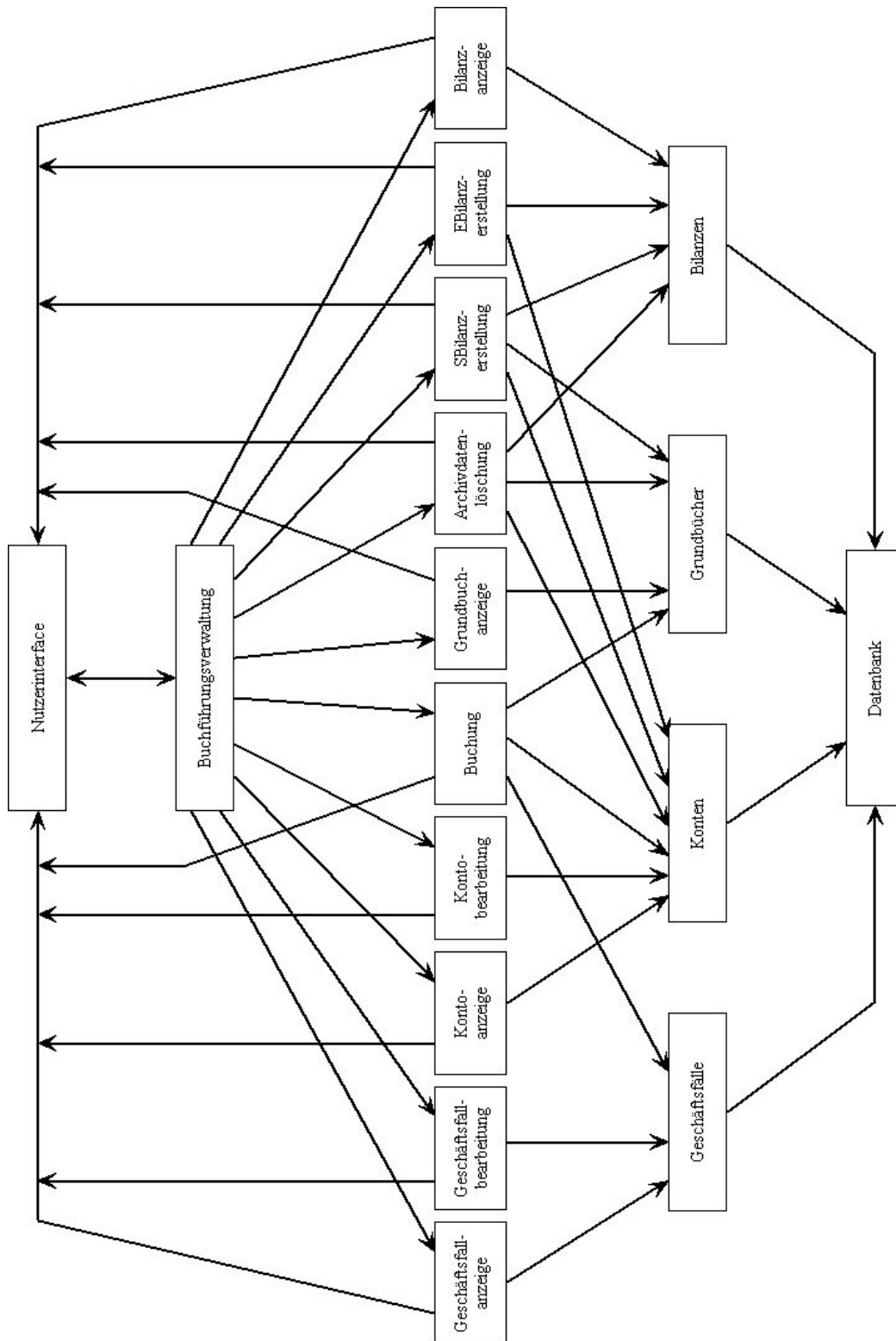
Chemnitz, den 05.06.2003

Inhaltsverzeichnis

1	Entwurfsdokumentation	3
1.1.	Entwurf der Systemarchitektur.....	3
1.2.	Definition der Leistungen der Architekturkomponenten	4
1.2.1	Modul Datenbanken	4
1.2.2	Modul Geschäftsfälle	7
1.2.3	Modul Konten	9
1.2.4	Modul Grundbücher	12
1.2.5	Modul Bilanzen	15
1.2.6	Modul Geschäftsfallanzeige	18
1.2.7	Modul Geschäftsfallbearbeitung	19
1.2.8	Modul Kontoanzeige	20
1.2.9	Modul Kontobearbeitung	21
1.2.10	Modul Buchung.....	23
1.2.11	Modul Grundbuchanzeige	24
1.2.12	Modul Archivdatenlöschung	25
1.2.13	Modul SBilanzerstellung.....	27
1.2.14	Modul EBilanzerstellung	29
1.2.15	Modul Bilanzanzeige.....	30
1.2.16	Modul Parameter	31
1.2.17	Modul Fehlerbehandlung	37
1.2.18	Modul Hilfe	38
1.2.19	Modul Buchführungsverwaltung.....	39
1.2.20	Modul Nutzerinterface	41

1 Entwurfsdokumentation

1.1 Entwurf der Systemarchitektur



Verschiedene weitere Hilfsmodule (z.B. zur Bereitstellung von Datentypen, für die Onlinehilfe und zur Fehlerbehandlung) sind aus Gründen der Übersichtlichkeit nicht im obigen Diagramm enthalten.

1.2 Definition der Leistungen der Architekturkomponenten (Modulspezifikationen)

1.2.1 Modul Datenbank

Modul: **Datenbank** **Bearbeiter:** Dominique Petersen

Bezeichner: MDatenbank **Änderungsstand:** 02.06.03

Kurzbeschreibung: Stellt alle Funktionen für den Zugriff auf die MySQL-Datenbank bereit.

Export: DB_connect
DB_close
DB_QuerySend
DB_QueryGrepNumberAffectedRows
DB_QueryGrepInsertID
DB_QueryGrepRow
DB_QueryFetchArray

Import: T_Error - aus Modul Fehlerbehandlung

DB_connect: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Die Funktion versucht, eine dauerhafte Verbindung zur MySQL-Datenbank aufzubauen. Bei Erfolg wird FehlerNr = 0 zurückgeliefert, ansonsten ein entsprechend anderer Wert.

DB_close: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: bestehende Verbindung zur MySQL-Datenbank

Inhalt: Die Funktion versucht, die Verbindung zur MySQL-Datenbank zu schließen. Bei Erfolg wird FehlerNr = 0 zurückgeliefert, ansonsten ein entsprechend anderer Wert.

DB_QuerySend:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
DB_Datenbank	in	Char[80]	Verwendete Datenbank (z.B. Bilanzen)
DB_Query	in	Char[80]	Query, der an die Datenbank geschickt wird
DB_ResID	out	Integer	Referenz auf das Ergebnis der Abfrage

Voraussetzung: bestehende Verbindung zur MySQL-Datenbank

Inhalt: Sendet einen Query an den MySQL-Datenbankserver. Der Rückgabewert ist eine Ressource-ID, mittels derer auf das Ergebnis der Abfrage zugegriffen werden kann.

DB_QueryGrepNumberAffectedRows:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
DB_ResID	in	integer	Referenz auf das Ergebnis der Abfrage
DB_AffectedRows	out	integer	Anzahl der von DB_ResID beeinflussten Zeilen

Voraussetzung: bestehende Verbindung zur MySQL-Datenbank,
DB_ResID ist ein gültiger Rückgabewert von DB_QuerySend

Inhalt: Liefert die Anzahl der Datenbankzeilen, die durch die letzte Operation betroffen waren.

DB_QueryGrepRow:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
DB_ResID	in	integer	Referenz auf das Ergebnis der Abfrage
DB_Row	out	array	Beinhaltet eine Zeile von dem Datensatz, auf den DB_ResID zeigt (liefert 0 nach dem letzten Datensatz)

Voraussetzung: bestehende Verbindung zur MySQL-Datenbank ,
DB_ResID ist ein gültiger Rückgabewert von DB_QuerySend

Inhalt: Liefert die nächste Zeile einer Abfrage als assoziatives Array zurück. Gibt 0 (FALSE) nach dem letzten Datensatz.

