

EÖTVÖS LORÁND
UNIVERSITY | BUDAPEST

Class Relationship

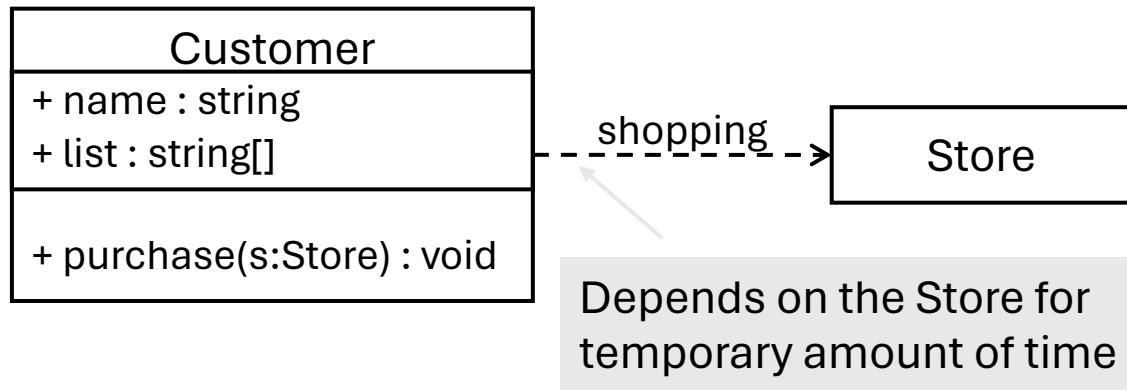
Association, Composition, Aggregation
Practice Concrete Examples

Object Oriented Programming | 2024 Spring
Practice 9

Presented By Tarlan Ahadli
Supervised By Prof. Teréz Anna Várkonyi

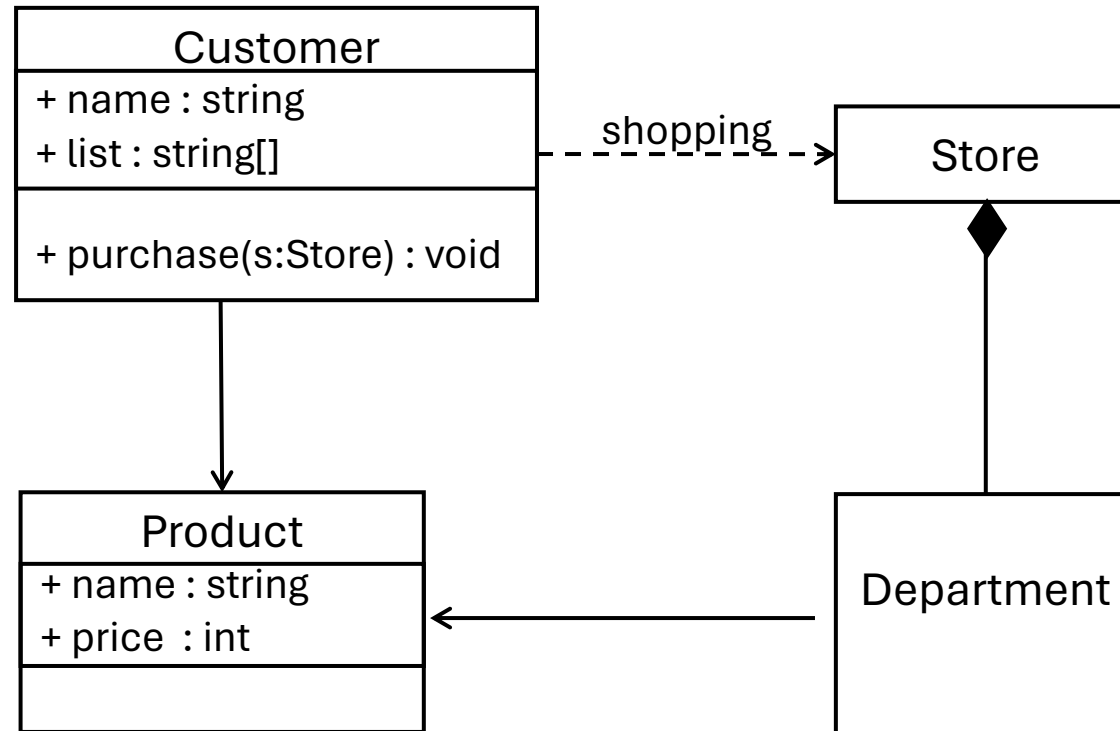
Problem 1

A small-town store has a food and a technical department. The customers have a shopping list, which contains the name of the products they would like to buy. They buy all the products on their list they can find, but they do this differently at the two departments. At the food department, they just put the first product they find into their cart. At the technical department, they always buy the cheapest product from among products with the same name. Simulate the purchasing process.



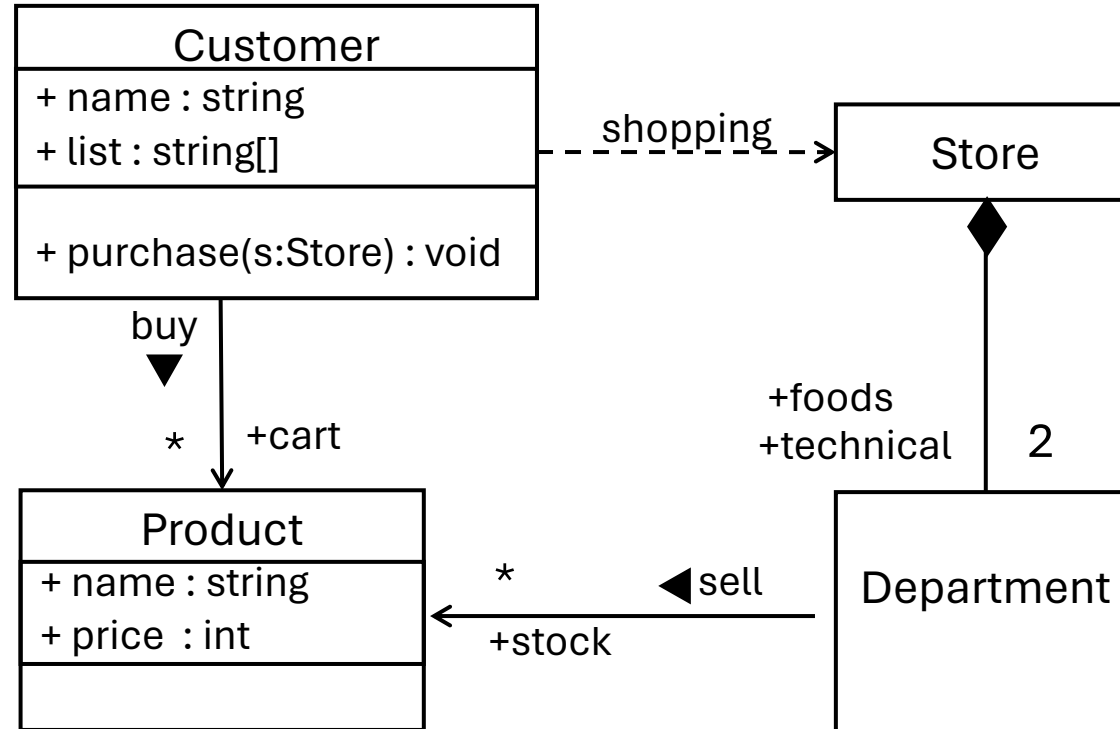
Problem 1

A small-town store has a food and a technical department. The customers have a shopping list, which contains the name of the products they would like to buy. They buy all the products on their list they can find, but they do this differently at the two departments. At the food department, they just put the first product they find into their cart. At the technical department, they always buy the cheapest product from among products with the same name. Simulate the purchasing process.



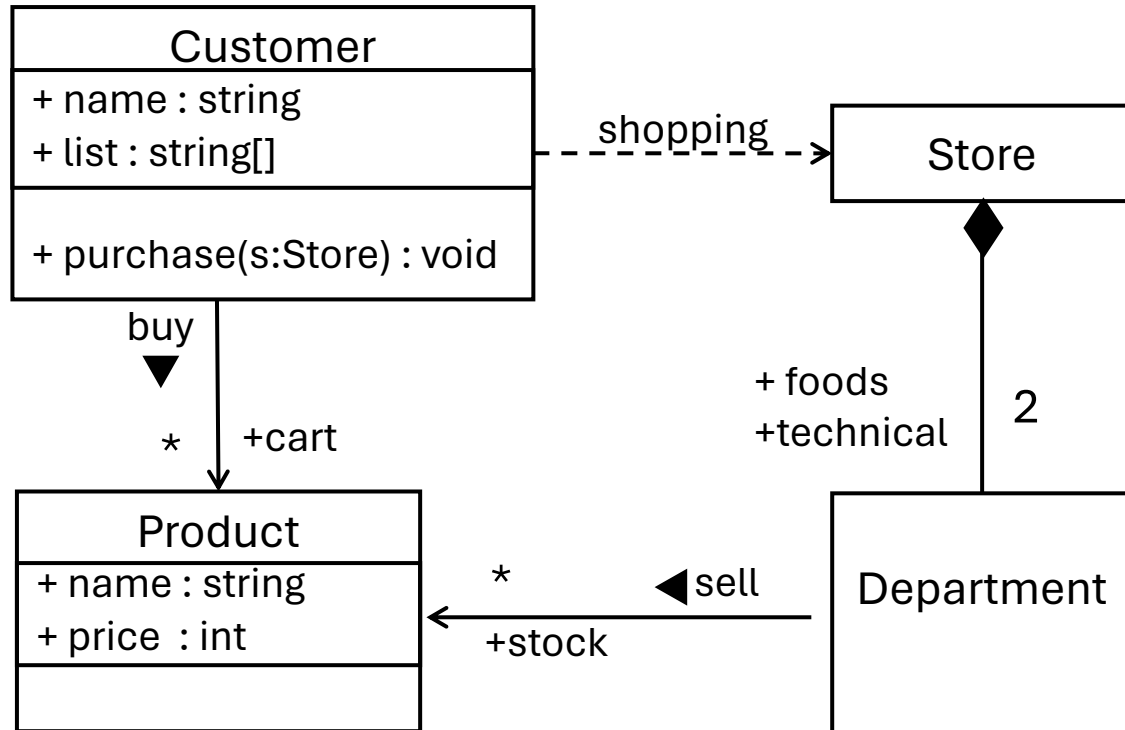
Problem 1

A small-town store has a food and a technical department. The customers have a shopping list, which contains the name of the products they would like to buy. They buy all the products on their list they can find, but they do this differently at the two departments. At the food department, they just put the first product they find into their cart. At the technical department, they always buy the cheapest product from among products with the same name. Simulate the purchasing process.



