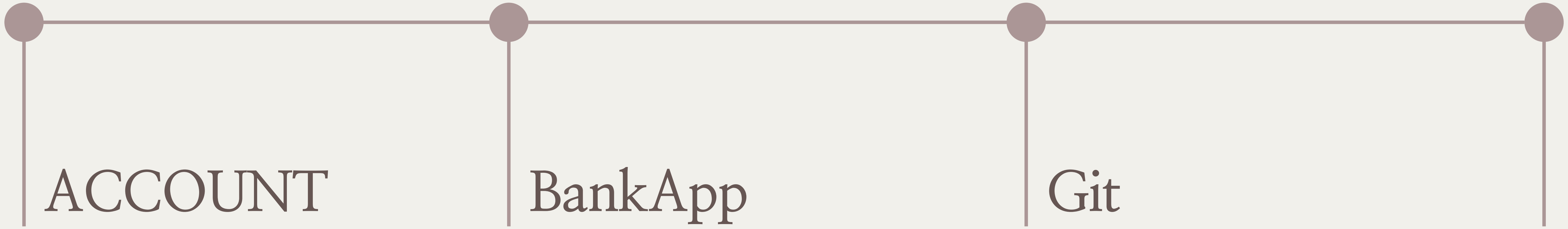
정기 수행평가 1회

박서홍

CONTENTS



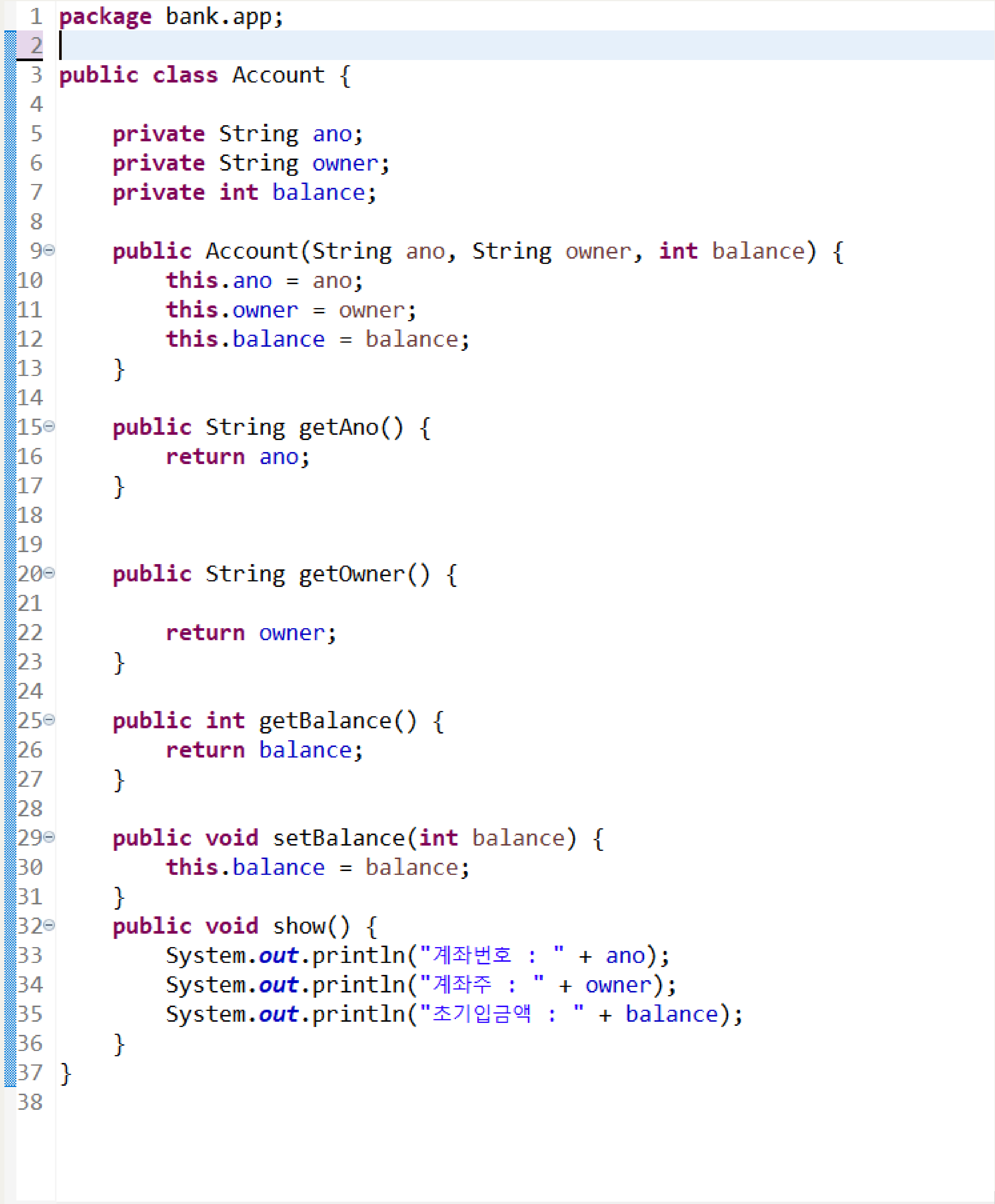
ACCOUNT

BankApp

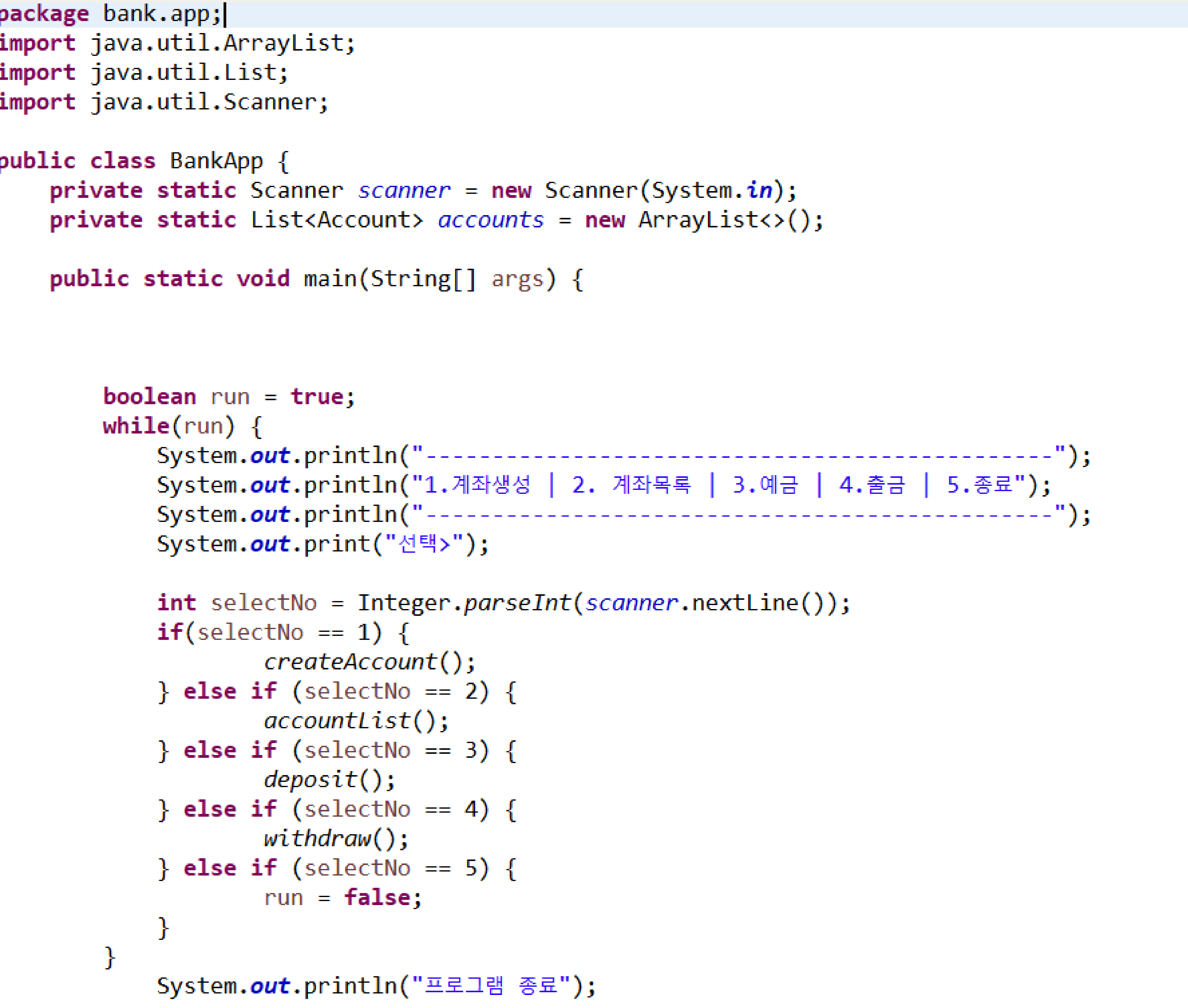
Git

Account

Account 클래스생성



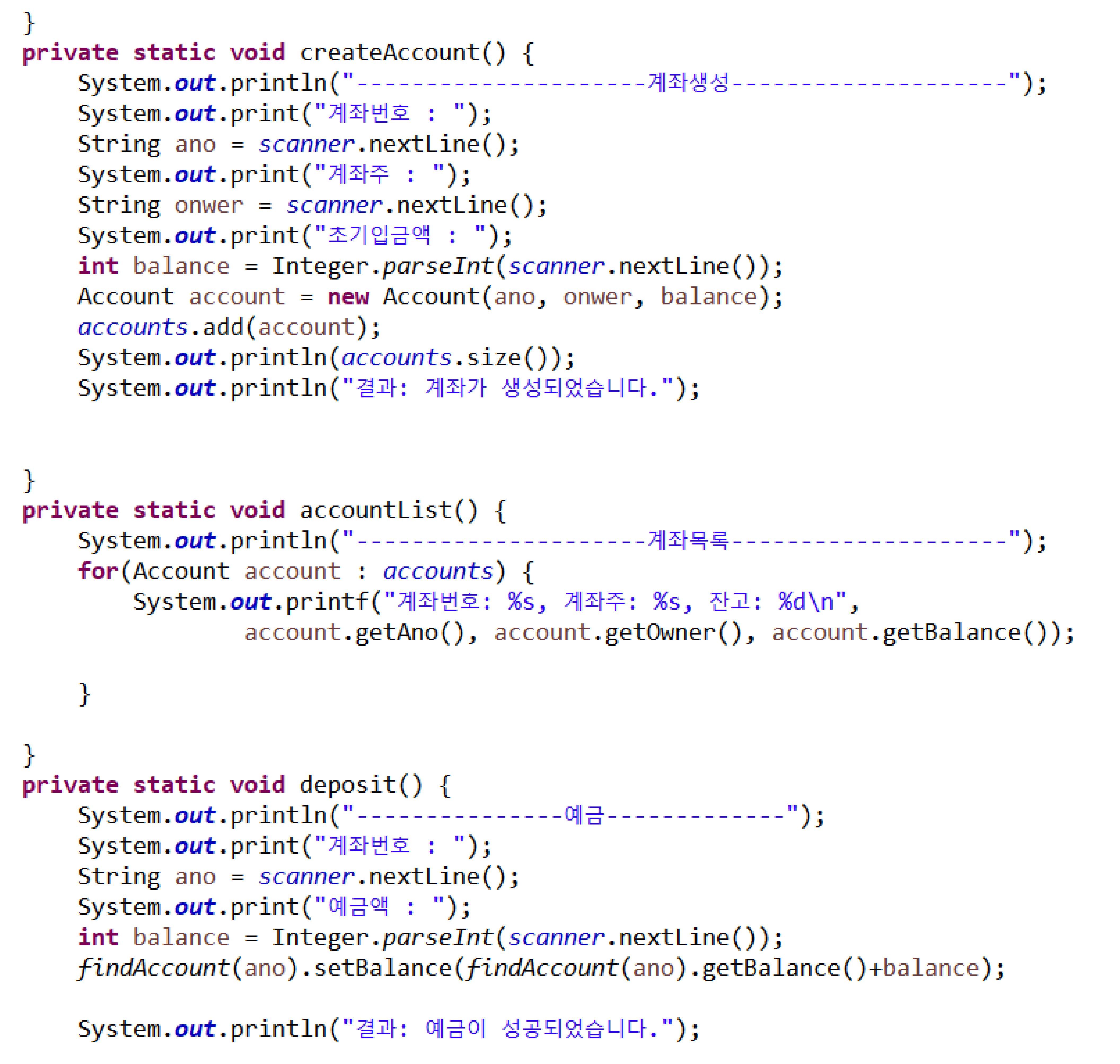
```java
package bank.app;

public class Account {

    private String ano;
    private String owner;
    private int balance;

    public Account(String ano, String owner, int balance) {
        this.ano = ano;
        this.owner = owner;
        this.balance = balance;
    }

    public String getAno() {
        return ano;
    }


    public String getOwner() {

        return owner;
    }

    public int getBalance() {
        return balance;
    }

    public void setBalance(int balance) {
        this.balance = balance;
    }
    public void show() {
        System.out.println("계좌번호 : " + ano);
        System.out.println("계좌주 : " + owner);
        System.out.println("초기입금액 : " + balance);
    }
}
```



```
package bank.app;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class BankApp {
    private static Scanner scanner = new Scanner(System.in);
    private static List<Account> accounts = new ArrayList<>();

    public static void main(String[] args) {



        boolean run = true;
        while(run) {
            System.out.println("-----------------------------------------------");
            System.out.println("1.계좌생성 | 2. 계좌목록 | 3.예금 | 4.출금 | 5.종료");
            System.out.println("-----------------------------------------------");
            System.out.print("선택>");

            int selectNo = Integer.parseInt(scanner.nextLine());
            if(selectNo == 1) {
                    createAccount();
            } else if (selectNo == 2) {
                    accountList();
            } else if (selectNo == 3) {
                    deposit();
            } else if (selectNo == 4) {
                    withdraw();
            } else if (selectNo == 5) {
                    run = false;
            }
        }
            System.out.println("프로그램 종료");
```

