BankAccount in Java

```
class BankAccount {
  private int balance;
  private String accNum;
  public BankAccount(String a) {
    accNum = a;
    balance = 0;
  } //constructor
  public void deposit(int amount) {
    balance = balance + amount;
  } // deposit
  public int getBalance() {
    return balance;
  } // getBalance
} // BankAccount
```

```
  class BankClient {
  private BankAccount bk1
         = new BankAccount(“a123”);
  ...
  public void cutDebt(int m) {
           int am1 = bk1.getBalance();
           if (am1 < 0)
                bk1.deposit(m);
  } // cutDebt
} // BankClient
```

BankAccount in Pascal

```
program BankAccountEx(input, output);
  type BankAccount =
    record
      balance: Integer;
      accNum: String;
    end;
  var bk1, bk2: BankAccount;
      am1: Integer;
  procedure mkBankAccount
   (var b: BankAccount; a: String);
  begin
    b.balance := 0;
    b.accNum := a;
  end {mkBankAccount};
```

```
    procedure deposit(
      var b: BankAccount; amount: Integer);
    begin
      b.balance := b.balance + amount;
    end {deposit};
    function getBalance(b:BankAccount):Integer;
    begin
      getBalance := b.balance;
    end {getBalance};
  begin
    mkBankAccount(bk1, “1234”);
    ... deposit(bk1, 6);
    ... am1 := getBalance(bk1); ...
  end.
```

Packages in Ada

```
package BankAccounts is
  type BankAccount is private;
  procedure mkBankAccount(b: out BankAccount;
                          a: in Integer);
  procedure deposit(b: in out BankAccount;
                          amount: in Integer);
  function getBalance(b: BankAccount)
                          return Integer;
private
  type BankAccount is
    record
      balance: Integer;
      accNum: Integer;
    end record;
end BankAccounts;
```

Packages in Ada

Require also a ***package body***.

```
package body BankAccounts is
  -- definitions of mkBankAccount,
  -- deposit and getBalance
end BankAccounts;
```

C++ example

```
class Cost {
private:
  int cents, dollars;
public:
  Cost(int d, int c) {
    dollars = d;
    cents = c;
  } //constructor
  void add(int d, int c) {
    dollars += d;
    cents += c;
  } //add
  int getDollars() const {return dollars;}
  int getCents() const {return cents;}
};
```

C++

```
void main() {
  Cost dress(45, 95);
  Cost* book = new Cost(15, 50);
  ...
  dress.add(5, 0);
  book->add(3, 15);
  ...
}
```

Java equivalent

```
class Cost {
  private int cents, dollars;
  public Cost(int d, int c){
    dollars = d;
    cents = c;
  } //constructor
  public void add(int d, int c) {
    dollars += d;
    cents += c;
  } //add
  public int getDollars() {return dollars;}
  public int getCents() {return cents;}
} // Cost
```

```
public class Example {
    private Cost dress = new Cost(45, 95);
    private Cost book = new Cost(15, 50);

    public static void main(String [] args) {
        Example ex = new Example();
    } // main

    public Example() {
        ...
        dress.add(5, 0);
        book.add(3, 15);
        ...
    } // constructor
    ...
} // Example
```