Problem 1

A small-town store has a food and a technical department. The customers have a shopping list, which contains the name of the products they would like to buy. They buy all the products on their list they can find, but they do this differently at the two departments. At the food department, they just put the first product they find into their cart. At the technical department, they always buy the cheapest product from among products with the same name. Simulate the purchasing process.



IDEA: He looks through his list in the food department and tries to find a product with the same name for each of the items. If successful, he puts the product into his cart.

After this, the same list is looked through again in the technical department. He finds the cheapest product with the same name for each of the items and puts it into his cart.

Food Department : Linear Search

forall name **in** list **loop:**

l, product := SEARCH $p \in s.\text{foods.stock}$ name = p.name

if l **then**

cart.insert(product)

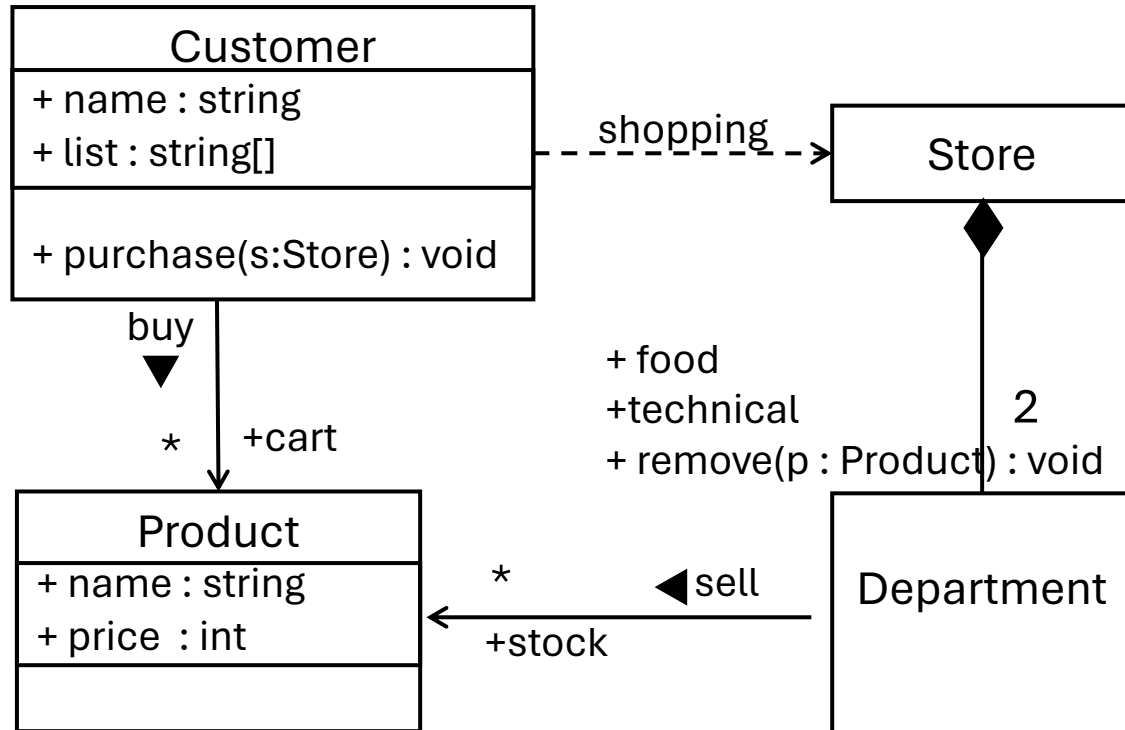
s.foods.remove(product)

endif

endloop

Problem 1

A small-town store has a food and a technical department. The customers have a shopping list, which contains the name of the products they would like to buy. They buy all the products on their list they can find, but they do this differently at the two departments. At the food department, they just put the first product they find into their cart. At the technical department, they always buy the cheapest product from among products with the same name. Simulate the purchasing process.



IDEA: He looks through his list in the food department and tries to find a product with the same name for each of the items. If successful, he puts the product into his cart.

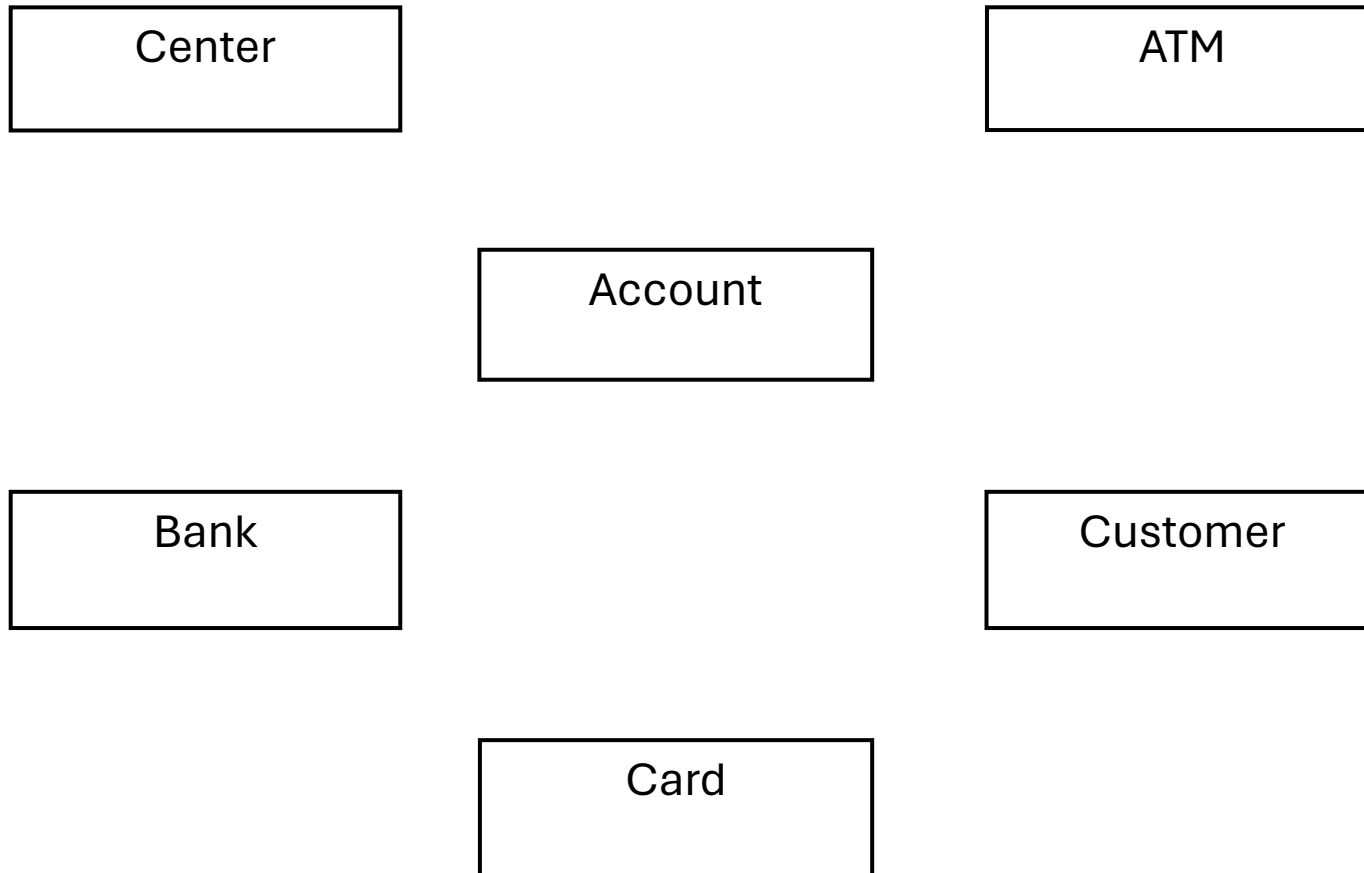
After this, the same list is looked through again in the technical department. He finds the cheapest product with the same name for each of the items and puts it into his cart.