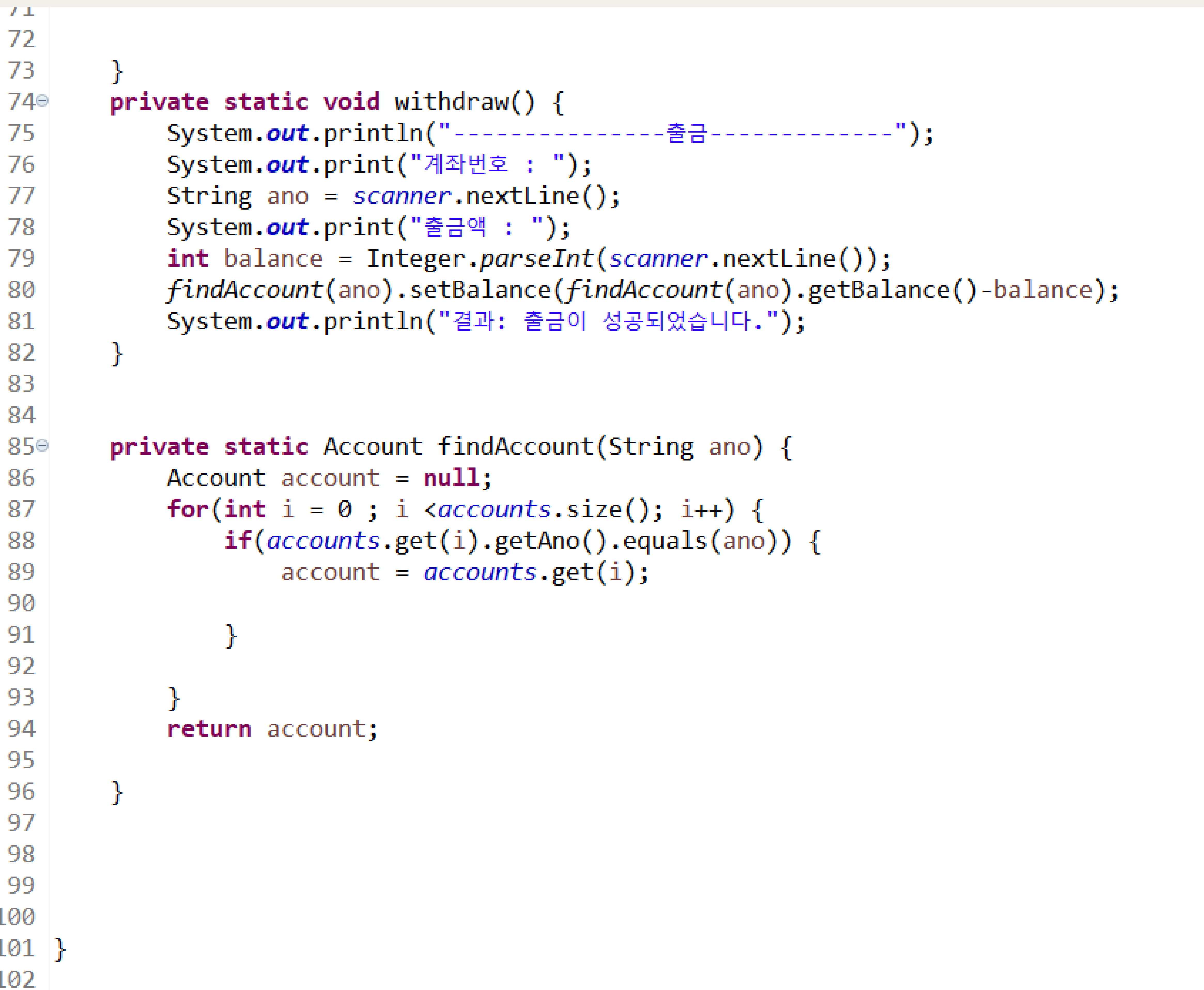
bank app
클래스생성

bank app 클래스생성



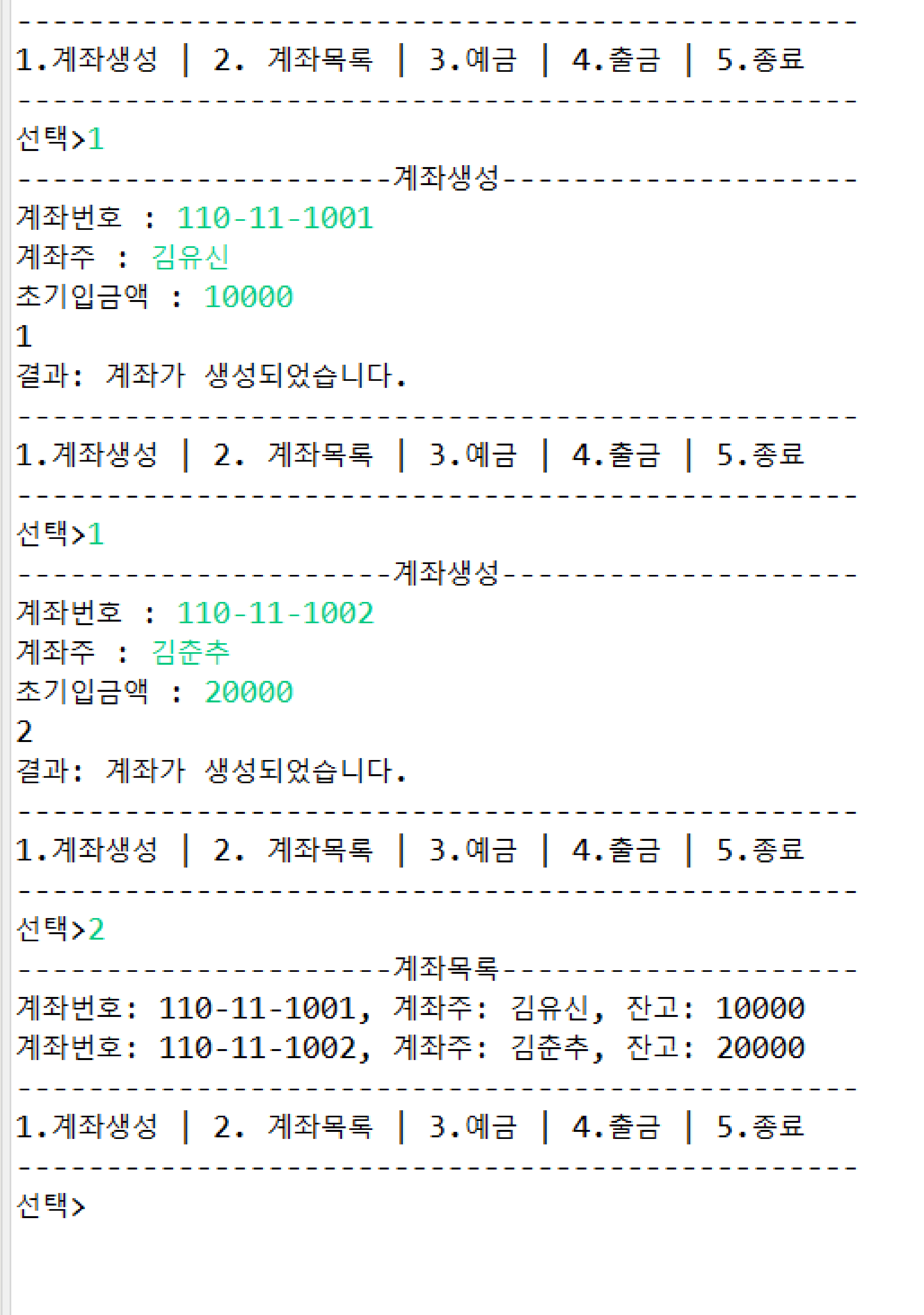
```
}
private static void createAccount() {
    System.out.println("---------------------계좌생성--------------------");
    System.out.print("계좌번호 : ");
    String ano = scanner.nextLine();
    System.out.print("계좌주 : ");
    String onwer = scanner.nextLine();
    System.out.print("초기입금액 : ");
    int balance = Integer.parseInt(scanner.nextLine());
    Account account = new Account(ano, onwer, balance);
    accounts.add(account);
    System.out.println(accounts.size());
    System.out.println("결과: 계좌가 생성되었습니다.");


}
private static void accountList() {
    System.out.println("---------------------계좌목록--------------------");
    for(Account account : accounts) {
        System.out.printf("계좌번호: %s, 계좌주: %s, 잔고: %d\n",
                account.getAno(), account.getOwner(), account.getBalance());

    }

}
private static void deposit() {
    System.out.println("---------------예금-------------");
    System.out.print("계좌번호 : ");
    String ano = scanner.nextLine();
    System.out.print("예금액 : ");
    int balance = Integer.parseInt(scanner.nextLine());
    findAccount(ano).setBalance(findAccount(ano).getBalance()+balance);

    System.out.println("결과: 예금이 성공되었습니다.");
```

