

## Method Detail

### applyInterest

Function signatures have been removed; you need to determine their proper format from the UML class diagram

This method applies the current interest rate to the current balance in this account

---

### getInterestRate

Accessor; retrieves the interest rate applied to all BankAccounts

**Returns:**

the interestRate

---

### setInterestRate

Mutator; sets the interest rate applied to all BankAccounts

**Parameters:**

interestRate - the interestRate to set

**Returns:**

true if successful; false if specified interestRate < 0 or > 100

---

### getType

Accessor; retrieves the account type for this account

**Returns:**

the account type; BankAccount.SAVINGS or BankAccount.CHECKING

---

### getBalance

Accessor; retrieves the current balance of this account

**Returns:**

the balance

---

### getHistory

Accessor; retrieves the history of all transactions since creation of this account

**Returns:**

the transaction history

---

**withdraw**  


This method is used to withdraw funds from the account

**Parameters:**

amount - the sum of money to be withdrawn

**Returns:**

true on success; false **if** amount exceeds the current balance, or the amount is negative. The account balance is not modified if unsuccessful. The transaction history is updated in all cases, and indicates either a successful withdrawal, or indicates the specific error which caused the failure.

---

**deposit**  


This method is used to deposit funds to the account

**Parameters:**

amount - the sum of money to be deposited

**Returns:**

true on success; false if the amount to be deposited is  $< 0$ . The account balance is not modified if unsuccessful. The transaction history is updated in all cases, and indicates either a successful deposit, or indicates the specific error which caused the failure.

---

**Package** **Class** [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

There is no description above for the 3-argument BankAccount constructor, but you can determine what it does from the actual comments you need to place in your implementation:

```
/**
 * Constructor
 * @param initialBalance the beginning balance for this account
 * @param type the account type; BankAccount.SAVINGS or BankAccount.CHECKING
 * @param acctNo the account number, for example "C123-00" or "S456-99"
 * If the value for initialBalance is  $< 0$ , the beginning balance is set to 0.
 * If the value for type is invalid, the account type is set to BankAccount.CHECKING
 * This constructor also initializes the transaction history maintained for each account.
 */
```