Tech. Depaertment : Cond. Min. Search

```
forall name in list loop
    l, min, product := MIN p ∈ s.technical.stock {name = p.name} p.price
    if l then
        cart.insert(product)
        s.technical.remove(product)
    endif
endloop
```

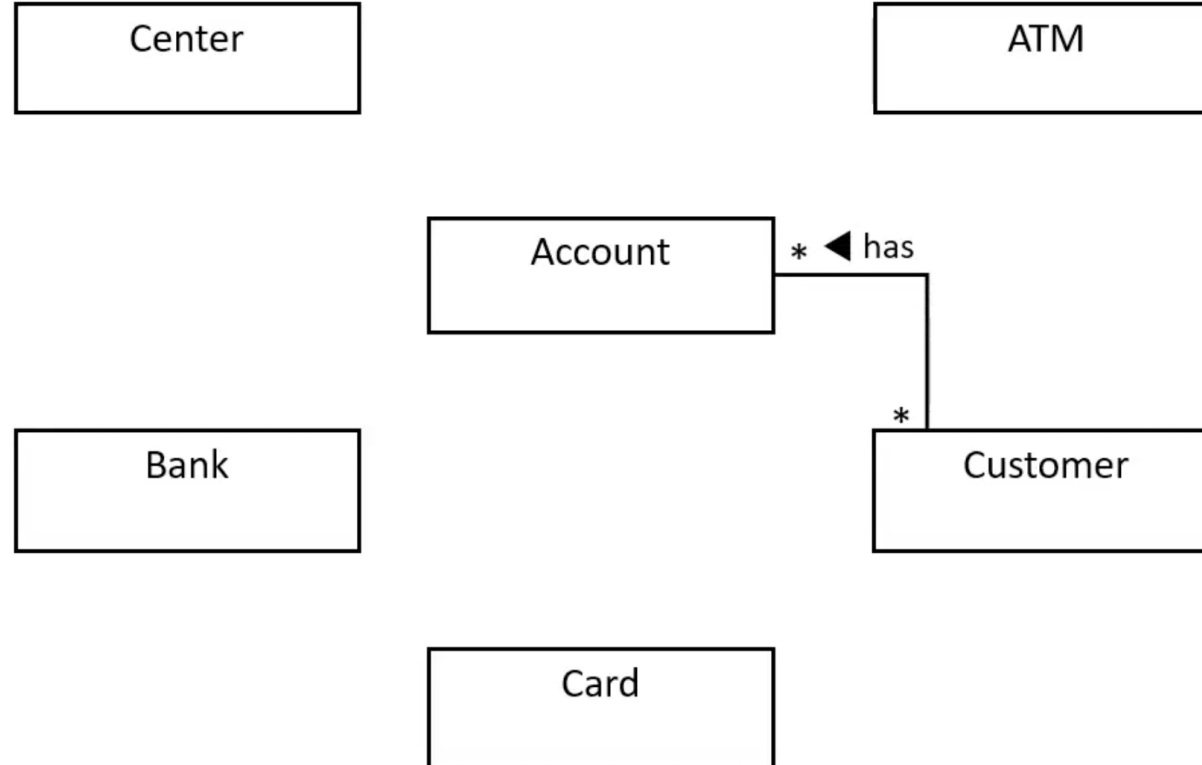
Problem 2

Customers are standing in line at an ATM to withdraw money from their accounts. The clients have debit cards, each of which has a PIN code and an account. The clients can withdraw money one after another by giving their card and code. If the given code is the same as the code of the card, the ATM gives out the money, as long as subtracting the amount from the balance yields a positive number. To check this, the ATM can query the balance of the client from a center by sending the data of the card to the center. The ATM can also send a report about the transaction to the center so the bank can subtract the amount from the account. Simulate the process of withdrawing from the ATM.



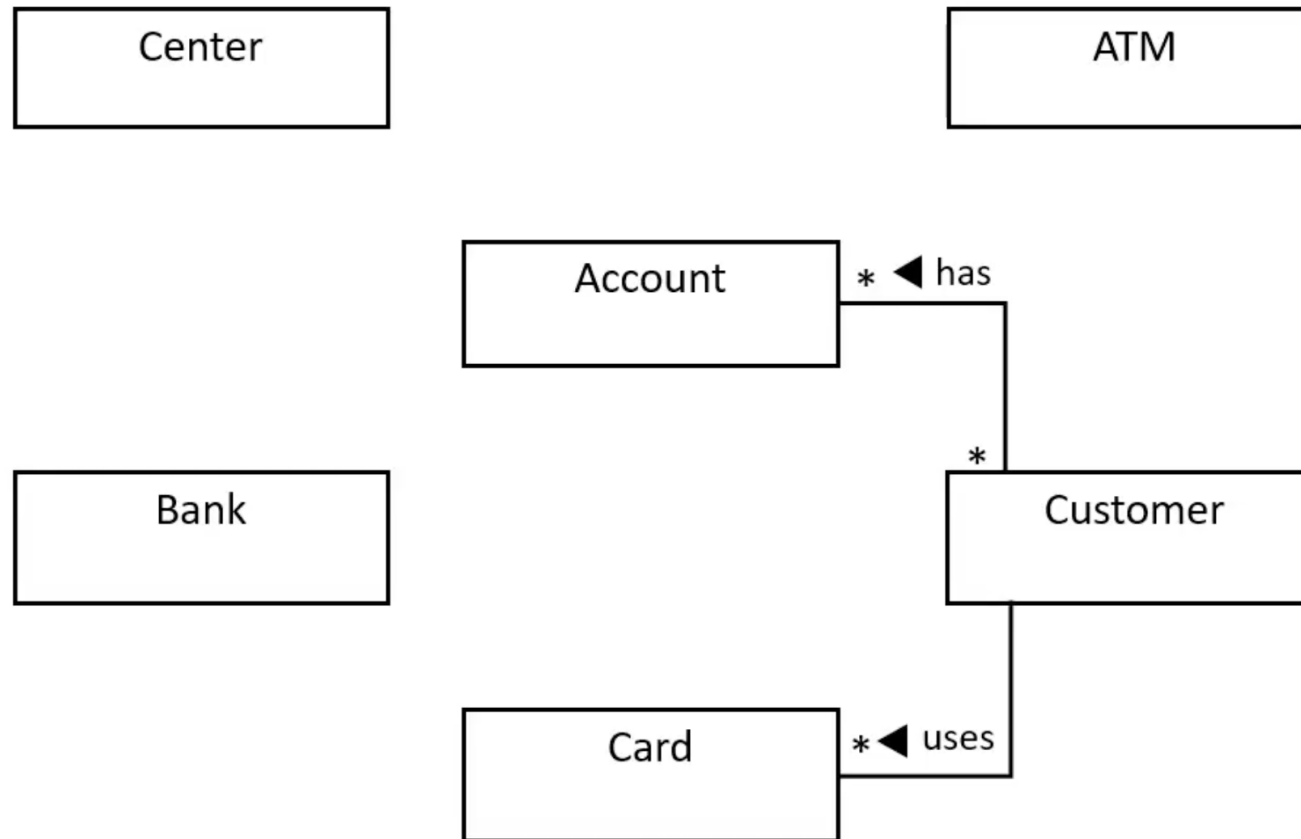
Problem 2

Customers are standing in line at an ATM to withdraw money from their accounts. The clients have debit cards, each of which has a PIN code and an account. The clients can withdraw money one after another by giving their card and code. If the given code is the same as the code of the card, the ATM gives out the money, as long as subtracting the amount from the balance yields a positive number. To check this, the ATM can query the balance of the client from a center by sending the data of the card to the center. The ATM can also send a report about the transaction to the center so the bank can subtract the amount from the account. Simulate the process of withdrawing from the ATM.



