Formula 🗸										Global Conditions			Variables	
precondition statement			postcondition					Global Collutions			L	OCAL int newBalance		
precondition		Statement			{balanceChangedResult(\result,					_		RETURN boolean ret		
{true}		statement			balance, \old(balance), x)}			OVERDRAFT_LIMIT = 0			PARAM int x			
													PUBLIC int balance	
→												PUBI	LIC final int OVERDRAFT	
	Composition													
	precondition					postcondition								
•					{balanceChanged				Result(\result, balance,					
	{true}							\old(b	palance), x)}					
	statement 1			inte		rmediate condition			stateme					
	statement1			{newE	nce = balance	= balance + x}		statement2						
Statement1 $\sqrt{}$														
	-4-4) (0.				>	
			ement postco		Н	Composition								
{true}	newBalance = balance + x;		{newBalance = balance + x}		lt		precondition				postcondition			
SelectionStatement1					'	{newBalance = balance + x} {balance					ceCha	hangedResult(\result, balance,		
					١l	(iicwbaiance – baian			\			old(b	alance), x)}	
SelectionStatement IFFI			FI	*		state		nt 1	intermediate condition		on	statement 2		
guards				statement1			(balanceChang		ChangedResult	ngedResult(ret,		statement2		
newBalance <			newBalance >			State	Statement		balance, \old(balance), x)}		x)}	Statementz		
this.OVERDRAFT_LIMIT precon		ondition	this.OVERDRAFT_LIMIT ndition			ReturnStatement1								
(newBalance = balance + x) & (newBalance = balance + x)														
(newBalance < staten			(newBalance >=			preco	condition		ReturnStatement			postcondition		
					{balanceChangedResult(ret,						{balanceChangedResult(\resul			
statement			statement			balance, \old			ret;			t, balance, \old(balance), x)}		
	posto	ondition			1			- · ·						
{balanceChang	gedResult(r	et, balance, \	old(balan	ce), x)}	J		7	Statemer	nt3					
Statement2						precon			lition statement		ent		postcondition	
precondition		stateme	tement pos		tcor	tcondition		{modifiable(\nothing):		palance = newBalance; ret = true;			
{modifiable(\nothing);			ret = false;			angedResult(r		(newBalance		balance = ne			{balanceChangedResult(
newBalance = balance +		ret = fa						+ x) & (newB		ret = t			ret, balance, \old(balance), x)}	
x) & (newBalance <		12.122/		\old(balan			this.OVERDRA		AFT_LIMIT)}				(2.3(23.31100)/ //)	

\old(balance), x)}

this.OVERDRAFT_LIMIT)}

Variables