DB_QueryFetchArray:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
DB_ResID	in	integer	Referenz auf das Ergebnis der Abfrage
DB_Array	out	array	liefert den nächsten Datensatz einer Abfrage als assoziatives Array zurück

Voraussetzung: bestehende Verbindung zur MySQL-Datenbank ,
DB_ResID ist ein gültiger Rückgabewert von DB_QuerySend

Inhalt: Liefert den nächsten Datensatz einer Abfrage als assoziatives Array zurück.

1.2.2 Modul Geschäftsfälle

Modul:	Geschaeftsfaelle	Bearbeiter:	Annette Keusch
Bezeichner:	MGeschaeftsfaelle	Änderungsstand:	01.06.03
Kurzbeschreibung:	Stellt alle Funktionen zur Verwaltung der persistenten Geschäftsfalldaten bereit.		
Export:	GF_new GF_get GF_find GF_delete		
Import:	DB_connect DB_close DB_QuerySend DB_QueryGrepNumberAffectedRows DB_QueryGrepRow DB_QueryFetchArray T_Error T_GF_Vorgangsdaten	- aus Modul Datenbank - aus Modul Datenbank - aus Modul Datenbank - aus Modul Datenbank - aus Modul Datenbank - aus Modul Fehlerbehandlung - aus Modul Parameter	

GF_new: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
GF_Nr	In	integer	Nummer des Geschäftsfalls
GF_Name	In	char[80]	Name des Geschäftsfalls
GF_Vorgangsdaten	In	T_GF_Vorgangsdaten	Vorgangsdaten des Geschäftsfalls
FehlerNr	Out	T_Error	Fehlercode

Voraussetzung: ein Geschäftsfall mit GF_Nr existiert noch nicht

Inhalt: Die Funktion erstellt aus GF_Nr, GF_Name und GF_Vorgangsdaten einen neuen Geschäftsfalldatensatz und speichert diesen in der Datenbank ab. Zurückgeliefert wird die Fehlernummer, die gleich Null ist, wenn alles erfolgreich gespeichert wurde.

GF_get: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
GF_Nr	Inout	integer	Nummer des Geschäftsfalls
GF_Name	Out	char[80]	Name des Geschäftsfalls
GF_Vorgangsdaten	Out	T_GF_Vorgangsdaten	Vorgangsdaten des Geschäftsfalls
FehlerNr	Out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Die Funktion sucht in der Datenbank nach dem Geschäftsfall mit der GF_Nr. Wird er gefunden, so werden alle Speicherdaten des Geschäftsfalls (GF_Nr, GF_Name, GF_Vorgangsdaten) und die FehlerNr = 0 zurückgegeben. Ist der Geschäftsfall nicht vorhanden, wird nur die entsprechende FehlerNr zurückgegeben.

GF_find:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
GF_Nr	Out	integer	Nummer des Geschäftsfalls
GF_Name	Out	char[80]	Name des Geschäftsfalls
FehlerNr	Out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Die Funktion durchsucht die Datenbank und liefert ein Array der Elemente GF_Nr und GF_Name aller existierenden Geschäftsfälle zurück. Ist mindestens ein Geschäftsfall gefunden worden und es sind keine Fehler aufgetreten, so wird außerdem FehlerNr = 0 zurückgegeben, ansonsten eine entsprechende FehlerNr.

GF_delete:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
GF_Nr	In	integer	Nummer des Geschäftsfalls
FehlerNr	Out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Die Funktion sucht in der Datenbank nach dem Geschäftsfall mit der GF_Nr. Wird er gefunden, so werden alle Speicherdaten des Geschäftsfalls (GF_Nr, GF_Name, GF_Vorgangsdaten) aus der Datenbank gelöscht und die FehlerNr = 0 zurückgegeben. Ist der Geschäftsfall nicht vorhanden oder der Geschäftsfall konnte nicht gelöscht werden, wird nur die entsprechende FehlerNr zurückgegeben.

1.2.3 Modul Konten

Modul:	Konten	Bearbeiter:	Annette Keusch
Bezeichner:	MKonten	Änderungsstand:	02.06.03
Kurzbeschreibung:	Stellt alle Funktionen zur Verwaltung der persistenten Kontodaten bereit.		
Export:	Kto_new Kto_get Kto_find Kto_delete Kto_insertContentRow Kto_changeFlag		
Import:	DB_connect DB_close DB_QuerySend DB_QueryGrepNumberAffectedRows DB_QueryGrepInsertID DB_QueryGrepRow T_Error T_Kto_Art T_Datumsflag T_Kto_Inhalt T_Kto_Seite T_Buchungseintrag	- aus Modul Datenbank - aus Modul Datenbank - aus Modul Datenbank - aus Modul Datenbank - aus Modul Datenbank - aus Modul Datenbank - aus Modul Fehlerbehandlung - aus Modul Parameter - aus Modul Parameter - aus Modul Parameter - aus Modul Parameter	

Kto_new: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
Kto_Nr	In	integer	Nummer des Kontos
Kto_Name	In	char[80]	Name des Kontos
Kto_Art	In	T_Kto_Art	Art des Kontos („aktiv“, „passiv“, „SBK“ oder „EK“)
Kto_Datumsflag	In	T_Datumsflag	Gibt an, ob das Konto schon abgeschlossen oder noch gültig ist
Kto_Wertigkeit	In	integer	Gibt die Priorität des Kontos an
OberKto_Nr	In	integer	Nummer des Oberkontos
FehlerNr	Out	T_Error	Fehlercode

Voraussetzung: Kto_Nr > 0000, Konto mit OberKto_Nr als Kto_Nr existiert bereits (außer bei OberKto_Nr = 0000)

Inhalt: Wenn noch kein Konto mit der Kto_Nr in der Datenbank existiert, erstellt die Funktion einen neuen Kontodatensatz mit Kto_Nr, Kto_Name, Kto_Art, Kto_Datumsflag, Kto_Wertigkeit, OberKto_Nr und dem (noch leeren) Kto_Inhalt und speichert den Datensatz in der Datenbank ab. Verläuft alles korrekt, wird die FehlerNr = 0 zurückgegeben, ansonsten der jeweilige Fehlercode.

Kto_get:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
Kto_Nr	Inout	integer	Nummer des Kontos
Kto_Name	Out	char[80]	Name des Kontos
Kto_Art	Out	T_Kto_Art	Art des Kontos („aktiv“, „passiv“, „SBK“ oder „EK“)
Kto_Datumsflag	Inout	T_Datumsflag	Gibt an, ob das Konto schon abgeschlossen oder noch gültig ist
Kto_Wertigkeit	Out	integer	Gibt die Priorität des Kontos an
OberKto_Nr	Out	integer	Nummer des Oberkontos
Kto_Inhalt	Out	T_Kto_Inhalt	Beinhaltet die Buchungseinträge und Summen der Soll- und Habenseite
FehlerNr	Out	T_Error	Fehlercode

Voraussetzung: Kto_Nr > 0000

Inhalt:

Die Funktion sucht in der Datenbank nach dem Konto mit der Kto_Nr und dem Kto_Datumsflag. Wird das Konto gefunden, so wird der gesamte Kontodatensatz (bestehend aus Kto_Nr, Kto_Name, Kto_Art, Kto_Datumsflag, Kto_Wertigkeit, OberKto_Nr, Kto_Inhalt) und die FehlerNr = 0 zurückgegeben. Ist das Konto nicht vorhanden, wird nur die entsprechende FehlerNr zurückgegeben.

Kto_find:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
Kto_Nr	Inout	integer	Nummer des Kontos
Kto_Name	Inout	char[80]	Name des Kontos
Kto_Art	In	T_Kto_Art	Art des Kontos („aktiv“, „passiv“, „SBK“ oder „EK“)
Kto_Datumsflag	Inout	T_Datumsflag	Gibt an, ob das Konto schon abgeschlossen oder noch gültig ist
Kto_DatumsflagMax	In	T_Datumsflag	gibt spätestes Abschlussdatum der zu suchenden Konten an
Kto_Wertigkeit	In	integer	Gibt die Priorität des Kontos an
OberKto_Nr	in	integer	Nummer des Oberkontos
Kto_leer	In	integer	wenn Kto_leer != 0 werden Kto_SeitenSumme beider Kontoseiten auf 0 überprüft (zum Finden leerer Konten)
FehlerNr	Out	T_Error	Fehlercode

Voraussetzung: in-Parameter sind jeweils optional, Kto_Datumsflag und Kto_DatumsflagMax treten nicht gleichzeitig auf

Inhalt:

Die Funktion durchsucht die Datenbank und liefert ein Array der Elemente Kto_Nr, Kto_Name und Kto_Datumsflag aller die Kriterien (angegebene in-Parameter) erfüllenden Konten zurück. Ist mindestens ein Konto gefunden worden und es sind keine Fehler aufgetreten, so wird FehlerNr = 0 zurückgegeben, ansonsten eine entsprechende FehlerNr.