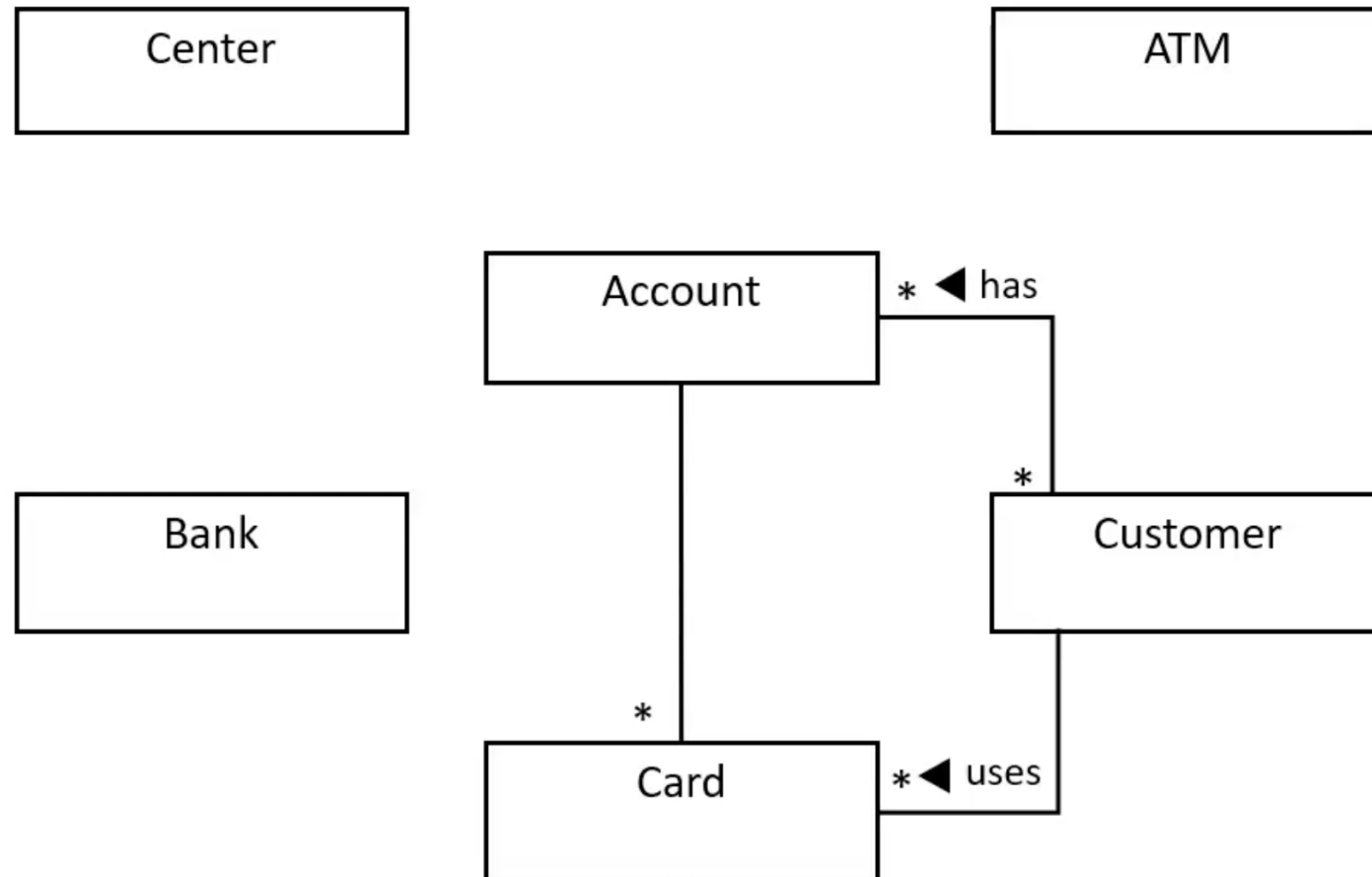
Problem 2

Customers are standing in line at an ATM to withdraw money from their accounts. The clients have debit cards, each of which has a PIN code and an account. The clients can withdraw money one after another by giving their card and code. If the given code is the same as the code of the card, the ATM gives out the money, as long as subtracting the amount from the balance yields a positive number. To check this, the ATM can query the balance of the client from a center by sending the data of the card to the center. The ATM can also send a report about the transaction to the center so the bank can subtract the amount from the account. Simulate the process of withdrawing from the ATM.



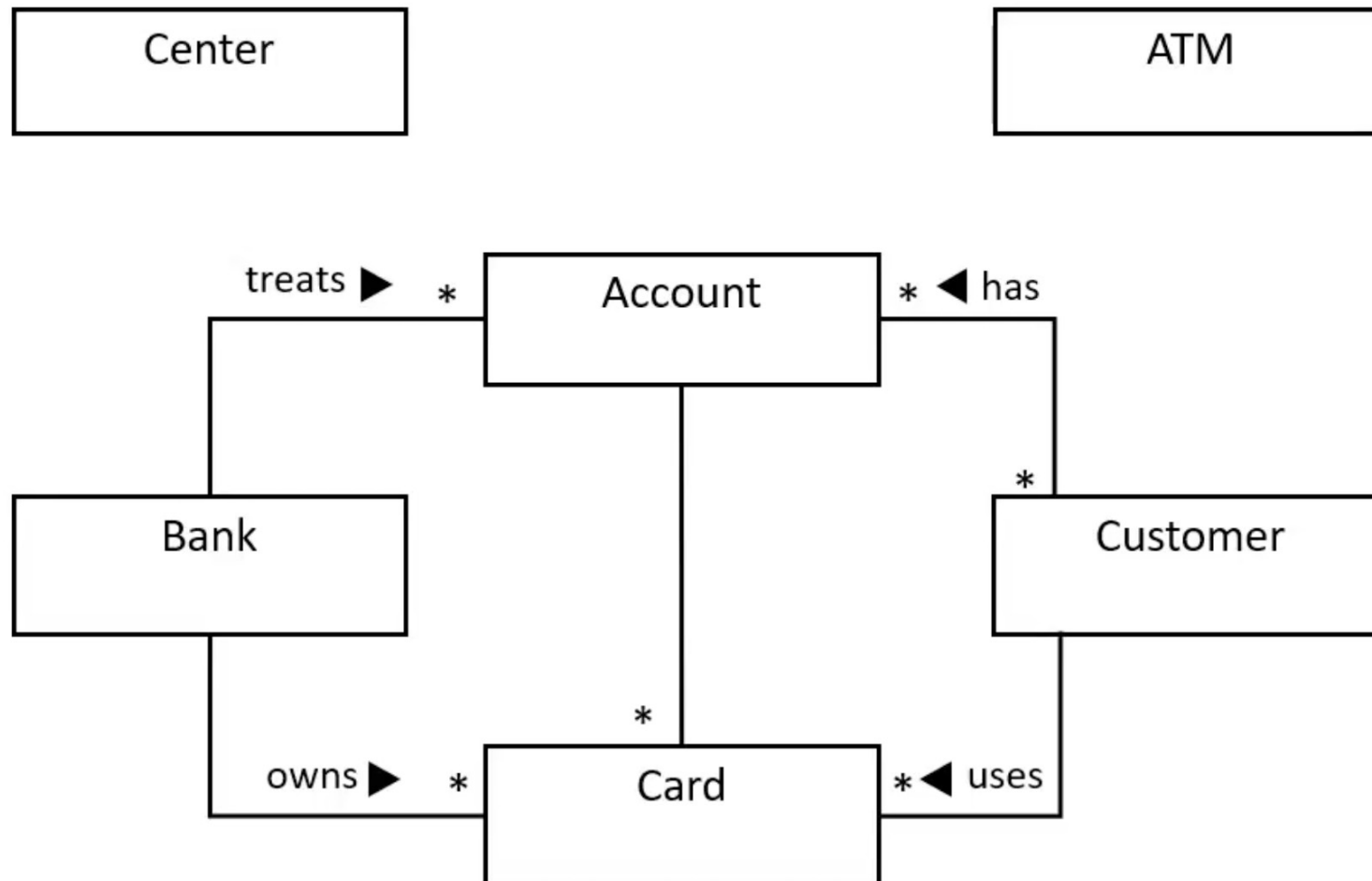
Problem 2

Customers are standing in line at an ATM to withdraw money from their accounts. The clients have debit cards, each of which has a PIN code and an account. The clients can withdraw money one after another by giving their card and code. If the given code is the same as the code of the card, the ATM gives out the money, as long as subtracting the amount from the balance yields a positive number. To check this, the ATM can query the balance of the client from a center by sending the data of the card to the center. The ATM can also send a report about the transaction to the center so the bank can subtract the amount from the account. Simulate the process of withdrawing from the ATM.