bank app
클래스생성



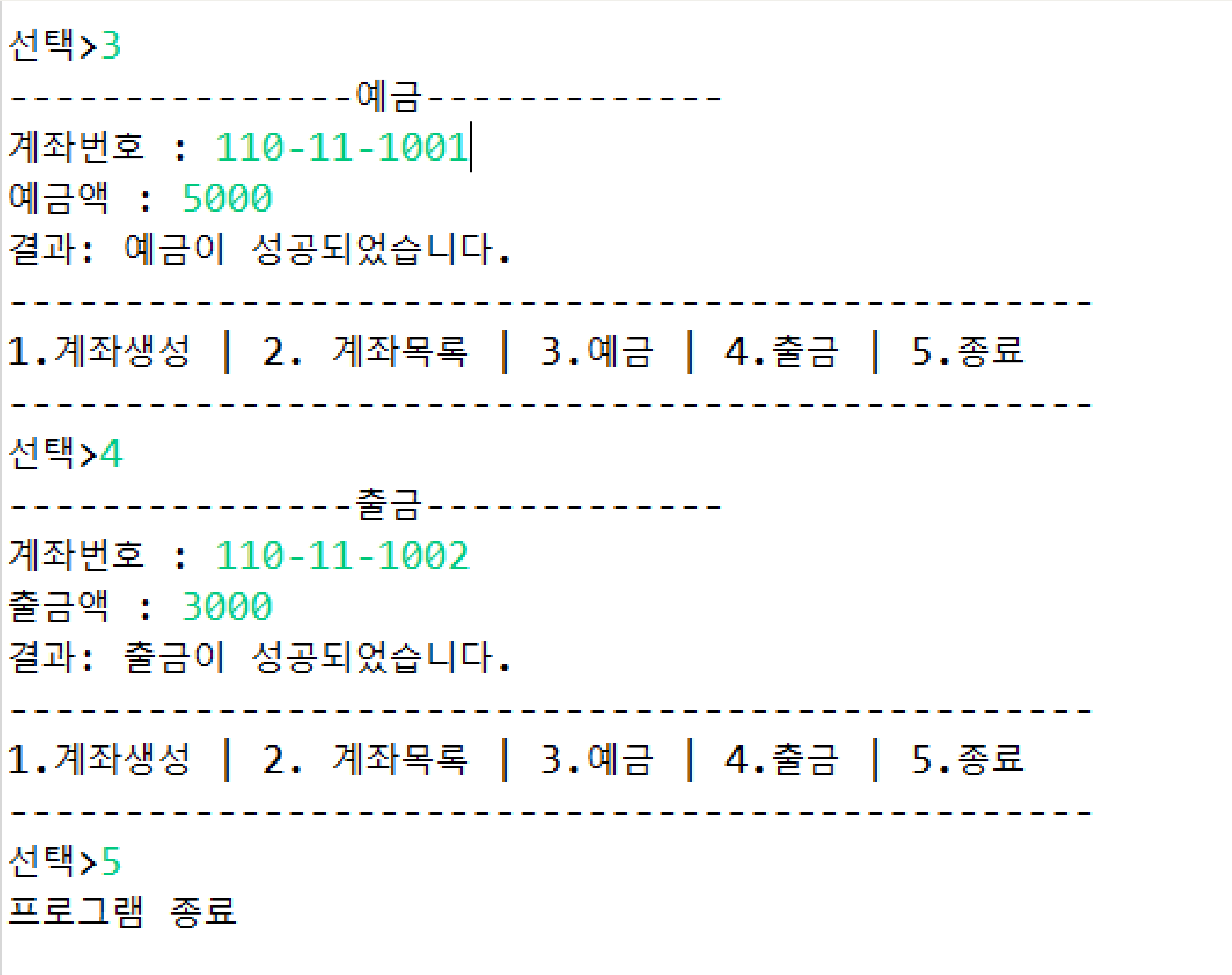
```java

    }
    private static void withdraw() {
        System.out.println("---------------출금-------------");
        System.out.print("계좌번호 : ");
        String ano = scanner.nextLine();
        System.out.print("출금액 : ");
        int balance = Integer.parseInt(scanner.nextLine());
        findAccount(ano).setBalance(findAccount(ano).getBalance()-balance);
        System.out.println("결과: 출금이 성공되었습니다.");
    }


    private static Account findAccount(String ano) {
        Account account = null;
        for(int i = 0 ; i <accounts.size(); i++) {
            if(accounts.get(i).getAno().equals(ano)) {
                account = accounts.get(i);

            }

        }
        return account;

    }




}