Kto_delete:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
Kto_Nr	in	integer	Nummer des Kontos
Kto_Datumsflag	in	T_Datumsflag	Gibt an, ob das Konto schon abgeschlossen oder noch gültig ist
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: Konto muss leer oder älter als 10 Jahre sein

Inhalt: Die Funktion sucht in der Datenbank nach dem Konto mit der Kto_Nr und dem Kto_Datumsflag. Wird das Konto gefunden, so wird der gesamte Kontodatensatz (bestehend aus Kto_Nr, Kto_Name, Kto_Art, Kto_Datumsflag, Kto_Wertigkeit, OberKto_Nr, Kto_Inhalt) gelöscht und die FehlerNr = 0 zurückgegeben. Ist das Konto nicht vorhanden oder es trat ein Fehler beim Löschen auf, wird nur die entsprechende FehlerNr zurückgegeben.

Kto_insertContentRow:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
Kto_Nr	in	integer	Nummer des Kontos
Kto_Seite	in	T_Kto_Seite	identifiziert die Kontoseite, auf der die Buchung einzutragen ist
Buchungseintrag	in	T_Buchungseintrag	einzutragende Buchung
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: Konto muss gültig sein (noch nicht abgeschlossen)

Inhalt: Die Funktion sucht das aktuelle Konto mit Kto_Nr in der Datenbank. Ist das Konto vorhanden, speichert es den neuen Buchungseintrag auf der durch Kto_Seite bestimmten Seite des Kto_Inhalts, aktualisiert die betroffene Kto_SeitenSumme und gibt die FehlerNr = 0 zurück. Existiert das Konto nicht oder ein Fehler bei der Speicherung ist erfolgt, so liefert die Funktion den entsprechenden Fehlercode zurück.

Kto_changeFlag:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
Kto_Nr	in	integer	Nummer des Kontos
Kto_Datumsflag	in	T_Datumsflag	aktuelles Datumsflag des Kontos
Kto_DatumsflagNew	in	T_Datumsflag	neues Datumsflag des Kontos
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: Konto ist noch nicht endgültig abgeschlossen

Inhalt: Die Funktion sucht in der Datenbank nach dem Konto mit Kto_Nr und Kto_Datumsflag. Wird das Konto gefunden, so wird das in seinem Datensatz gespeicherte Kto_Datumsflag durch Kto_DatumsflagNew ersetzt und die FehlerNr = 0 zurückgegeben. Ist das Konto nicht vorhanden oder es trat ein Fehler auf, wird eine entsprechende FehlerNr != 0 zurückgegeben.

1.2.4 Modul Grundbücher

Modul:	Grundbuecher	Bearbeiter:	Annette Keusch
Bezeichner:	MGrundbuecher	Änderungsstand:	03.06.03
Kurzbeschreibung:	Stellt alle Funktionen zur Verwaltung der persistenten Grundbuchdaten bereit.		
Export:	GB_new GB_get GB_find GB_delete GB_insertContentRow GB_changeFlag		
Import:	DB_connect DB_close DB_QuerySend DB_QueryGrepNumberAffectedRows DB_QueryGrepInsertID DB_QueryGrepRow T_Error T_Datumsflag T_GB_Inhalt T_Grundbucheintrag	 - aus Modul Datenbank - aus Modul Datenbank - aus Modul Datenbank - aus Modul Datenbank - aus Modul Datenbank - aus Modul Datenbank - aus Modul Fehlerbehandlung - aus Modul Parameter - aus Modul Parameter - aus Modul Parameter	

GB_new: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
GB_Datumsflag	in	T_Datumsflag	Gibt an, ob das Grundbuch schon abgeschlossen oder noch gültig ist
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Die Funktion speichert das Grundbuch unter dem GB_Datumsflag in der Datenbank, es besitzt noch keinen Inhalt. Verläuft alles korrekt, wird die FehlerNr = 0 zurückgegeben, ansonsten der jeweilige Fehlercode.

GB_get:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
GB_Datumsflag	inout	T_Datumsflag	Gibt an, ob das Grundbuch schon abgeschlossen oder noch gültig ist
GB_Inhalt	out	T_GB_Inhalt	Beinhaltet die einzelnen Buchungen im Geschäftsjahr
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Die Funktion sucht in der Datenbank nach dem Grundbuch mit dem GB_Datumsflag. Wird das Grundbuch gefunden, so werden GB_Datumsflag, GB_Inhalt und die FehlerNr = 0 zurückgegeben. Ist das Grundbuch nicht vorhanden, wird nur die entsprechende FehlerNr zurückgegeben.

GB_find:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
GB_Datumsflag	inout	T_Datumsflag	identifiziert das Grundbuch
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: in-Parameter optional

Inhalt: Die Funktion durchsucht die Datenbank und liefert ein Array des Elementes GB_Datumsflag aller Grundbuecher zurück, deren Datumsflag \leq GB_Datumsflag ist bzw. die überhaupt existieren (für den Fall, dass GB_Datumsflag nicht als in-Parameter angegeben ist). Ist mindestens ein Grundbuch gefunden worden und es sind keine Fehler aufgetreten, so wird FehlerNr = 0 zurückgegeben, ansonsten eine entsprechende FehlerNr.

GB_delete:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
GB_Datumsflag	in	T_Datumsflag	Gibt an, ob das Grundbuch schon abgeschlossen oder noch gültig ist
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: Grundbuch muss älter als 10 Jahre sein

Inhalt: Die Funktion sucht in der Datenbank nach dem Grundbuch mit dem GB_Datumsflag. Wird das Grundbuch gefunden, so wird sein gesamter Datensatz (GB_Datumsflag und GB_Inhalt) gelöscht und die FehlerNr = 0 zurückgegeben. Ist das Grundbuch nicht vorhanden oder es trat ein Fehler beim Löschen auf, wird nur die entsprechende FehlerNr zurückgegeben.

GB_insertContentRow:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
GB_Datumsflag	in	T_Datumsflag	Gibt an, ob das Grundbuch schon abgeschlossen oder noch gültig ist
Grundbucheintrag	in	T_Grundbucheintrag	einzutragende Buchung
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: Grundbuch muss gültig sein (d.h. es handelt sich um das laufende Grundbuch)

Inhalt: Die Funktion sucht das Grundbuch mit dem GB_Datumsflag in der Datenbank. Ist das Grundbuch vorhanden, speichert es den neuen Grundbucheintrag im GB_Inhalt des gefundenen Grundbuchs ab und gibt die FehlerNr = 0 zurück. Existiert das Grundbuch nicht oder ein Fehler bei der Speicherung ist erfolgt, so liefert die Funktion den entsprechenden Fehlercode zurück.

GB_changeFlag:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
GB_Datumsflag	in	T_Datumsflag	identifiziert das Grundbuch
GB_DatumsflagNew	in	T_Datumsflag	entspricht dem Abschlussdatum des Grundbuchs
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: bei dem Grundbuch mit GB_Datumsflag handelt es sich um das laufende Grundbuch

Inhalt: Die Funktion sucht in der Datenbank nach dem Grundbuch mit dem GB_Datumsflag. Wird das Grundbuch gefunden, so wird das in seinem Datensatz gespeicherte GB_Datumsflag durch GB_DatumsflagNew ersetzt (das Grundbuch wird dadurch abgeschlossen) und die FehlerNr = 0 zurückgegeben. Ist das Grundbuch nicht vorhanden oder es trat ein Fehler auf, wird eine entsprechende FehlerNr != 0 zurückgegeben.

1.2.5 Modul Bilanzen

Modul: **Bilanzen** **Bearbeiter:** Annette Keusch

Bezeichner: MBilanzen **Änderungsstand:** 03.06.03

Kurzbeschreibung: Stellt alle Funktionen zur Verwaltung der persistenten Bilanzdaten bereit.

