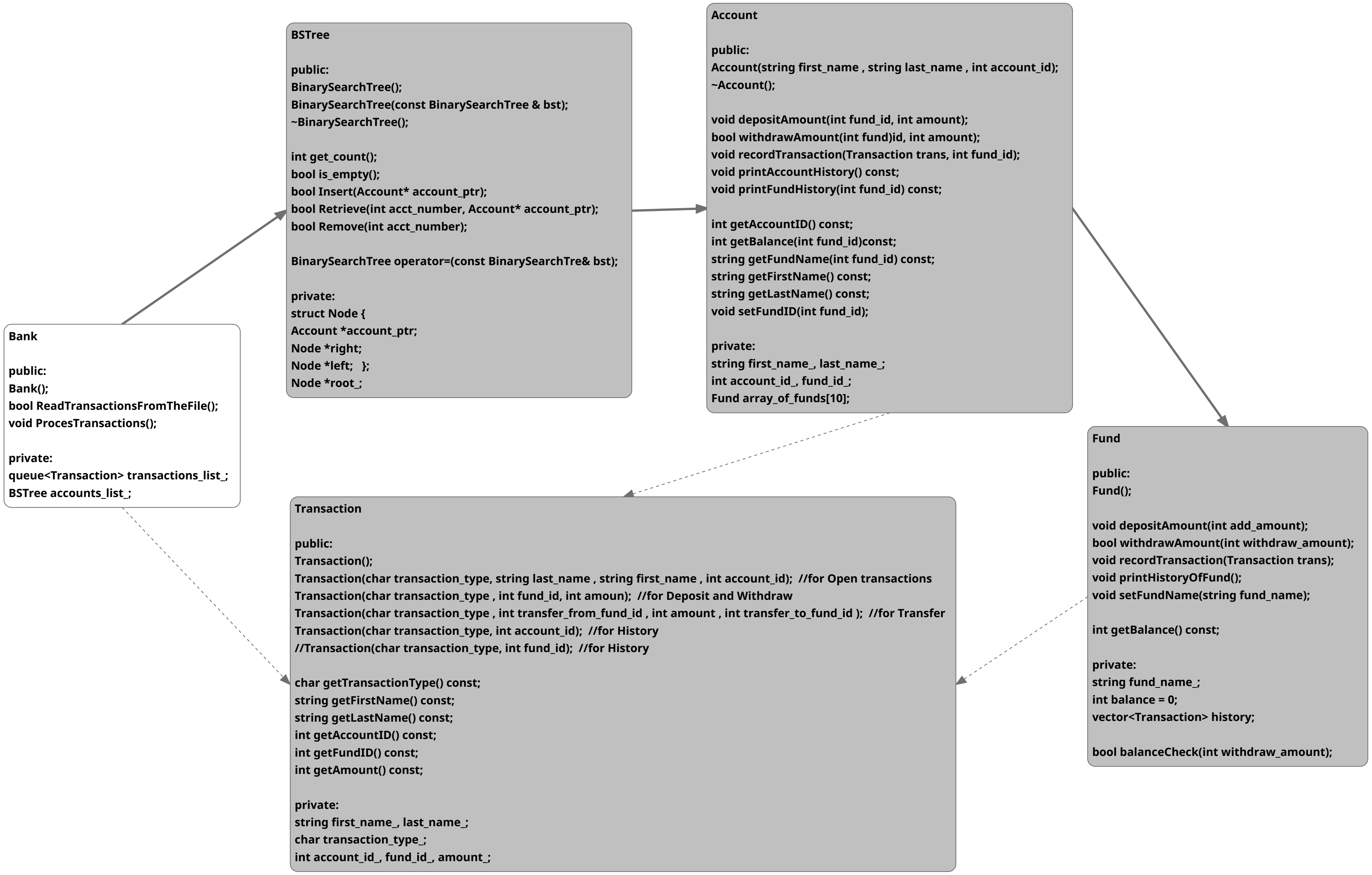

```
BSTree

public:
BinarySearchTree();
BinarySearchTree(const BinarySearchTree & bst);
~BinarySearchTree();

int get_count();
bool is_empty();
bool Insert(Account* account_ptr);
bool Retrieve(int acct_number, Account* account_ptr);
bool Remove(int acct_number);

BinarySearchTree operator=(const BinarySearchTre& bst);

private:
struct Node {
Account *account_ptr;
Node *right;
Node *left;   };
Node *root_;
```

```
Account

public:
Account(string first_name , string last_name , int account_id);
~Account();

void depositAmount(int fund_id, int amount);
bool withdrawAmount(int fund)id, int amount);
void recordTransaction(Transaction trans, int fund_id);
void printAccountHistory() const;
void printFundHistory(int fund_id) const;

int getAccountID() const;
int getBalance(int fund_id)const;
string getFundName(int fund_id) const;
string getFirstName() const;
string getLastName() const;
void setFundID(int fund_id);

private:
string first_name_, last_name_;
int account_id_, fund_id_;
Fund array_of_funds[10];
```

```
Bank

public:
Bank();
bool ReadTransactionsFromTheFile();
void ProcesTransactions();

private:
queue<Transaction> transactions_list_;
BSTree accounts_list_;
```

```
Fund

public:
Fund();

void depositAmount(int add_amount);
bool withdrawAmount(int withdraw_amount);
void recordTransaction(Transaction trans);
void printHistoryOfFund();
void setFundName(string fund_name);

int getBalance() const;

private:
string fund_name_;
int balance = 0;
vector<Transaction> history;

bool balanceCheck(int withdraw_amount);
```

```
Transaction

public:
Transaction();
Transaction(char transaction_type, string last_name , string first_name , int account_id);  //for Open transactions
Transaction(char transaction_type , int fund_id, int amoun);  //for Deposit and Withdraw
Transaction(char transaction_type , int transfer_from_fund_id , int amount , int transfer_to_fund_id );  //for Transfer
Transaction(char transaction_type, int account_id);  //for History
//Transaction(char transaction_type, int fund_id);  //for History

char getTransactionType() const;
string getFirstName() const;
string getLastName() const;
int getAccountID() const;
int getFundID() const;
int getAmount() const;

private:
string first_name_, last_name_;
char transaction_type_;
int account_id_, fund_id_, amount_;
```