```

bank app 실행



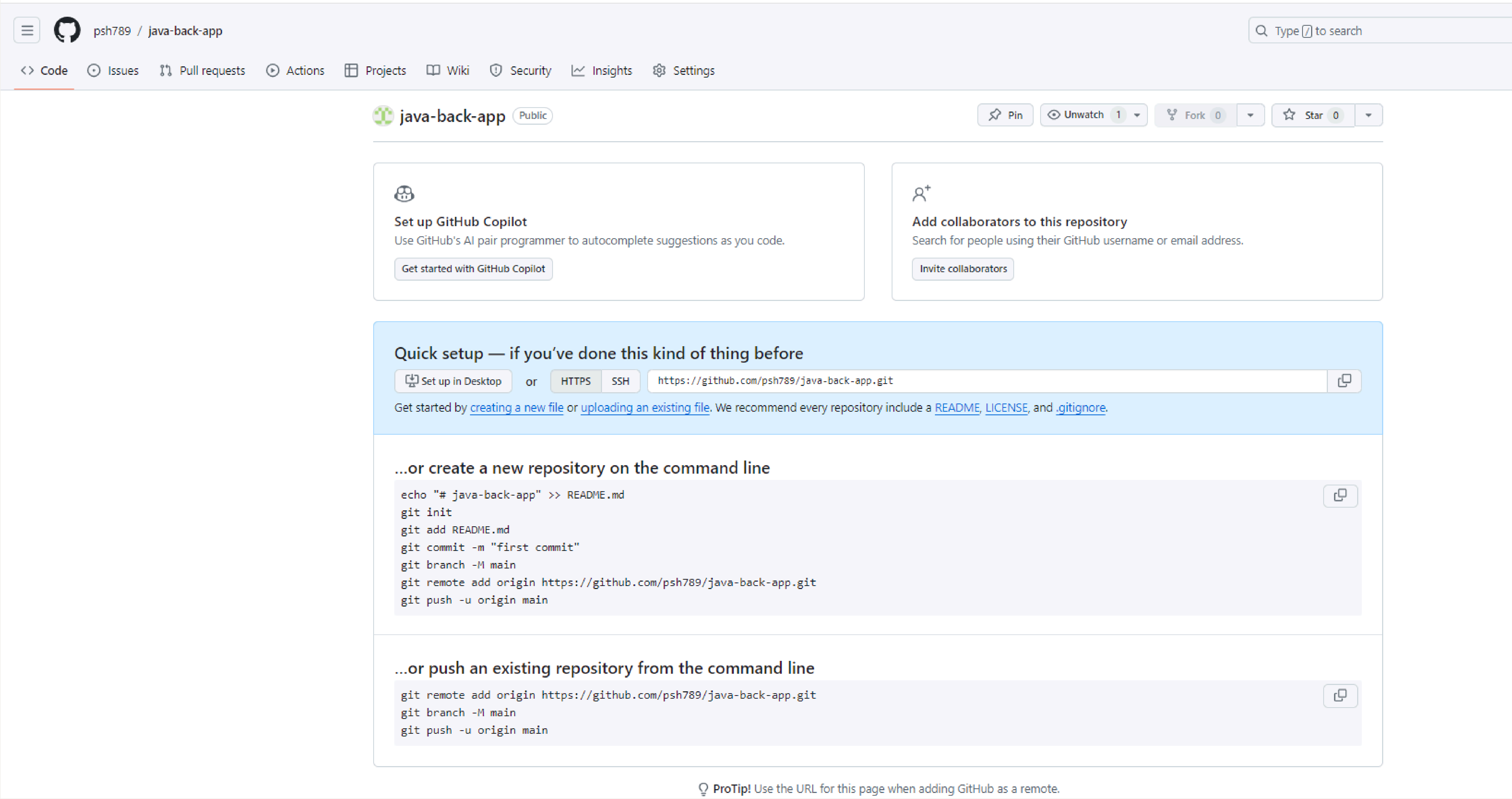
```
------------------------------------------------
1.계좌생성 | 2. 계좌목록 | 3.예금 | 4.출금 | 5.종료
------------------------------------------------
선택>1
--------------------계좌생성--------------------
계좌번호 : 110-11-1001
계좌주 : 김유신
초기입금액 : 10000
1
결과: 계좌가 생성되었습니다.
------------------------------------------------
1.계좌생성 | 2. 계좌목록 | 3.예금 | 4.출금 | 5.종료
------------------------------------------------
선택>1
--------------------계좌생성--------------------
계좌번호 : 110-11-1002
계좌주 : 김춘추
초기입금액 : 20000
2
결과: 계좌가 생성되었습니다.
------------------------------------------------
1.계좌생성 | 2. 계좌목록 | 3.예금 | 4.출금 | 5.종료
------------------------------------------------
선택>2
--------------------계좌목록--------------------
계좌번호: 110-11-1001, 계좌주: 김유신, 잔고: 10000
계좌번호: 110-11-1002, 계좌주: 김춘추, 잔고: 20000
------------------------------------------------
1.계좌생성 | 2. 계좌목록 | 3.예금 | 4.출금 | 5.종료
------------------------------------------------
선택>
```

bank app 실행



```
선택>3
---------------예금-------------
계좌번호 : 110-11-1001
예금액 : 5000
결과: 예금이 성공되었습니다.
-----------------------------------------------
1.계좌생성 | 2. 계좌목록 | 3.예금 | 4.출금 | 5.종료
-----------------------------------------------
선택>4
---------------출금-------------
계좌번호 : 110-11-1002
출금액 : 3000
결과: 출금이 성공되었습니다.
-----------------------------------------------
1.계좌생성 | 2. 계좌목록 | 3.예금 | 4.출금 | 5.종료
-----------------------------------------------
선택>5
프로그램 종료
```

git hub 폴더생성



Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Get started with GitHub Copilot

Add collaborators to this repository

Search for people using their GitHub username or email address.

Invite collaborators

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH `https://github.com/psh789/java-back-app.git`