Export: **Bz_new**
Bz_get
Bz_find
Bz_delete
Bz_insertContentRow

Import: **DB_connect** - aus Modul Datenbank
DB_close - aus Modul Datenbank
DB_QuerySend - aus Modul Datenbank
DB_QueryGrepNumberAffectedRows - aus Modul Datenbank
DB_QueryGrepInsertID - aus Modul Datenbank
DB_QueryGrepRow - aus Modul Datenbank
T_Error - aus Modul Fehlerbehandlung
T_B_Inhalt - aus Modul Parameter
T_B_Art - aus Modul Parameter
T_Datum - aus Modul Parameter
T_B_Seite - aus Modul Parameter
T_B_Posten - aus Modul Parameter

Bz_new: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
B_Datum	in	T_Datum	Erstellungsdatum der Bilanz
B_Name	in	char[80]	Name der Bilanz
B_Art	in	T_B_Art	Art der Bilanz, entweder „SB“ oder „EB“
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Die Funktion erstellt aus den Parametern B_Name, B_Datum und B_Art einen neuen Bilanzdatensatz und speichert die neue (noch leere) Bilanz in der Datenbank ab. Verläuft alles korrekt, wird die FehlerNr = 0 zurückgegeben, ansonsten der jeweilige Fehlercode.

Bz_get:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
B_Datum	inout	T_Datum	Erstellungsdatum der Bilanz
B_Art	inout	T_B_Art	Art der Bilanz, entweder „SB“ oder „EB“
B_Name	inout	char[80]	Name der Bilanz
B_Inhalt	out	T_B_Inhalt	Beinhaltet die Nummern, Namen und Beträge der (abgeschlossenen) Konten auf der Aktiva- und Passivseite
FehlerNr	Out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Die Funktion sucht anhand der ausgewählten Kriterien (B_Datum, B_Art und/oder B_Name) in der Datenbank nach der passenden Bilanz. Wird die Bilanz gefunden, so werden ihre gesamten Speicherdaten (B_Datum, B_Art, B_Name, B_Inhalt) und die FehlerNr = 0 zurückgegeben. Ist die Bilanz nicht vorhanden, wird nur die entsprechende FehlerNr zurückgegeben.

Bz_find:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
B_Datum	inout	T_Datum	Erstellungsdatum der Bilanz
B_DatumMax	in	T_Datum	spätestes Erstellungsdatum der zu suchenden Bilanzen
B_Art	inout	T_B_Art	Art der Bilanz, entweder „SB“ oder „EB“
B_Name	inout	char[80]	Name der Bilanz
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: in-Parameter sind jeweils optional, B_Datum und B_DatumMax treten nicht gleichzeitig auf

Inhalt: Die Funktion durchsucht die Datenbank und liefert ein Array der Elemente B_Datum, B_Art und B_Name aller die Kriterien erfüllenden Bilanzen zurück. Ist mindestens eine Bilanz gefunden worden und es sind keine Fehler aufgetreten, so wird FehlerNr = 0 zurückgegeben, ansonsten eine entsprechende FehlerNr.

Bz_delete:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
B_Datum	in	T_Datum	Erstellungsdatum der Bilanz
B_Art	in	T_B_Art	Art der Bilanz, entweder „SB“ oder „EB“
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: die Bilanz muss älter als 10 Jahre sein

Inhalt: Die Funktion sucht in der Datenbank nach der Bilanz mit dem B_Datum und der B_Art. Wird die Bilanz gefunden, so wird sie aus der Datenbank gelöscht und die FehlerNr = 0 zurückgegeben. Ist die Bilanz nicht vorhanden oder ein Fehler bei der Löschung trat auf, wird nur die entsprechende FehlerNr zurückgegeben.

Bz_insertContentRow:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
B_Datum	in	T_Datum	Erstellungsdatum der Bilanz
B_Art	in	T_B_Art	Art der Bilanz, entweder „SB“ oder „EB“
B_Seite	in	T_B_Seite	benennt diejenige Seite der Bilanz, auf der ein neuer Eintrag erfolgen soll
B_Posten	in	T_B_Posten	einzutragender Bilanzposten
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Die Funktion fügt B_Posten in die durch die Parameter vorgegebene Bilanz (auf deren durch B_Seite bestimmte Seite) in der Datenbank ein und aktualisiert B_SeitenSumme der betroffenen Bilanzseite. Verläuft alles korrekt, wird FehlerNr = 0 zurückgegeben, ansonsten der jeweilige Fehlercode.

1.2.6 Modul Geschäftsfallanzeige

Modul: **Geschäftsfallanzeige** **Bearbeiter:** Annette Keusch

Bezeichner: MGeschäftsfallanzeige **Änderungsstand:** 01.06.03

Kurzbeschreibung: Stellt eine Funktion zur Anzeige von Geschäftsfällen bereit.

Export: **GF_anzeigen**

Import: **GF_get** - aus Modul Geschaeftsfaelle
StartAnzeigen - aus Modul Nutzerinterface
AnzeigenGF - aus Modul Nutzerinterface
T_GF_Vorgangsdaten - aus Modul Parameter
T_Error - aus Modul Fehlerbehandlung

GF_anzeigen: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
GF_Nr	inout	Integer	Nummer des Geschäftsfalls
GF_Name	out	char[80]	Name des Geschäftsfalls
GF_Vorgangsdaten	out	T_GF_Vorgangsdaten	Vorgangsdaten des Geschäftsfalls
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Die Funktion ruft mit dem Parameter GF_Nr die Funktion GF_get auf und liefert deren Rückgabewerte zurück.

1.2.7 Modul Geschäftsfallbearbeitung

Modul: **Geschaeftsfallbearbeitung** **Bearbeiter:** Annette Keusch

Bezeichner: MGeschaeftsfallbearbeitung **Änderungsstand:** 02.06.03

Kurzbeschreibung: Stellt alle Funktionen zur Bearbeitung von Geschäftsfällen bereit.

Export: **GF_anlegen**
GF_loeschen

Import: **GF_new** - aus Modul Geschaeftsfaelle
GF_delete - aus Modul Geschaeftsfaelle
StartBearbeiten - aus Modul Nutzerinterface
BearbeitenGFAnlegen - aus Modul Nutzerinterface
BearbeitenGFBest - aus Modul Nutzerinterface
BearbeitenGFLoeschen - aus Modul Nutzerinterface
BearbeitenGFLoeschenBest - aus Modul Nutzerinterface
T_GF_Vorgangsdaten - aus Modul Parameter
T_Error - aus Modul Fehlerbehandlung

GF_anlegen: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
GF_Nr	inout	integer	Nummer des Geschäftsfalls
GF_Name	inout	char[80]	Name des Geschäftsfalls
GF_Vorgangsdaten	inout	T_GF_Vorgangsdaten	Vorgangsdaten des Geschäftsfalls
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Die Funktion ruft mit den übergebenen Parametern GF_Nr, GF_Name und GF_Vorgangsdaten die Funktion GF_new auf und liefert deren Rückgabewert zurück. Falls die von GF_new zurückgelieferte FehlerNr = 0 ist, werden zusätzlich die übergebenen Parameter zurückgeliefert.

GF_loeschen: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
GF_Nr	in	integer	Nummer des Geschäftsfalls
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Die Funktion ruft mit dem Parameter GF_Nr die Funktion GF_delete auf und liefert deren Rückgabewert (FehlerNr) zurück.

1.2.8 Modul Kontoanzeige

Modul:	Kontoanzeige	Bearbeiter:	Annette Keusch
Bezeichner:	MKontoanzeige	Änderungsstand:	02.06.03
Kurzbeschreibung:	Stellt eine Funktion zur Anzeige von Konten bereit.		
Export:	Kto_anzeigen		
Import:	Kto_get	- aus Modul Konten	
	StartAnzeigen	- aus Modul Nutzerinterface	
	AnzeigenKontoWaehlen	- aus Modul Nutzerinterface	
	AnzeigenKonto	- aus Modul Nutzerinterface	
	T_Datumsflag	- aus Modul Parameter	
	T_Kto_Art	- aus Modul Parameter	
	T_Kto_Inhalt	- aus Modul Parameter	
	T_Error	- aus Modul Fehlerbehandlung	

Kto_anzeigen: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
Kto_Nr	inout	Integer	Kontonummer des Kontos
Kto_Datumsflag	inout	T_Datumsflag	Datumsflag des Kontos
Kto_Name	out	Char[80]	Name des Kontos
Kto_Art	out	T_Kto_Art	Art des Kontos (aktiv, passiv, ...)
OberKto_Nr	out	Integer	Kontonummer des Oberkontos, über das das Konto abgeschlossen wird
Kto_Inhalt	out	T_Kto_Inhalt	Inhalt des Kontos
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: Kto_Nr > 0000

Inhalt: Die Funktion ruft mit den Parametern Kto_Nr und Kto_Datumsflag die Funktion Kto_get auf und liefert deren Rückgabewerte (bis auf Kto_Wertigkeit) zurück.