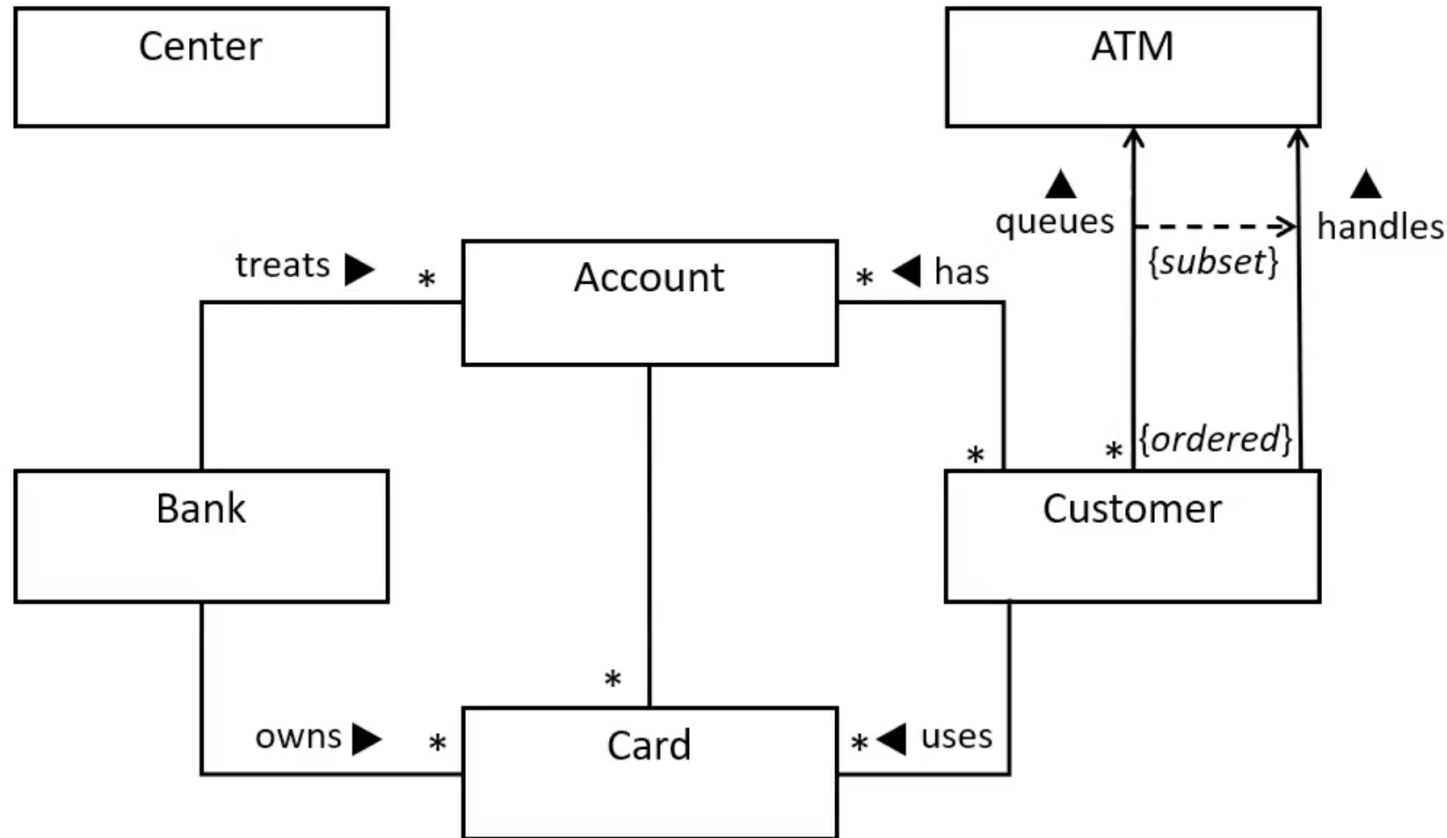
Problem 2

Customers are standing in line at an ATM to withdraw money from their accounts. The clients have debit cards, each of which has a PIN code and an account. The clients can withdraw money one after another by giving their card and code. If the given code is the same as the code of the card, the ATM gives out the money, as long as subtracting the amount from the balance yields a positive number. To check this, the ATM can query the balance of the client from a center by sending the data of the card to the center. The ATM can also send a report about the transaction to the center so the bank can subtract the amount from the account. Simulate the process of withdrawing from the ATM.



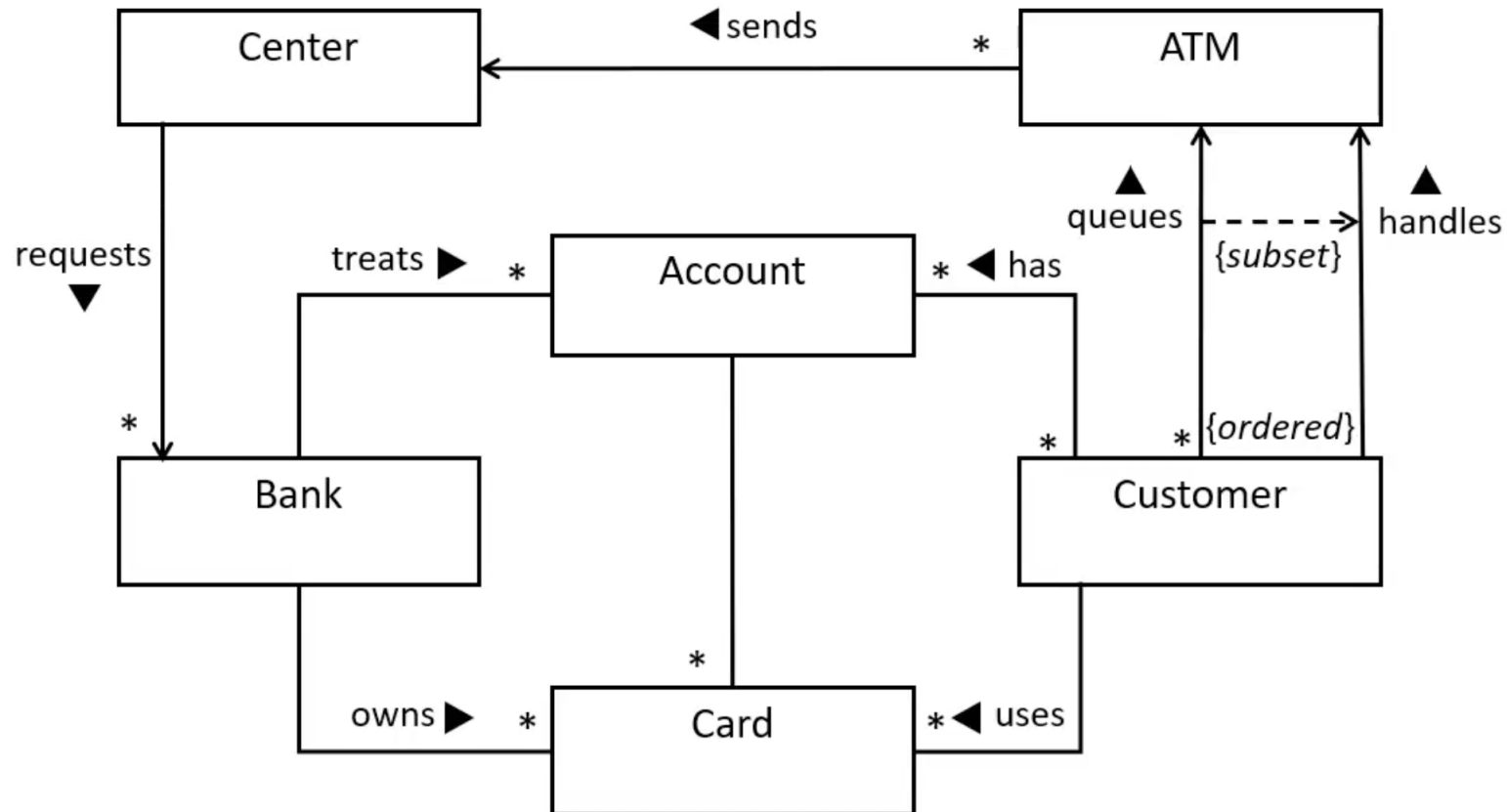
Problem 2

Customers are standing in line at an ATM to withdraw money from their accounts. The clients have debit cards, each of which has a PIN code and an account. The clients can withdraw money one after another by giving their card and code. If the given code is the same as the code of the card, the ATM gives out the money, as long as subtracting the amount from the balance yields a positive number. To check this, the ATM can query the balance of the client from a center by sending the data of the card to the center. The ATM can also send a report about the transaction to the center so the bank can subtract the amount from the account. Simulate the process of withdrawing from the ATM.



