

Operation:	buchdaten_eingeben		
Description:	Bei Anschaffung eines Buches muß der Bibliothekar die entsprechenden Daten in den Rechner eingeben.		
Reads:	supplied	titel	: Buch.titel
	supplied	author	: Buch.author
	supplied	verlag	: Buch.verlag
	supplied	isbn	: Buch.isbn
	supplied	auflage	: Buch.auflage
	supplied	jahr	: Buch.jahr
Changes:	new Buch Buch.*		
Sends:			
Assumes:			
Results:	Buch.titel	=	titel
	Buch.author	=	author
	Buch.verlag	=	verlag
	Buch.isbn	=	isbn
	Buch.auflage	=	auflage
	Buch.jahr	=	jahr
	Buch.zustand	=	"vorhanden"
	final Buch. id \notin m mit m = (initial Buch.id)		

Operation:	kundendaten_eingeben		
Description:	Bei Neuanmeldung eines Kunden muß der Bibliothekar die entsprechenden Daten in den Rechner eingeben.		
Reads:	supplied	name	: Kunde.Daten.name
	supplied	vorname	: Kunde.Daten.vorname
	supplied	adresse	: Kunde.Daten.adresse
	supplied	gebdatum	: Kunde.Daten.gebdatum
Changes:	new Kunde Kunde.*		
Sends:			
Assumes:			
Results:	Kunde.Daten.name	=	name
	Kunde.Daten.vorname	=	vorname
	Kunde.Daten.adresse	=	adresse
	Kunde.Daten.gebdatum	=	gebdatum
	Kunde.Konto.mahngebühr	=	0
	final Kunde.Daten.id \notin m mit m = (initial Kunde.Daten.id)		

Operation: suchdaten_eingeben_buch																													
Description: Bibliothekar gibt die Suchdaten in den Rechner ein.																													
Reads:	<table> <tr><td>supplied</td><td>titel</td><td>:</td><td>Buch.titel</td></tr> <tr><td>supplied</td><td>author</td><td>:</td><td>Buch.author</td></tr> <tr><td>supplied</td><td>verlag</td><td>:</td><td>Buch.verlag</td></tr> <tr><td>supplied</td><td>isbn</td><td>:</td><td>Buch.isbn</td></tr> <tr><td>supplied</td><td>auflage</td><td>:</td><td>Buch.auflage</td></tr> <tr><td>supplied</td><td>jahr</td><td>:</td><td>Buch.jahr</td></tr> <tr><td>supplied</td><td>id</td><td>:</td><td>Buch.id</td></tr> </table>	supplied	titel	:	Buch.titel	supplied	author	:	Buch.author	supplied	verlag	:	Buch.verlag	supplied	isbn	:	Buch.isbn	supplied	auflage	:	Buch.auflage	supplied	jahr	:	Buch.jahr	supplied	id	:	Buch.id
supplied	titel	:	Buch.titel																										
supplied	author	:	Buch.author																										
supplied	verlag	:	Buch.verlag																										
supplied	isbn	:	Buch.isbn																										
supplied	auflage	:	Buch.auflage																										
supplied	jahr	:	Buch.jahr																										
supplied	id	:	Buch.id																										
Changes:																													
Sends:	Terminal {buchdaten, nicht_gefunden}																												
Assumes:	titel ≠ "" ∨ author ≠ "" ∨ verlag ≠ "" ∨ isbn ≠ "" ∨ auflage ≠ "" ∨ jahr ≠ "" ∨ id ≠ ""																												
Results:	if exists (titel ∨ author ∨ verlag ∨ isbn ∨ auflage ∨ jahr ∨ id) then is_sent (buchdaten) else is_sent (nicht_gefunden)																												

Operation: suchdaten_eingeben_kunde																					
Description: Bibliothekar gibt die Suchdaten in den Rechner ein.																					
Reads:	<table> <tr><td>supplied</td><td>name</td><td>:</td><td>Kunde.Daten.name</td></tr> <tr><td>supplied</td><td>vorname</td><td>:</td><td>Kunde.Daten.vorname</td></tr> <tr><td>supplied</td><td>adresse</td><td>:</td><td>Kunde.Daten.adresse</td></tr> <tr><td>supplied</td><td>gebdatum</td><td>:</td><td>Kunde.Daten.gebdatum</td></tr> <tr><td>supplied</td><td>id</td><td>:</td><td>Kunde.Daten.id</td></tr> </table>	supplied	name	:	Kunde.Daten.name	supplied	vorname	:	Kunde.Daten.vorname	supplied	adresse	:	Kunde.Daten.adresse	supplied	gebdatum	:	Kunde.Daten.gebdatum	supplied	id	:	Kunde.Daten.id
supplied	name	:	Kunde.Daten.name																		
supplied	vorname	:	Kunde.Daten.vorname																		
supplied	adresse	:	Kunde.Daten.adresse																		
supplied	gebdatum	:	Kunde.Daten.gebdatum																		
supplied	id	:	Kunde.Daten.id																		
Changes:																					
Sends:	Terminal {kundendaten, nicht_gefunden}																				
Assumes:	name ≠ "" ∨ vorname ≠ "" ∨ adresse ≠ "" ∨ gebdatum ≠ "" ∨ id ≠ ""																				
Results:	if exists (name ∨ vorname ∨ adresse ∨ gebdatum ∨ id) then is_sent (kundendaten) else is_sent (nicht_gefunden)																				