1.2.9 Modul Kontobearbeitung

Modul:	Kontobearbeitung	Bearbeiter:	Annette Keusch
Bezeichner:	MKontobearbeitung	Änderungsstand:	02.06.03
Kurzbeschreibung:	Stellt alle Funktionen zur Bearbeitung von Konten bereit.		
Export:	Kto_anlegen Kto_loeschen		
Import:	Kto_new	- aus Modul Konten	
	Kto_get	- aus Modul Konten	
	Kto_insertContentRow	- aus Modul Konten	
	Kto_delete	- aus Modul Konten	
	Kto_find	- aus Modul Konten	
	StartBearbeiten	- aus Modul Nutzerinterface	
	BearbeitenKonto	- aus Modul Nutzerinterface	
	BearbeitenKtoBest	- aus Modul Nutzerinterface	
	T_Error	- aus Modul Fehlerbehandlung	
	T_Datumsflag	- aus Modul Parameter	
	T_Kto_Art	- aus Modul Parameter	

Kto_anlegen: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
Kto_Nr	in	integer	Kontonummer des Kontos
Kto_Name	in	char[80]	Name des Kontos
Kto_Art	in	T_Kto_Art	Art des Kontos (aktiv, passiv, ...)
OberKto_Nr	in	integer	Kontonummer des Oberkontos, über das das Konto abgeschlossen wird
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: Kto_Nr > 0000, alle in-Parameter müssen angegeben sein

Inhalt: Die Funktion überprüft zunächst mit Kto_find, ob in der Datenbank bereits ein aktuelles Konto mit Kto_Nr existiert. Ist dem nicht so, wird mithilfe von Kto_get die Kto_Wertigkeit des angegebenen Oberkontos (außer bei OberKto_Nr = 0000) in Erfahrung gebracht, um die Kto_Wertigkeit des anzulegenden Kontos berechnen zu können. Anschließend ruft die Funktion mit sämtlichen in-Parametern, der ermittelten Kto_Wertigkeit und einem generierten Kto_Datumsflag die Funktion Kto_new auf. Handelt es sich bei dem angelegten Konto um ein Hauptkonto (Kto_Wertigkeit = 1), wird zusätzlich noch ein Anfangsbestand in Höhe von 0 mithilfe der Funktion Kto_insertContentRow in den Kto_Inhalt des angelegten Kontos eingetragen. Trat an irgendeiner Stelle ein Fehler auf, wird die entsprechende FehlerNr zurückgeliefert, ansonsten FehlerNr = 0.

Kto_loeschen: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
Kto_Nr	in	integer	Kontonummer des Kontos
Kto_Datumsflag	in	T_Datumsflag	Abschlussdatum des Kontos
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: Kto_Nr > 0000, nur Konten angegeben, deren Löschung erlaubt ist

Inhalt: Die Funktion findet mit rekursiven Aufrufen von Kto_find sämtliche direkten und indirekten (aktuellen) Unterkonten des Kontos in der Datenbank. Diese werden dann einschließlich des ursprünglich zu löschenden Kontos sukzessive mithilfe der Funktion Kto_delete aus der Datenbank gelöscht. Verlieft der Vorgang erfolgreich, wird FehlerNr = 0 zurückgeliefert, ansonsten eine entsprechend andere Fehlernummer.

1.2.10 Modul Buchung

Modul:	Buchung	Bearbeiter:	Annette Keusch
Bezeichner:	MBuchung	Änderungsstand:	05.06.03
Kurzbeschreibung:	Stellt eine Funktion zum Buchen auf Konten und Grundbuechern bereit.		
Export:	buchen		
Import:	GB_insertContentRow - aus Modul Grundbuecher Kto_insertContentRow - aus Modul Konten StartBuchung - aus Modul Nutzerinterface BuchungDirekt - aus Modul Nutzerinterface BuchungGF - aus Modul Nutzerinterface BuchungBest - aus Modul Nutzerinterface T_Datum - aus Modul Parameter T_Kontenliste - aus Modul Parameter T_Error - aus Modul Fehlerbehandlung		

buchen: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
BuchungsDatum	in	T_Datum	Datum der Buchung
Beleg_Kommentar	in	Char[80]	Kommentar zur Buchung
Kontenliste	in	T_Kontenliste	notwendige Buchungsinformationen
Beleg_ID	out	Char[20]	identifiziert eine Buchung, zusammengesetzt aus Datum und Zeit
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: alle durch Kontenliste gegebenen Konten sind existent und gültig,
Summe der Soll-Beträge = Summe der Haben-Beträge

Inhalt: Die Funktion generiert eine Beleg_ID und stellt aus Buchungsdatum, Beleg_ID, Beleg_Kommentar und Kontenliste einen Grundbucheintrag zusammen, mit dem die Funktion GB_insertContentRow aufgerufen wird, um den Eintrag in den GB_Inhalt des laufenden Grundbuchs in der Datenbank zu speichern (Buchung aufs Grundbuch). Anschließend wird für jeden Kontenlisteneintrag der Kontenliste wie folgt verfahren: aus der generierten Beleg_ID und dem im Kontenlisteneintrag angegebenen Buchungsbetrag wird ein Buchungseintrag erstellt, der zusammen mit der im Kontenlisteneintrag bezeichneten Kto_Nr und betr_Kto_Seite an die Funktion Kto_insertContentRow übergeben wird (Buchung aufs Konto). Trat bei diesen Aktionen kein Fehler auf, wird FehlerNr = 0 und die generierte Beleg_ID zurückgegeben, ansonsten lediglich eine entsprechend andere Fehlernummer.

1.2.11 Modul Grundbuchanzeige

Modul: Grundbuchanzeige **Bearbeiter:** Annette Keusch
Bezeichner: MGrundbuchanzeige **Änderungsstand:** 02.06.03
Kurzbeschreibung: Stellt eine Funktion zur Anzeige von Grundbüchern bereit.
Export: GB_anzeigen
Import: GB_get - aus Modul Grundbuecher
StartAnzeigen - aus Modul Nutzerinterface
AnzeigenGB - aus Modul Nutzerinterface
T_Datumsflag - aus Modul Parameter
T_GB_Inhalt - aus Modul Parameter
T_Error - aus Modul Fehlerbehandlung

GB_anzeigen: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
GB_Datumsflag	inout	T_Datumsflag	Abschlussdatum des Grundbuchs
GB_Inhalt	out	T_GB_Inhalt	Inhalt des Grundbuchs
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Die Funktion ruft mit dem Parameter GB_Datumsflag die Funktion GB_get auf und liefert deren Rückgabewerte zurück.

1.2.12 Modul Archivdatenlöschung

Modul:	Archivdatenloeschung	Bearbeiter:	Annette Keusch
Bezeichner:	MArchivdatenloeschung	Änderungsstand:	05.06.03
Kurzbeschreibung:	Stellt alle Funktionen zum Löschen der verschiedenen persistenten Datenbestände bereit (Konten, Bilanzen, Grundbücher), welche zum Zeitpunkt der jeweiligen Funktionsaufrufe älter als 10 Jahre sind.		
Export:	AKto_loeschen ABz_loeschen AGB_loeschen		
Import:	Kto_find Bz_find GB_find Kto_delete Bz_delete GB_delete StartBearbeiten BearbeitenArchivLoeschen BearbeitenArchivKBest BearbeitenArchivGBBest BearbeitenArchivBzBest T_Error	- aus Modul Konten - aus Modul Bilanzen - aus Modul Grundbuecher - aus Modul Konten - aus Modul Bilanzen - aus Modul Grundbuecher - aus Modul Nutzerinterface - aus Modul Nutzerinterface - aus Modul Nutzerinterface - aus Modul Nutzerinterface - aus Modul Fehlerbehandlung	

AKto_loeschen: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Die Funktion generiert den Parameter Kto_DatumsflagMax := heute() – 10 Jahre und ruft mit diesem die Funktion Kto_find auf. Ist die zurückgegebene FehlerNr = 0, so werden alle gefundenen Konten sukzessive mithilfe der Funktion Kto_delete aus der Datenbank gelöscht. Verlieft der gesamte Vorgang erfolgreich, wird FehlerNr = 0 zurückgeliefert, ansonsten eine entsprechend andere Fehlernummer..