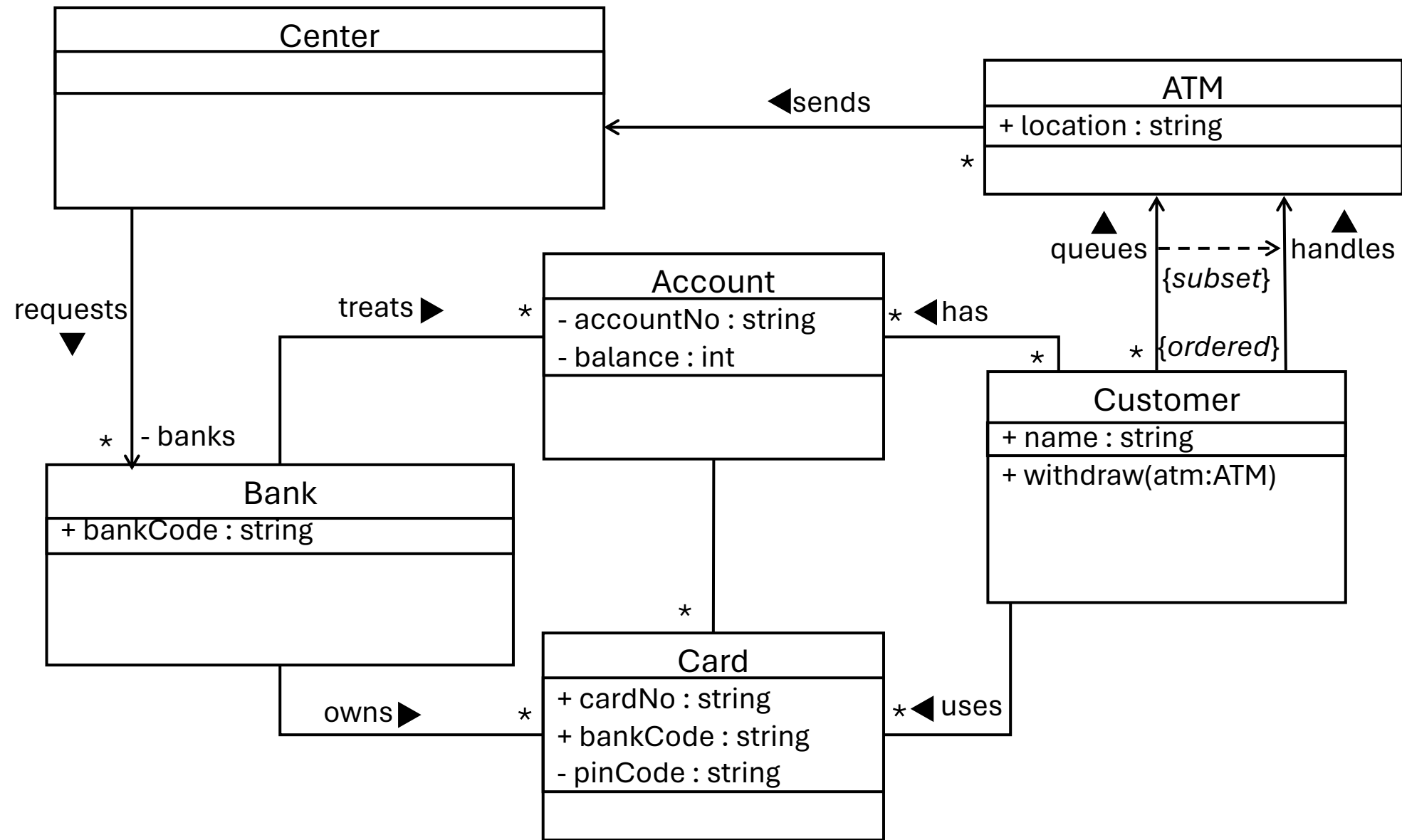
Problem 2

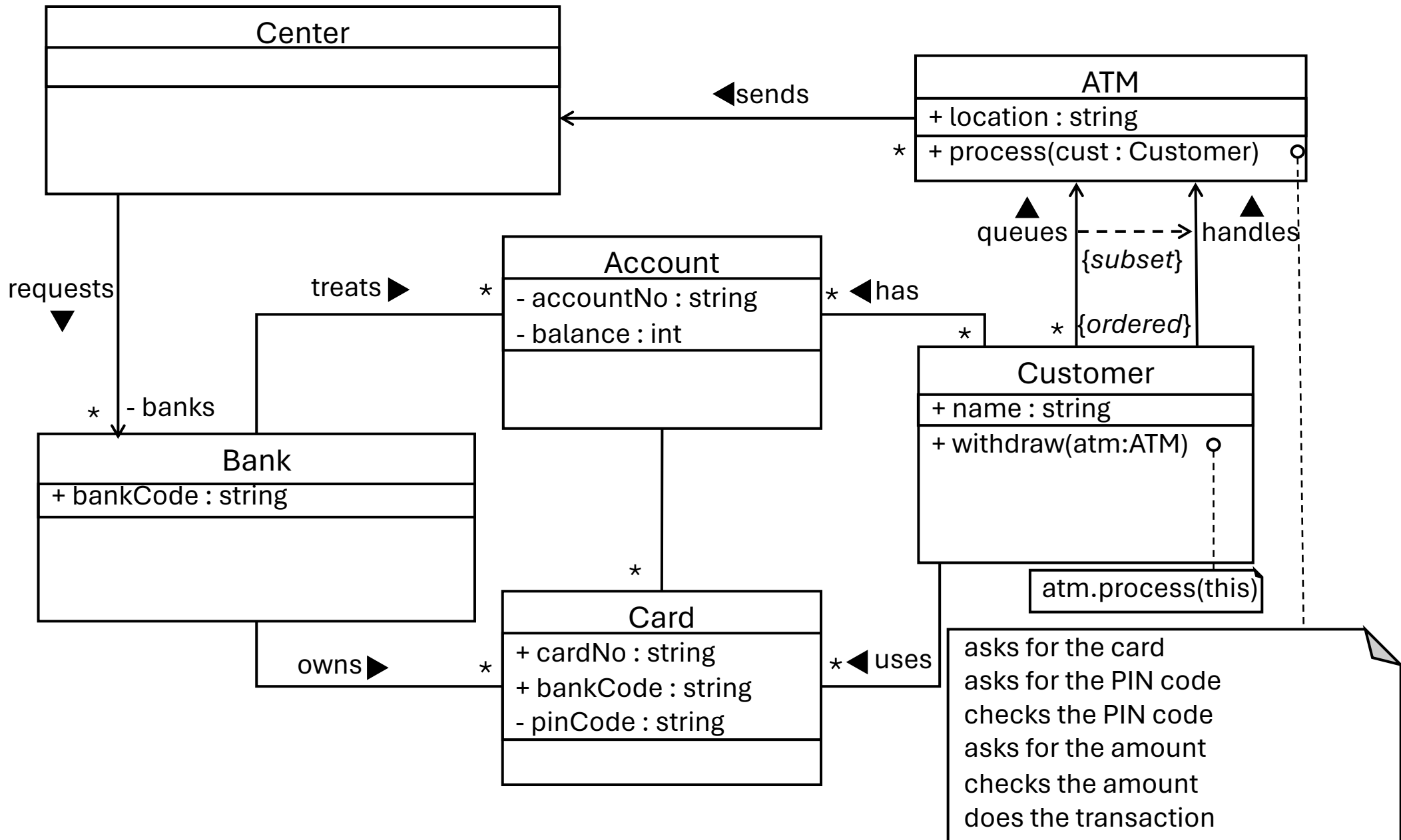
Customers are standing in line at an ATM to withdraw money from their accounts. The clients have debit cards, each of which has a PIN code and an account. The clients can withdraw money one after another by giving their card and code. If the given code is the same as the code of the card, the ATM gives out the money, as long as subtracting the amount from the balance yields a positive number. To check this, the ATM can query the balance of the client from a center by sending the data of the card to the center. The ATM can also send a report about the transaction to the center so the bank can subtract the amount from the account. Simulate the process of withdrawing from the ATM.