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

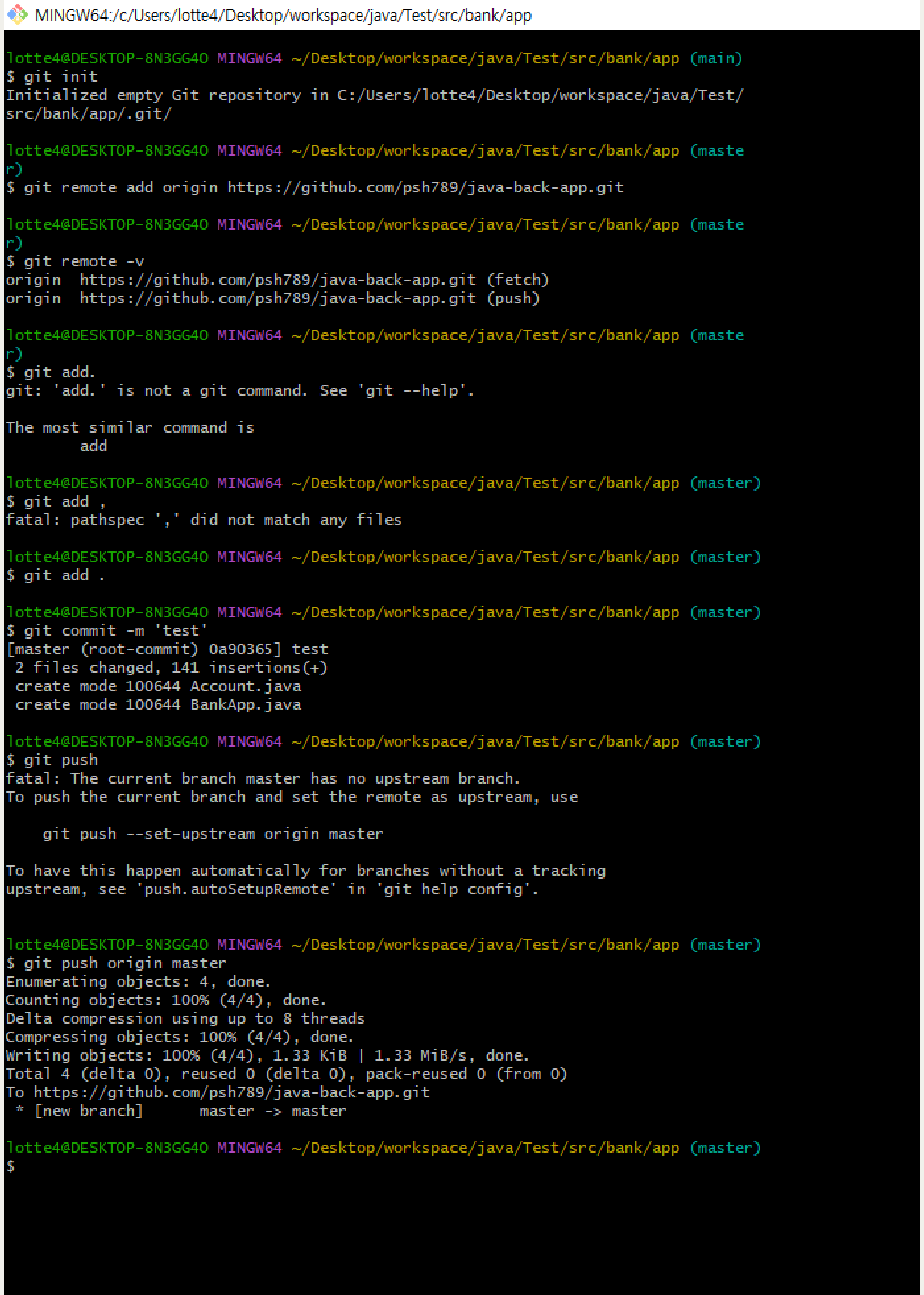
```
echo "# java-back-app" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/psh789/java-back-app.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/psh789/java-back-app.git
git branch -M main
git push -u origin main
```

ProTip! Use the URL for this page when adding GitHub as a remote.