ABz_loeschen: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Die Funktion generiert den Parameter B_DatumMax := heute() – 10 Jahre und ruft mit diesem die Funktion Bz_find auf. Ist die zurückgegebene FehlerNr = 0, so werden alle gefundenen Bilanzen sukzessive mithilfe der Funktion Bz_delete aus der Datenbank gelöscht. Verlieft der gesamte Vorgang erfolgreich, wird FehlerNr = 0 zurückgeliefert, ansonsten eine entsprechend andere Fehlernummer.

AGB_loeschen:

Typ: Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Die Funktion generiert den Parameter GB_Datumsflag := heute() – 10 Jahre und ruft mit diesem die Funktion GB_find auf. Ist die zurückgegebene FehlerNr = 0, so werden alle gefundenen Grundbücher sukzessive mithilfe der Funktion GB_delete aus der Datenbank gelöscht. Verläuft der gesamte Vorgang erfolgreich, wird FehlerNr = 0 zurückgeliefert, ansonsten eine entsprechend andere Fehlernummer.

1.2.13 Modul SBilanzerstellung

Modul:	SBilanzerstellung	Bearbeiter:	Nadine Pollmer
Bezeichner:	MSBilanzerstellung	Änderungsstand:	04.06.03
Kurzbeschreibung:	Stellt die Funktion zur Schlussbilanzerstellung bereit.		
Export:	SBz_erstellen		
Import:	StartBilanz	- aus Modul Nutzerinterface	
	BilanzSB	- aus Modul Nutzerinterface	
	BilanzSBName	- aus Modul Nutzerinterface	
	BilanzSBEBName	- aus Modul Nutzerinterface	
	BilanzABKorrektur	- aus Modul Nutzerinterface	
	Kto_find	- aus Modul Konten	
	Kto_new	- aus Modul Konten	
	Kto_get	- aus Modul Konten	
	Kto_insertContentRow	- aus Modul Konten	
	Kto_changeFlag	- aus Modul Konten	
	GB_new	- aus Modul Grundbuecher	
	GB_changeFlag	- aus Modul Grundbuecher	
	GB_find	- aus Modul Grundbuecher	
	Bz_new	- aus Modul Bilanzen	
	Bz_get	- aus Modul Bilanzen	
	Bz_insertContentRow	- aus Modul Bilanzen	
	T_Datum	- aus Modul Parameter	
	T_Kto_Inhalt	- aus Modul Parameter	
	T_B_Art	- aus Modul Parameter	
	T_B_Inhalt	- aus Modul Parameter	
	T_Error	- aus Modul Fehlerbehandlung	

SBz_erstellen: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
B_Name	inout	char[80]	Name der Schlussbilanz
B_Datum	inout	T_Datum	Datum des Abschlusstages
Kto_Nr	inout	integer	Nummer eines Kontos
Kto_Name	inout	char[80]	Name eines Kontos
Kto_Inhalt	inout	T_Kto_Inhalt	Beinhaltet die Buchungseinträge und Summen der Soll und Habenseiten eines Kontos
AbschlussDatum	in	T_Datum	Datum des Abschlusstages
Betrag	in	real	Geldbetrag
B_Art	out	T_B_Art	„SB“ für Schlussbilanz
B_Inhalt	out	T_B_Inhalt	Inhalt der Bilanz
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: es existiert bereits mindestens eine Eröffnungsbilanz

Inhalt: Die Funktion erstellt eine neue leere Bilanz mit B_Name und B_Datum. Anschließend werden alle Konten mit dem Datumsflag „0“ in der Datenbank gesucht, der Saldo berechnet und auf das jeweilige Konto gebucht. Dann werden die Salden der Konten mit der Wertigkeit „1“ in das Schlussbilanzkonto gebucht, die alten Konten bekommen das AbschlussDatum als Datumsflag.

Der Schlussbilanzkontoinhalt wird in die neue Bilanz eingetragen und zurückgegeben. Nun werden neue Konten (inklusive SBK) mit den jeweiligen Anfangsbeständen (welche bei Wertigkeit „1“ geändert werden konnten) erstellt, dabei bleiben Kto_Name und Kto_Nr gleich. Dann wird das aktuelle Grundbuch abgeschlossen und ein neues leeres Grundbuch erstellt.

1.2.14 Modul EBilanzerstellung

Modul:	EBilanzerstellung	Bearbeiter:	Nadine Pollmer
Bezeichner:	MEBilanzerstellung	Änderungsstand:	04.06.03
Kurzbeschreibung:	Stellt die Funktion zur Eröffnungsbilanzerstellung bereit.		
Export:	EBz_erstellen		
Import:	StartBilanz	- aus Modul Nutzerinterface	
	BilanzEBmitInventar	- aus Modul Nutzerinterface	
	BilanzEB	- aus Modul Nutzerinterface	
	Kto_find	- aus Modul Konten	
	Kto_new	- aus Modul Konten	
	Kto_get	- aus Modul Konten	
	Kto_insertContentRow	- aus Modul Konten	
	Bz_new	- aus Modul Bilanzen	
	Bz_get	- aus Modul Bilanzen	
	Bz_insertContentRow	- aus Modul Bilanzen	
	Bz_find	- aus Modul Bilanzen	
	T_Datum	- aus Modul Parameter	
	T_Kto_Inhalt	- aus Modul Parameter	
	T_B_Art	- aus Modul Parameter	
	T_B_Inhalt	- aus Modul Parameter	
	T_Error	- aus Modul Fehlerbehandlung	

EBz_erstellen: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
B_Name	inout	char[80]	Name der Eröffnungsbilanz
B_Datum	inout	T_Datum	Datum des Abschlusstages / Eröffnungstages
Kto_Nr	inout	Integer	Nummer eines Kontos
Kto_Name	inout	char[80]	Name eines Kontos
Kto_Inhalt	inout	T_Kto_Inhalt	Beinhaltet die Buchungseinträge und Summen der Soll und Habenseiten eines Kontos
Betrag	in	Real	Geldbetrag
B_Art	out	T_B_Art	„EB“ für Eröffnungsbilanz
B_Inhalt	out	T_B_Inhalt	Inhalt der Bilanz
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Wenn es noch keine andere Bilanz in der Datenbank gibt, werden alle Konten mit der Wertigkeit „1“ herausgesucht und die Anfangsbestände eingetragen, diese werden ebenfalls in das SBK gebucht. Anschließend (egal ob bei erster Eröffnungsbilanz oder im Kontenabschlussprozess) erstellt die Funktion eine neue leere Bilanz mit B_Name und B_Datum. Ebenfalls wird ein neues Konto mit der Kto_Art „EK“ erstellt und die jeweiligen Beträge aus dem SBK entsprechend gebucht. Der B_Inhalt wird dann mit den Daten aus dem EK gefüllt und zurückgegeben. Die Bilanzerstellung ist beendet.

1.2.15 Modul Bilanzanzeige

Modul:	Bilanzanzeige	Bearbeiter:	Annette Keusch
Bezeichner:	MBilanzanzeige	Änderungsstand:	02.06.03
Kurzbeschreibung:	Stellt eine Funktion zur Anzeige von Bilanzen bereit.		
Export:	Bz_anzeigen		
Import:	Bz_get - aus Modul Bilanzen StartAnzeigen - aus Modul Nutzerinterface AnzeigenBilanz - aus Modul Nutzerinterface T_Datum - aus Modul Parameter T_B_Art - aus Modul Parameter T_B_Inhalt - aus Modul Parameter T_Error - aus Modul Fehlerbehandlung		

Bz_anzeigen: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
B_Datum	inout	T_Datum	Erstellungsdatum der Bilanz
B_Art	inout	T_B_Art	Art der Bilanz, entweder „SB“ oder „EB“
B_Name	out	Char[80]	Name der Bilanz
B_Inhalt	out	T_B_Inhalt	Beinhaltet die Nummern, Namen und Beträge der (abgeschlossenen) Konten auf der Aktiva- und Passivseite
FehlerNr	out	T_Error	Fehlercode

Voraussetzung: keine

Inhalt: Die Funktion ruft mit den Parametern B_Datum und B_Art die Funktion Bz_get auf und liefert deren Rückgabewerte zurück.