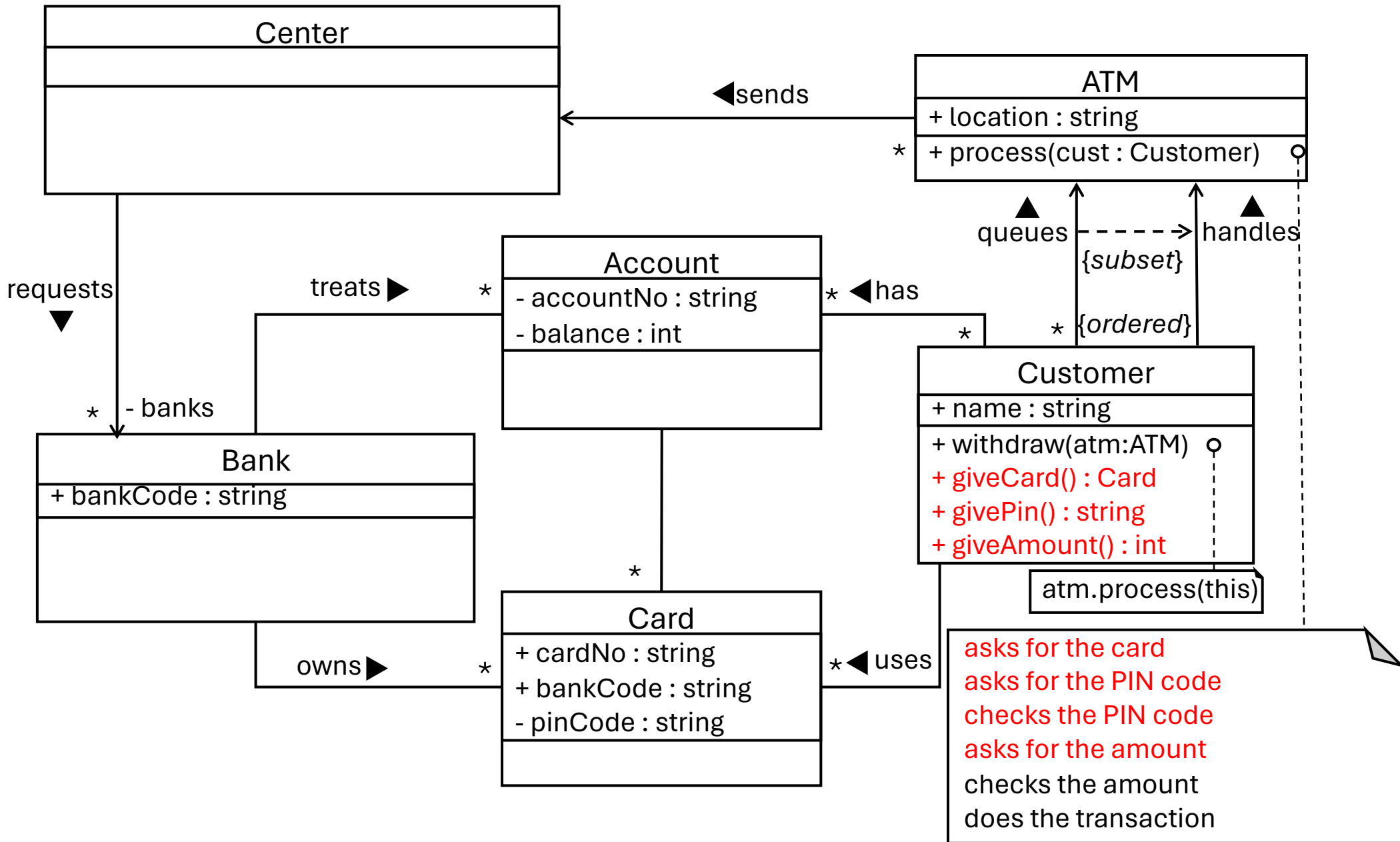


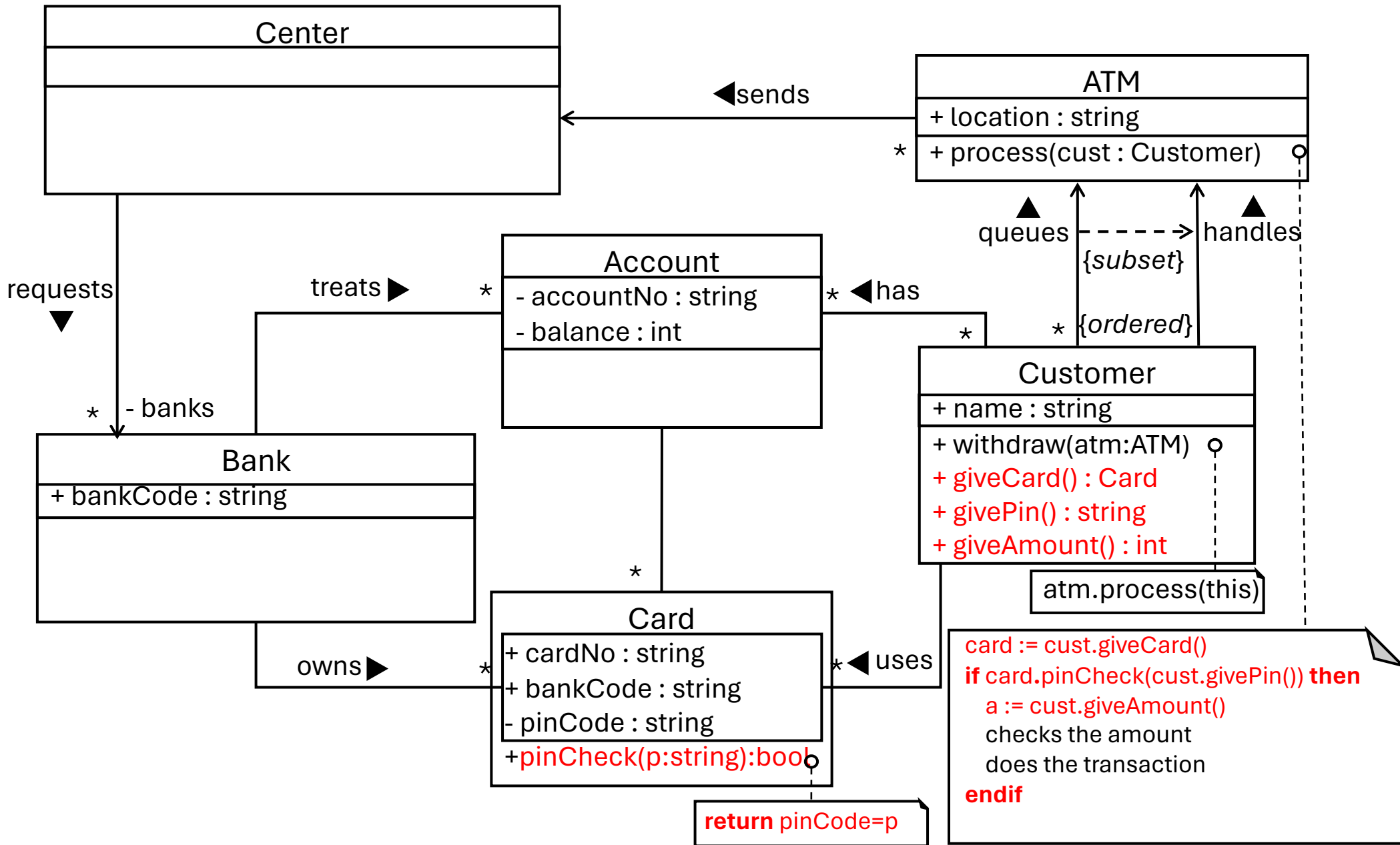
Problem 2

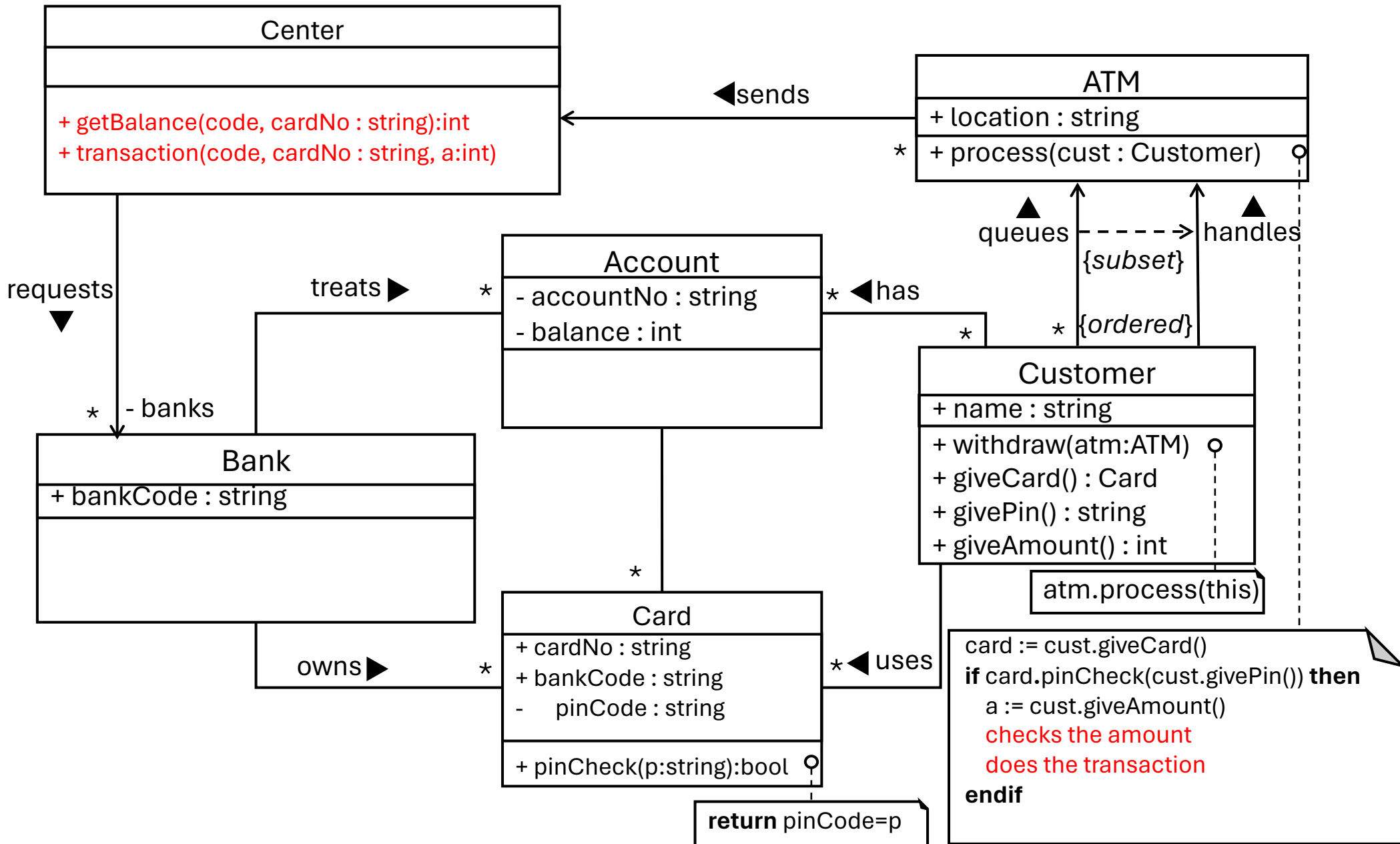
Customers are standing in line at an ATM to withdraw money from their accounts. The clients have debit cards, each of which has a PIN code and an account. The clients can withdraw money one after another by giving their card and code. If the given code is the same as the code of the card, the ATM gives out the money, as long as subtracting the amount from the balance yields a positive number. To check this, the ATM can query the balance of the client from a center by sending the data of the card to the center. The ATM can also send a report about the transaction to the center so the bank can subtract the amount from the account. Simulate the process of withdrawing from the ATM.