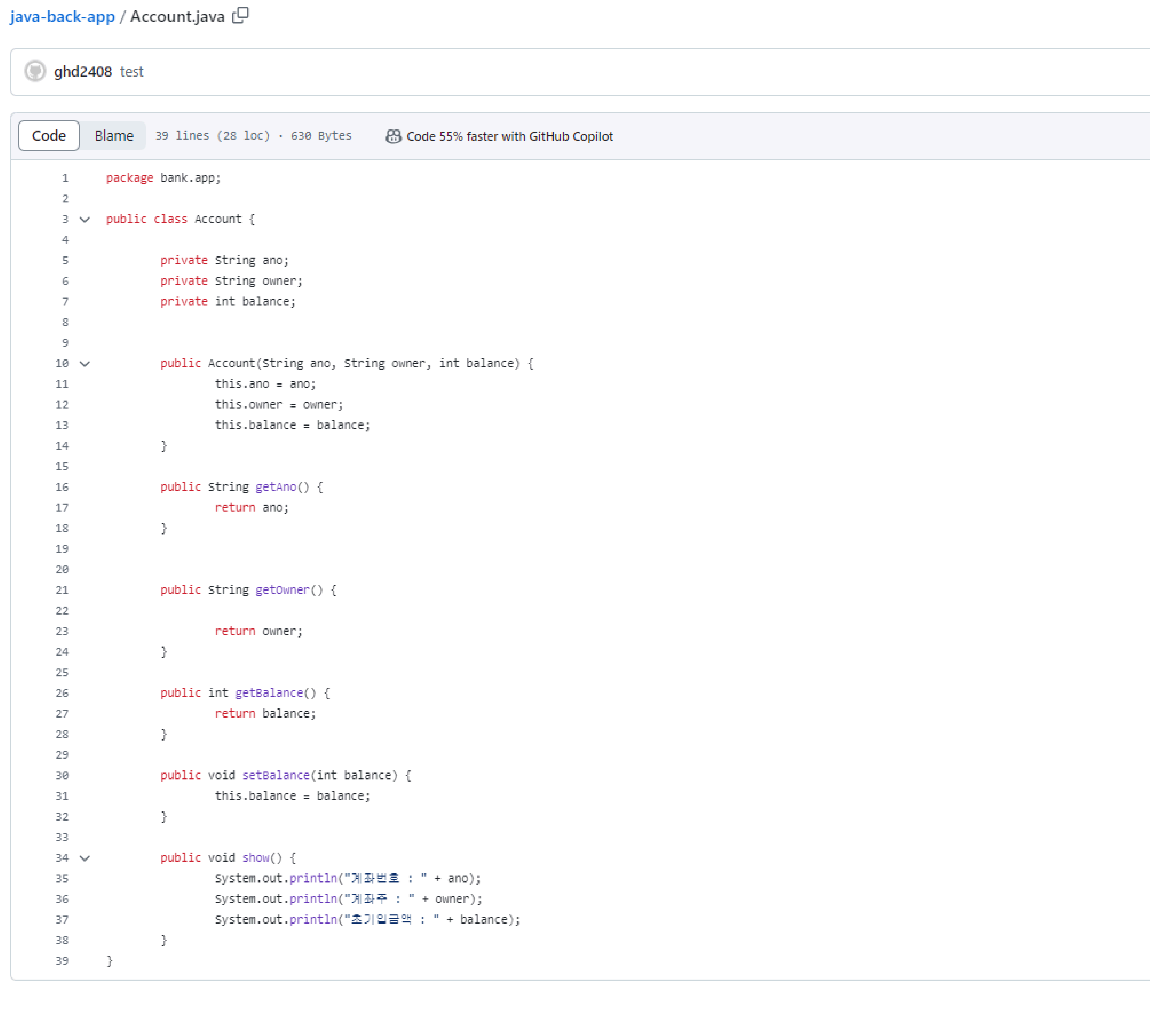
git hub 올리기



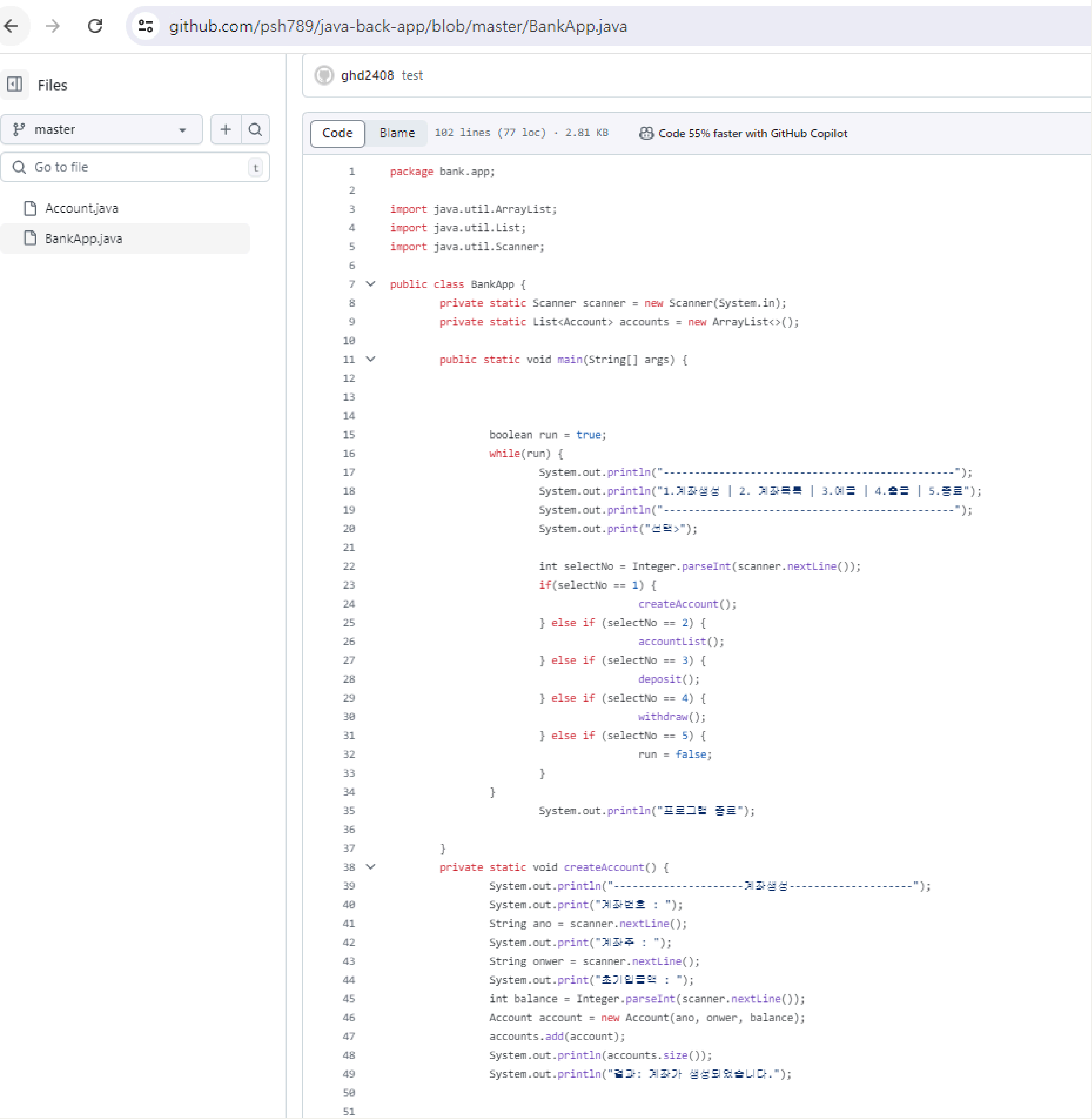
MINGW64:/c/Users/lotte4/Desktop/workspace/java/Test/src/bank/app

```
lotte4@DESKTOP-8N3GG40 MINGW64 ~/Desktop/workspace/java/Test/src/bank/app (main)
$ git init
Initialized empty Git repository in C:/Users/lotte4/Desktop/workspace/java/Test/
src/bank/app/.git/

lotte4@DESKTOP-8N3GG40 MINGW64 ~/Desktop/workspace/java/Test/src/bank/app (maste
r)
$ git remote add origin https://github.com/psh789/java-back-app.git

lotte4@DESKTOP-8N3GG40 MINGW64 ~/Desktop/workspace/java/Test/src/bank/app (maste
r)
$ git remote -v
origin  https://github.com/psh789/java-back-app.git (fetch)
origin  https://github.com/psh789/java-back-app.git (push)

lotte4@DESKTOP-8N3GG40 MINGW64 ~/Desktop/workspace/java/Test/src/bank/app (maste
r)
$ git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
        add

lotte4@DESKTOP-8N3GG40 MINGW64 ~/Desktop/workspace/java/Test/src/bank/app (master)
$ git add ,
fatal: pathspec ',' did not match any files

lotte4@DESKTOP-8N3GG40 MINGW64 ~/Desktop/workspace/java/Test/src/bank/app (master)
$ git add .

lotte4@DESKTOP-8N3GG40 MINGW64 ~/Desktop/workspace/java/Test/src/bank/app (master)
$ git commit -m 'test'
[master (root-commit) 0a90365] test
 2 files changed, 141 insertions(+)
 create mode 100644 Account.java
 create mode 100644 BankApp.java

lotte4@DESKTOP-8N3GG40 MINGW64 ~/Desktop/workspace/java/Test/src/bank/app (master)
$ git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin master

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.


lotte4@DESKTOP-8N3GG40 MINGW64 ~/Desktop/workspace/java/Test/src/bank/app (master)
$ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.33 KiB | 1.33 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/psh789/java-back-app.git
 * [new branch]      master -> master

lotte4@DESKTOP-8N3GG40 MINGW64 ~/Desktop/workspace/java/Test/src/bank/app (master)
$
```