1.2.16 Modul Parameter

Modul:	Parameter	Bearbeiter:	Nadine Pollmer
Bezeichner:	MParameter	Änderungsstand:	05.06.03
Kurzbeschreibung:	Stellt alle Datentypen und Variablen für das Buchführungsverwaltungssystem zur Verfügung.		
Export:	T_Datumsflag T_Datum T_Kto_Art T_Kto_Inhalt T_Kto_Seite T_B_Art T_B_Inhalt T_B_Seite T_B_Posten T_GB_Inhalt T_Grundbucheintrag T_Buchungseintrag T_GF_Vorgangsdaten T_Kontenliste help		
Import:	---		

T_Datumsflag: **Typ:** Datentyp

Type Abschlussdatum = record
Tag: [1..31];
Monat: [1..12];
Jahr: integer;
end;

T_Datumsflag = [-1, 0, Abschlussdatum];

T_Datum: **Typ:** Datentyp

Type T_Datum = record
Tag: [1..31];
Monat: [1..12];
Jahr: integer;
end;

T_Kto_Art: **Typ:** Datentyp

Type T_Kto_Art = [„aktiv“, „passiv“, „SBK“, „EK“];

T_Kto_Inhalt:**Typ:** Datentyp

Type T_Datum = record
 Tag: [1..31];
 Monat: [1..12];
 Jahr: integer;
end;

T_Zeit = record
 Stunde: [0..23];
 Minute: [0..59];
 Sekunde: [0..59];
end;

Beleg_ID = record
 Datum: T_Datum;
 Zeit: T_Zeit;
end;

T_Buchungseintrag = record
 Id: Beleg_ID;
 GegenKtoNr: integer;
 GegenKtoName: char[80];
 GegenBetrag: real;
end;

Kto_Seite = record
 Buchungseintrag: set of T_Buchungseintrag;
 Summe: real;
end;

T_Kto_Inhalt = record
 Soll: Kto_Seite;
 Haben: Kto_Seite;
end;

T_Kto_Seite:**Typ:** Datentyp

Type T_Kto_Seite = [„Soll“, „Haben“];

T_B_Art:**Typ:** Datentyp

Type T_B_Art = [„SB“, „EB“];

T_B_Inhalt:**Typ:** Datentyp

Type T_B_Posten = record
 BP_Nr: integer;
 BP_Name: char[80];
 BP_Betrag: real;
end;

B_Seite = record
 BSeiteninhalt: set of T_B_Posten;
 Summe: real;
end;

T_B_Inhalt = record
 Aktiva: B_Seite;
 Passiva: B_Seite;
end;

T_B_Seite:**Typ:** Datentyp

Type T_B_Seite = [„aktiva“, „passiva“];

T_B_Posten:**Typ:** Datentyp

Type T_B_Posten = record
 BP_Nr: integer;
 BP_Name: char[80];
 BP_Betrag: real;
end;

T_GB_Inhalt:**Typ:** Datentyp

```
Type  T_BSZ_blanko = record
      Kto_Nr:      integer;
      Kto_Name:    char[80];
      Betr_Kto_Seite: [„Soll“, „Haben“]
    end;

      T_Kontenliste = record
      BSZ_blanko:  T_BSZ_blanko;
      Buchungsbetrag: real;
    end;

      T_Zeit = record
      Stunde:      [0..23];
      Minute:      [0..59];
      Sekunde:     [0..59];
    end;

      T_Datum = record
      Tag:          [1..31];
      Monat:       [1..12];
      Jahr:         integer;
    end;

      Beleg_ID = record
      Datum:       T_Datum;
      Zeit:        T_Zeit;
    end;

      T_Grundbucheintrag = record
      Buchungsdatum: T_Datum;
      ID:            Beleg_ID;
      Beleg_Kommentar: char[80];
      Kontenliste:  set of T_Kontenliste;
    end;

      T_GB_Inhalt = set of T_Grundbucheintrag;
```

T_Grundbucheintrag:**Typ:** Datentyp

```
Type  T_BSZ_blanko = record
      Kto_Nr:      integer;
      Kto_Name:    char[80];
      Betr_Kto_Seite: [„Soll“, „Haben“]
    end;

      T_Kontenliste = record
      BSZ_blanko:  T_BSZ_blanko;
      Buchungsbetrag: real;
    end;

      T_Zeit = record
      Stunde:      [0..23];
      Minute:      [0..59];
      Sekunde:     [0..59];
    end;

      T_Datum = record
      Tag:          [1..31];
      Monat:       [1..12];
      Jahr:         integer;
    end;

      Beleg_ID = record
      Datum:        T_Datum;
      Zeit:         T_Zeit;
    end;

      T_Grundbucheintrag = record
      Buchungsdatum: T_Datum;
      ID:             Beleg_ID;
      Beleg_Kommentar: char[80];
      Kontenliste:   set of T_Kontenliste;
    end;
```

T_Buchungseintrag:**Typ:** Datentyp

Type T_Datum = record
 Tag: [1..31];
 Monat: [1..12];
 Jahr: integer;
end;

T_Zeit = record
 Stunde: [0..23];
 Minute: [0..59];
 Sekunde: [0..59];
end;

Beleg_ID = record
 Datum: T_Datum;
 Zeit: T_Zeit;
end;

T_Buchungseintrag = record
 Id: Beleg_ID;
 GegenKtoNr: integer;
 GegenKtoName: char[80];
 GegenBetrag: real;
end;

T_GF_Vorgangsdaten:**Typ:** Datentyp

Type T_BSZ_blanko = record
 Kto_Nr: integer;
 Kto_Name: char[80];
 Betr_Kto_Seite: [„Soll“, „Haben“]
end;

T_GF_Vorgangsdaten = set of T_BSZ_blanko;

T_Kontenliste:**Typ:** Datentyp

Type T_Kontenliste = array[2..n] of record
 BSZ_blanko: T_BSZ_blanko;
 Buchungsbetrag: real;
end;

help:**Typ:** Variable

help: boolean;

1.2.17 Modul Fehlerbehandlung

Modul:	Fehlerbehandlung	Bearbeiter:	Nadine Pollmer
Bezeichner:	MFehlerbehandlung	Änderungsstand:	05.06.03
Kurzbeschreibung:	Stellt die Funktion zur Dekodierung der FehlerNr bereit, enthält die Fehlercodetabelle und den Datentyp T_Error.		
Export:	Error_decode T_Error		
Import:	---		

T_Error: **Typ:** Datentyp

Type T_Error = integer;

Error_decode: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
FehlerNr	In	T_Error	Fehlercode
ErrorMsg	Out	char[100]	informierender Text zur Beschreibung des aufgetretenen Fehlers, ausgelesen aus der Fehlercodetabelle

Voraussetzung: keine

Inhalt: Die Funktion liest mit der gegebenen FehlerNr die Fehlernachricht ErrorMsg aus der Fehlercodetabelle (siehe unten) und gibt sie in einem String zurück.

FehlerNr	ErrorMsg
0000	Aktion erfolgreich durchgeführt
0001	Fehler bei Eingabe
0002	Fehler in Datenbank
...	...

1.2.18 Modul Hilfe

Modul: **Hilfe** **Bearbeiter:** Nadine Pollmer
Bezeichner: MHilfe **Änderungsstand:** 04.06.03
Kurzbeschreibung: Stellt die Hilfe für das Buchführungssystem bereit.
Export: Hilfeseite
Import: **help** - aus Modul Parameter

Hilfeseite: **Typ:** Funktion

Parameter:

Bezeichner	Typ	Format	Bedeutung
Help	in	T_help	Variable, die die Hilfeseite aufruft

Voraussetzung: keine

Inhalt: Auflistung der einzelnen Themen der Hilfeseite und ihre entsprechenden Beschreibungen.