Operation:	löscht_befehl_buch		
Description:	Löschen der ausgewählten Bücher.		
Reads:	supplied	del	: bool
	supplied	auswahl	: Buch
Changes:	Buch.*		
Sends:	Terminal {gelöscht, nicht_gelöscht}		
Assumes:	Buch.zustand = "vorhanden"		
Results:	if del then { $\neg \exists$ Buch.* mit Buch.* = auswahl.*; is_sent(gelöscht) } else is_sent (nicht_gelöscht)		

Operation:	löscht_befehl_kunde		
Description:	Löschen eines Kundenprofils.		
Reads:	supplied	del	: bool
	supplied	auswahl	: Kunde
Changes:	Kunde.*		
Sends:	Terminal {gelöscht, nicht_gelöscht}		
Assumes:			
Results:	if del then { $\neg \exists$ Kunde.* mit Kunde.* = auswahl.*; is_sent(gelöscht) } else is_sent (nicht_gelöscht)		

Operation:	ändern_kundendaten
Description:	Löschen der ausgewählten Bücher.
Reads:	supplied daten : Kunde.Daten
Changes:	Kunde.Daten.name Kunde.Daten.vorname Kunde.Daten.adresse Kunde.Daten.gebdatum
Sends:	
Assumes:	$\forall \text{ daten.}^* \neq ""$
Results:	Kunde.Daten.* = daten.*

Operation:	buchzustand_verändern
Description:	Ändern der Bucheigenschaft Zustand auf "vorhanden" oder "ausgeliehen".
Reads:	supplied vorhanden : bool
Changes:	Buch.zustand
Sends:	
Assumes:	
Results:	if vorhanden then Buch.zustand = "vorhanden" else Buch.zustand = "ausgeliehen"

Operation:	mahngebühr_löschen
Description:	Zurücksetzen der Mahngebühr eines Kunden auf Null.
Reads:	
Changes:	Kunde.Konto.mahngebühr
Sends:	
Assumes:	
Results:	Kunde.Konto.mahngebühr = 0

Operation:	leihschein_löschen
Description:	Ein Leihschein wird gelöscht, wenn jemand das entsprechende Buch zurückbringt.
Reads:	supplied del : bool supplied buchid : Buch.id supplied kundenkonto : Kunde.Konto
Changes:	Kunde.Konto.anzahl Kunde.Konto.leihscheine
Sends:	Terminal {gelöscht, nicht_gelöscht} Drucker {kundenkonto}
Assumes:	Kunde.Konto.anzahl > 0
Results:	if del then { $\neg \exists$ Kunde.Konto.leihscheine enthält Leihschein.buchid; dec (Kunde.Konto.anzahl); is_sent(gelöscht); is_sent(kundendaten);} else is_sent(nicht_gelöscht)

Operation:	leihschein_anlegen
Description:	Ein Leihschein wird angelegt, wenn ein Kunde ein Buch ausleiht.
Reads:	supplied buchdaten : Buch/Medium supplied kundendaten : Kunde supplied datum : Systemdatum
Changes:	kundendaten.konto.* new Leihschein
Sends:	Drucker {Leihschein.*} Monitor {kundendaten.*}
Assumes:	kundendaten.Konto.anzahl < 10 kundendaten.Konto.mahngebühr = 0 \exists Buch mit (buch = buchdaten \wedge zustand = "vorhanden")
Results:	Leihschein.buch.daten.titel = buchdaten.titel; Leihschein.rückgabedatum = <i>nächsterArbeitstag</i> (datum+30); Leihschein.buchid = buchdaten.id; inc (kundendaten.Konto.Leihschein.anzahl); kundendaten.Konto.rückgabedatum = leihschein.rückgabedatum; is_sent Drucker(Leihschein.*); is_sent Monitor(kundendaten.*);

Operation:	mahngebühr_berechnen		
Description:	Bei Überschreitung des Rückgabedatums wird die Mahngebühr neu berechnet.		
Reads:	supplied	datum	: Systemdatum
Changes:	Kunde.*		
Sends:			
Assumes:			
Results:	$\text{Kunde.Konto.mahngebühr} = \text{Preis} * \text{Kunde.Konto.anzahl} * \text{arbeitstage}(\text{datum}, \text{Kunde.Konto.rückgabedatum});$		