git hub
확인

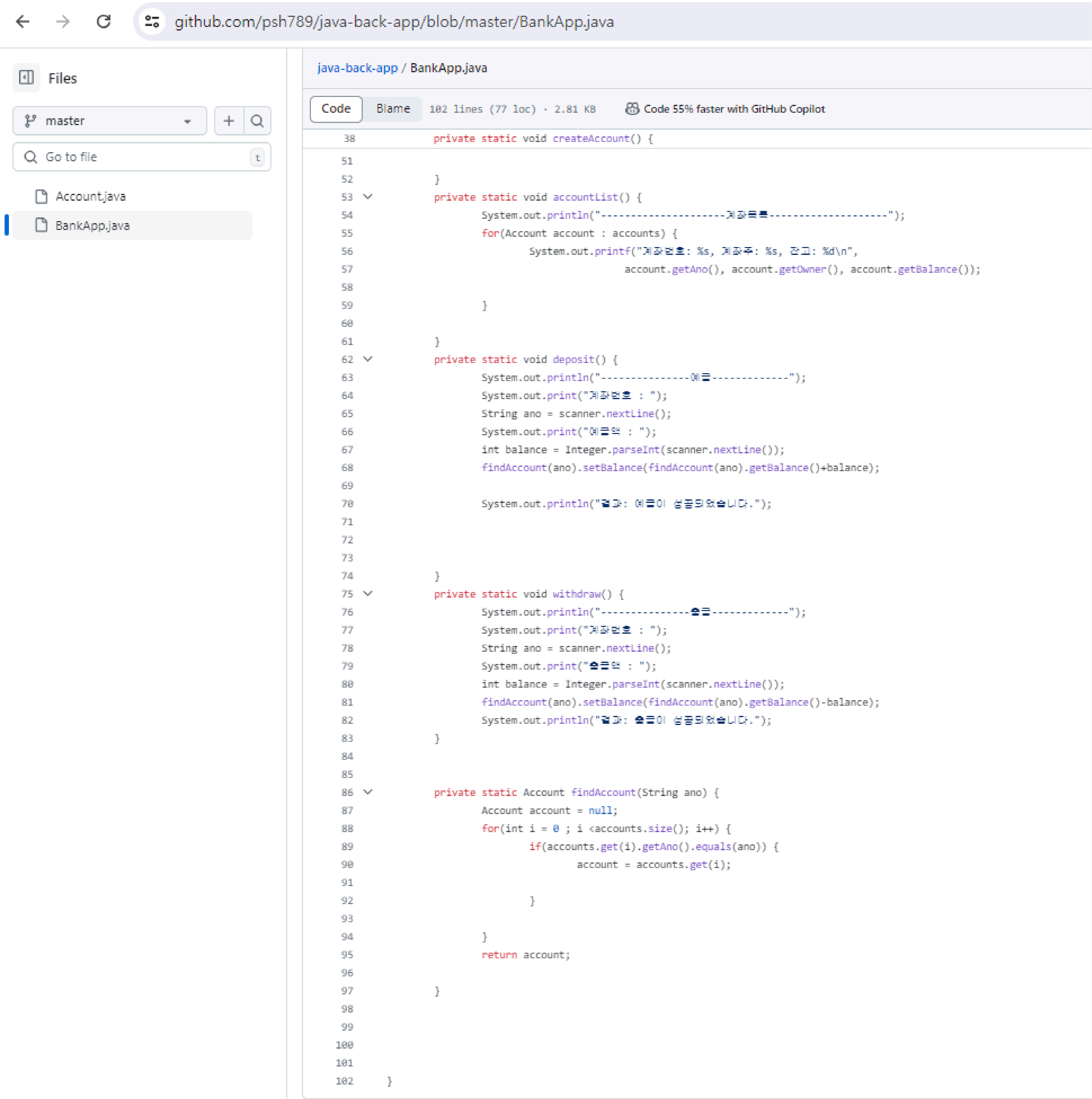

java-back-app / Account.java
ghd2408 test
Code
Blame
39 lines (28 loc) · 630 Bytes
Code 55% faster with GitHub Copilot

```java
package bank.app;

public class Account {

        private String ano;
        private String owner;
        private int balance;


        public Account(String ano, String owner, int balance) {
                this.ano = ano;
                this.owner = owner;
                this.balance = balance;
        }

        public String getAno() {
                return ano;
        }


        public String getOwner() {

                return owner;
        }

        public int getBalance() {
                return balance;
        }

        public void setBalance(int balance) {
                this.balance = balance;
        }

        public void show() {
                System.out.println("계좌번호 : " + ano);
                System.out.println("계좌주 : " + owner);
                System.out.println("초기입금액 : " + balance);
        }
}
```


github.com/psh789/java-back-app/blob/master/BankApp.java
Files
master
Go to file
t
Account.java
BankApp.java
ghd2408 test
Code
Blame
102 lines (77 loc) · 2.81 KB
Code 55% faster with GitHub Copilot

```java
package bank.app;

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class BankApp {
	private static Scanner scanner = new Scanner(System.in);
	private static List<Account> accounts = new ArrayList<>();

	public static void main(String[] args) {



		boolean run = true;
		while(run) {
			System.out.println("----------------------------------------------");
			System.out.println("1.계좌생성 | 2. 계좌목록 | 3.예금 | 4.출금 | 5.종료");
			System.out.println("----------------------------------------------");
			System.out.print("선택>");

			int selectNo = Integer.parseInt(scanner.nextLine());
			if(selectNo == 1) {
					createAccount();
			} else if (selectNo == 2) {
					accountList();
			} else if (selectNo == 3) {
					deposit();
			} else if (selectNo == 4) {
					withdraw();
			} else if (selectNo == 5) {
					run = false;
			}
		}
			System.out.println("프로그램 종료");

	}
	private static void createAccount() {
		System.out.println("--------------------계좌생성--------------------");
		System.out.print("계좌번호 : ");
		String ano = scanner.nextLine();
		System.out.print("계좌주 : ");
		String onwer = scanner.nextLine();
		System.out.print("초기입금액 : ");
		int balance = Integer.parseInt(scanner.nextLine());
		Account account = new Account(ano, onwer, balance);
		accounts.add(account);
		System.out.println(accounts.size());
		System.out.println("결과: 계좌가 생성되었습니다.");


```

git hub bank app 확인



github.com/psh789/java-back-app/blob/master/BankApp.java

Files
master
Go to file
t
Account.java
BankApp.java
java-back-app / BankApp.java
Code
Blame
102 lines (77 loc) · 2.81 KB
Code 55% faster with GitHub Copilot

```java
    private static void createAccount() {

    }
    private static void accountList() {
        System.out.println("--------------------계좌목록-------------------");
        for(Account account : accounts) {
            System.out.printf("계좌번호: %s, 계좌주: %s, 잔고: %d\n",
                    account.getAno(), account.getOwner(), account.getBalance());

        }

    }
    private static void deposit() {
        System.out.println("--------------예금------------");
        System.out.print("계좌번호 : ");
        String ano = scanner.nextLine();
        System.out.print("예금액 : ");
        int balance = Integer.parseInt(scanner.nextLine());
        findAccount(ano).setBalance(findAccount(ano).getBalance()+balance);

        System.out.println("결과: 예금이 성공되었습니다.");



    }
    private static void withdraw() {
        System.out.println("--------------출금------------");
        System.out.print("계좌번호 : ");
        String ano = scanner.nextLine();
        System.out.print("출금액 : ");
        int balance = Integer.parseInt(scanner.nextLine());
        findAccount(ano).setBalance(findAccount(ano).getBalance()-balance);
        System.out.println("결과: 출금이 성공되었습니다.");
    }


    private static Account findAccount(String ano) {
        Account account = null;
        for(int i = 0 ; i <accounts.size(); i++) {
            if(accounts.get(i).getAno().equals(ano)) {
                account = accounts.get(i);

            }

        }
        return account;

    }





}
```

git hub bank app 확인