1.2.19 Modul Buchführungsverwaltung

Modul:	Buchfuerungsverwaltung	Bearbeiter:	Nadine Pollmer
Bezeichner:	MBuchfuerungsverwaltung	Änderungsstand:	03.06.03
Kurzbeschreibung:	Programmlogik		
Export:	Start	- Startseite	
	StartBuchung	- Auswahl Buchung direkt oder über Geschäftsfall	
	BuchungDirekt	- direkte Buchung	
	BuchungGF	- Buchung über Geschäftsfall	
	BuchungBest	- Bestätigung einer Buchung	
	StartAnzeigen	- Auswahl dessen, was angezeigt werden soll	
	AnzeigenKontoWaehlen	- Auswahl des anzuzeigenden Kontos aus Kontenliste	
	AnzeigenKonto	- zeigt das gewählte Konto an	
	AnzeigenGB	- zeigt das gewählte Grundbuch an	
	AnzeigenGF	- zeigt den gewählten Geschäftsfall an	
	AnzeigenBilanz	- zeigt die gewählte Bilanz an	
	StartBearbeiten	- Auswahl dessen, was bearbeitet werden soll	
	BearbeitenKonto	- neues Konto anlegen	
	BearbeitenKtoBest	- Bestätigung der Aktion „Konto anlegen“	
	BearbeitenGFAnlegen	- neuen Geschäftsfall anlegen	
	BearbeitenGFBest	- Bestätigung der Aktion „Geschäftsfall anlegen“	
	BearbeitenGFLoeschen	- löscht einen Geschäftsfall	
	BearbeitenGFLoeschBest	- Bestätigung der Aktion „Geschäftsfall löschen“	
	BearbeitenArchivLoeschen	- Abfrage ob wirklich gelöscht werden soll	
	BearbeitenArchivLBest	- Bestätigung der Aktion „Archivdaten löschen“	
	StartBilanz	- Auswahl, welche Bilanz erstellt werden soll	
	BilanzEBmitInventar	- Anfangsbeträge eintragen und Eröffnungsbilanzname eintragen	
	BilanzSB	- Anzeige der Schlussbilanz	
	BilanzABKorrektur	- Anfangsbeträge korrigieren und Eröffnungsbilanzname eintragen	
	BilanzSBEBName	- Eingabe des Eröffnungsbilanznamens	
	BilanzSBName	- Eingabe des Schlussbilanznamens	
	BilanzEB	- Anzeige der Eröffnungsbilanz	
	Ende	- Auswahl, ob Ende des Programms	
Import:	Start, StartBuchung, BuchungDirekt, BuchungGF, BuchungBest, StartAnzeigen, AnzeigenKontoWaehlen, AnzeigenKonto, AnzeigenGB, AnzeigenGF, AnzeigenBilanz, StartBearbeiten, BearbeitenKonto, BearbeitenKtoBest, BearbeitenGFAnlegen, BearbeitenGFBest, BearbeitenGFLoeschen, BearbeitenGFLoeschBest, BearbeitenArchivLoeschen, BearbeitenArchivLBest, StartBilanz, BilanzEBmitInventar, BilanzSB, BilanzABKorrektur, BilanzSBName, BilanzSBEBName, BilanzEB, Ende		
	GF_anzeigen	- aus Modul Buchfuehrungsverwaltung	
	GF_loeschen, GF_anlegen	- aus Modul Geschaeftsfallanzeige	
	Kto_anzeigen	- aus Modul Kontoanzeige	
	Kto_anlegen	- aus Modul Kontobearbeitung	
	GB_anzeigen	- aus Modul Grundbuchanzeige	
	Bz_anzeigen	- aus Modul Bilanzanzeige	
	EBz_erstellen	- aus Modul Ebilanzerstellung	
	SBz_erstellen	- aus Modul Sbilanzerstellung	
	AKto_loeschen	- aus Modul Archivdatenlöschung	
	ABz_loeschen	- aus Modul Archivdatenlöschung	
	AGB_loeschen	- aus Modul Archivdatenlöschung	

buchen
Error_decode

- aus Modul Buchung
- aus Modul Fehlerbehandlung

Generische Beschreibung zu allen Exports:

Typ:	PHP4-Anteil der PHP4-Seite
Parameter:	Seiten, die durch Formulare aufgerufen werden, erhalten in Variablen (Benennung analog zum Datenkatalog) die Eingaben des Nutzers
Vorraussetzung:	keine
Inhalt:	Funktionalität für die analog benannten Webseiten, bei denen der HTML-Anteil genutzt wird, des Moduls „Nutzerinterface“. Das Modul ermittelt anhand der aktuellen Webseite, welche Module aufgerufen/genutzt werden und verteilt die Aufgaben nach „unten“ und gibt das Ergebnis an das Nutzerinterface zurück. Für einzelne Funktionalitäten der Exporte siehe auch Punkt 1.3 der Dokumentation: „Definition der Nutzerschnittstelle.“

1.2.20 Modul Nutzerinterface

Modul:	Nutzerinterface	Bearbeiter:	Nadine Pollmer
Bezeichner:	MNutzerinterface	Änderungsstand:	03.06.03
Kurzbeschreibung:	Oberfläche in Form von PHP4-Webseiten (HTML-Anteil)		
Export:	Start	- Startseite	
	StartBuchung	- Auswahl Buchung direkt oder über Geschäftsfall	
	BuchungDirekt	- direkte Buchung	
	BuchungGF	- Buchung über Geschäftsfall	
	BuchungBest	- Bestätigung einer Buchung	
	StartAnzeigen	- Auswahl dessen, was angezeigt werden soll	
	AnzeigenKontoWählen	- Auswahl des anzuzeigenden Kontos aus Kontenliste	
	AnzeigenKonto	- zeigt das gewählte Konto an	
	AnzeigenGB	- zeigt das gewählte Grundbuch an	
	AnzeigenGF	- zeigt den gewählten Geschäftsfall an	
	AnzeigenBilanz	- zeigt die gewählte Bilanz an	
	StartBearbeiten	- Auswahl dessen, was bearbeitet werden soll	
	BearbeitenKonto	- neues Konto anlegen	
	BearbeitenKtoBest	- Bestätigung der Aktion „Konto anlegen“	
	BearbeitenGFAnlegen	- neuen Geschäftsfall anlegen	
	BearbeitenGFBest	- Bestätigung der Aktion „Geschäftsfall anlegen“	
	BearbeitenGFLoeschen	- löscht einen Geschäftsfall	
	BearbeitenGFLoeschBest	- Bestätigung der Aktion „Geschäftsfall löschen“	
	BearbeitenArchivLoeschen	- Abfrage ob wirklich gelöscht werden soll	
	BearbeitenArchivLBest	- Bestätigung der Aktion „Archivdaten löschen“	
	StartBilanz	- Auswahl, welche Bilanz erstellt werden soll	
	BilanzEBmitInventar	- Anfangsbeträge eintragen und Eröffnungsbilanzname eintragen	
	BilanzSB	- Anzeige der Schlussbilanz	
	BilanzABKorrektur	- Anfangsbeträge korrigieren und Eröffnungsbilanzname eintragen	
	BilanzSBEBName	- Eingabe des Eröffnungsbilanznamens	
	BilanzSBName	- Eingabe des Schlussbilanznamens	
	BilanzEB	- Anzeige der Eröffnungsbilanz	
	Ende	- Auswahl, ob Ende des Programms	
Import:	Hilfeseite	- aus Modul Hilfe	
	Start, StartBuchung, BuchungDirekt, BuchungGF, BuchungBest, StartAnzeigen, AnzeigenKontoWählen, AnzeigenKonto, AnzeigenGB, AnzeigenGF, AnzeigenBilanz, StartBearbeiten, BearbeitenKonto, BearbeitenKtoBest, BearbeitenGFAnlegen, BearbeitenGFBest, BearbeitenGFLoeschen, BearbeitenGFLoeschBest, BearbeitenArchivLoeschen, BearbeitenArchivLBest, StartBilanz, BilanzEBmitInventar, BilanzSB, BilanzABKorrektur, BilanzSBName, BilanzSBEBName, BilanzEB, Ende	- aus Modul Buchfuehrungsverwaltung	

Generische Beschreibung zu allen Exports:

Typ:	HTML-Anteil der PHP4-Webseite
Parameter:	Die Ein- und Ausgaben der dynamischen Webseiten werden in Variablen übergeben und im Modul „Buchfuehrungsverwaltung“ gefüllt.
Vorraussetzung:	Die Seite wird nicht direkt vom Anwender, sondern indirekt durch den HTML-Teil in der PHP4-Seite aufgerufen.
Inhalt:	Siehe Punkt 1.3 der Dokumentation: „Definition der Nutzerschnittstelle“