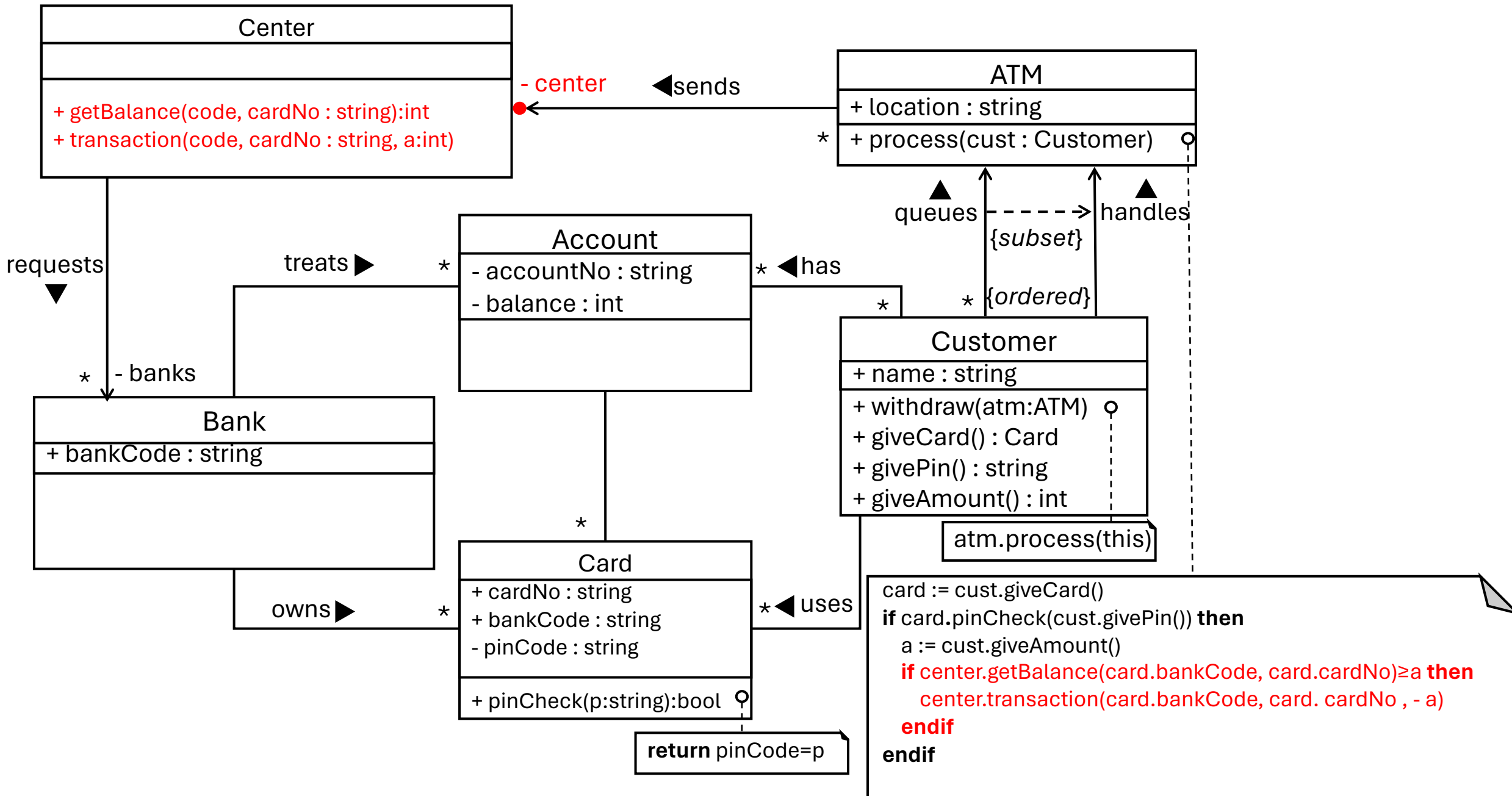


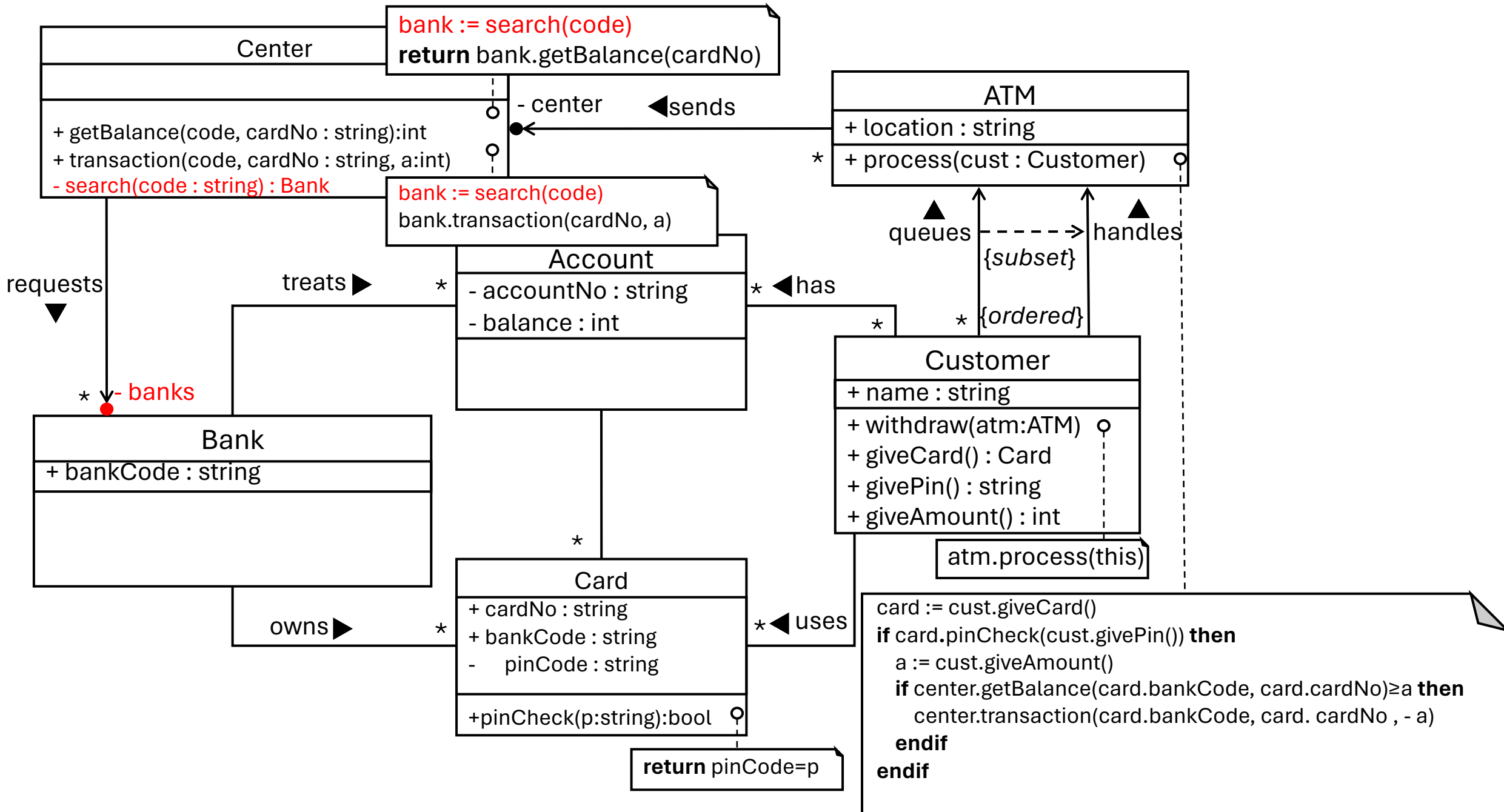


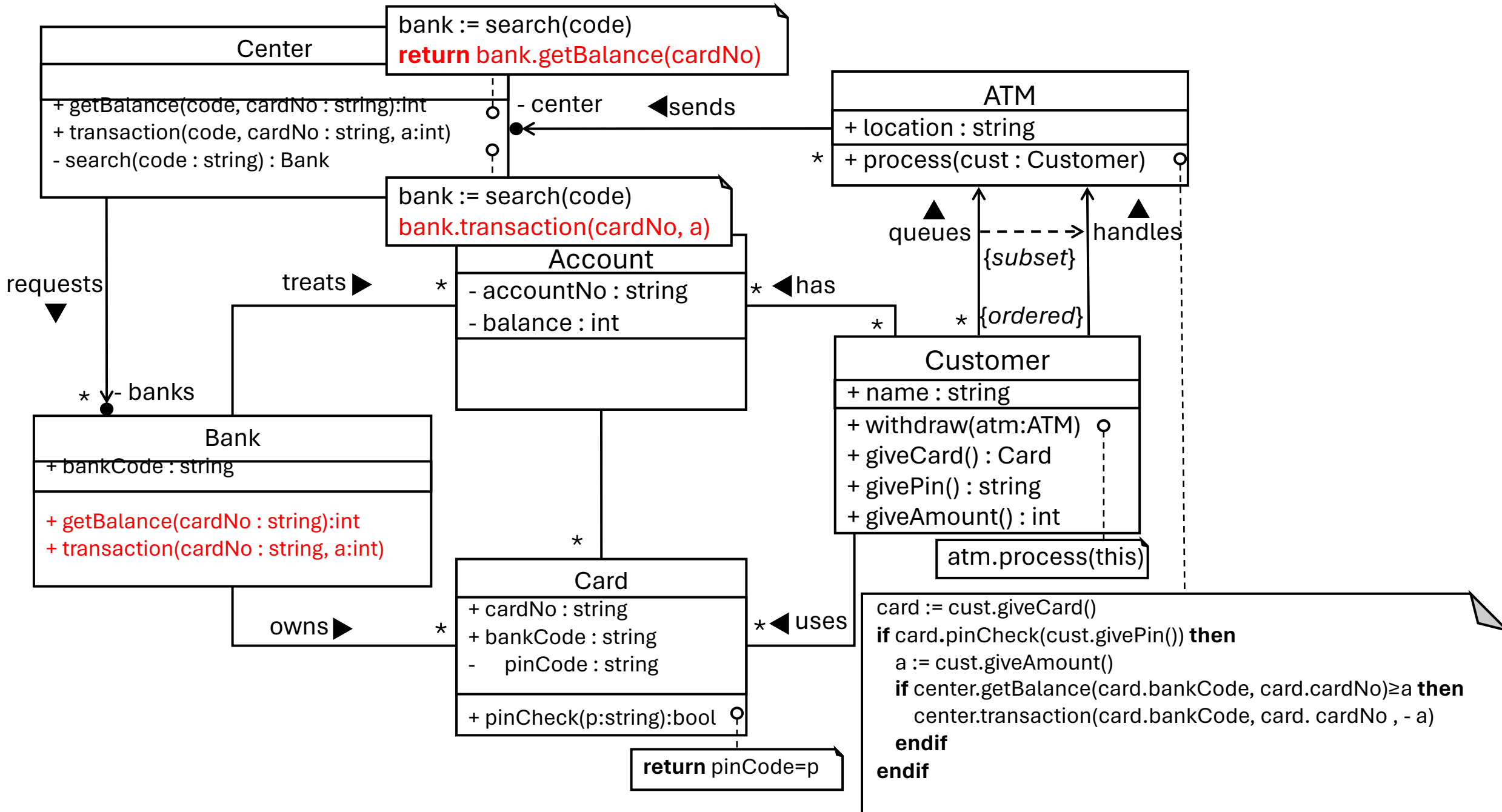


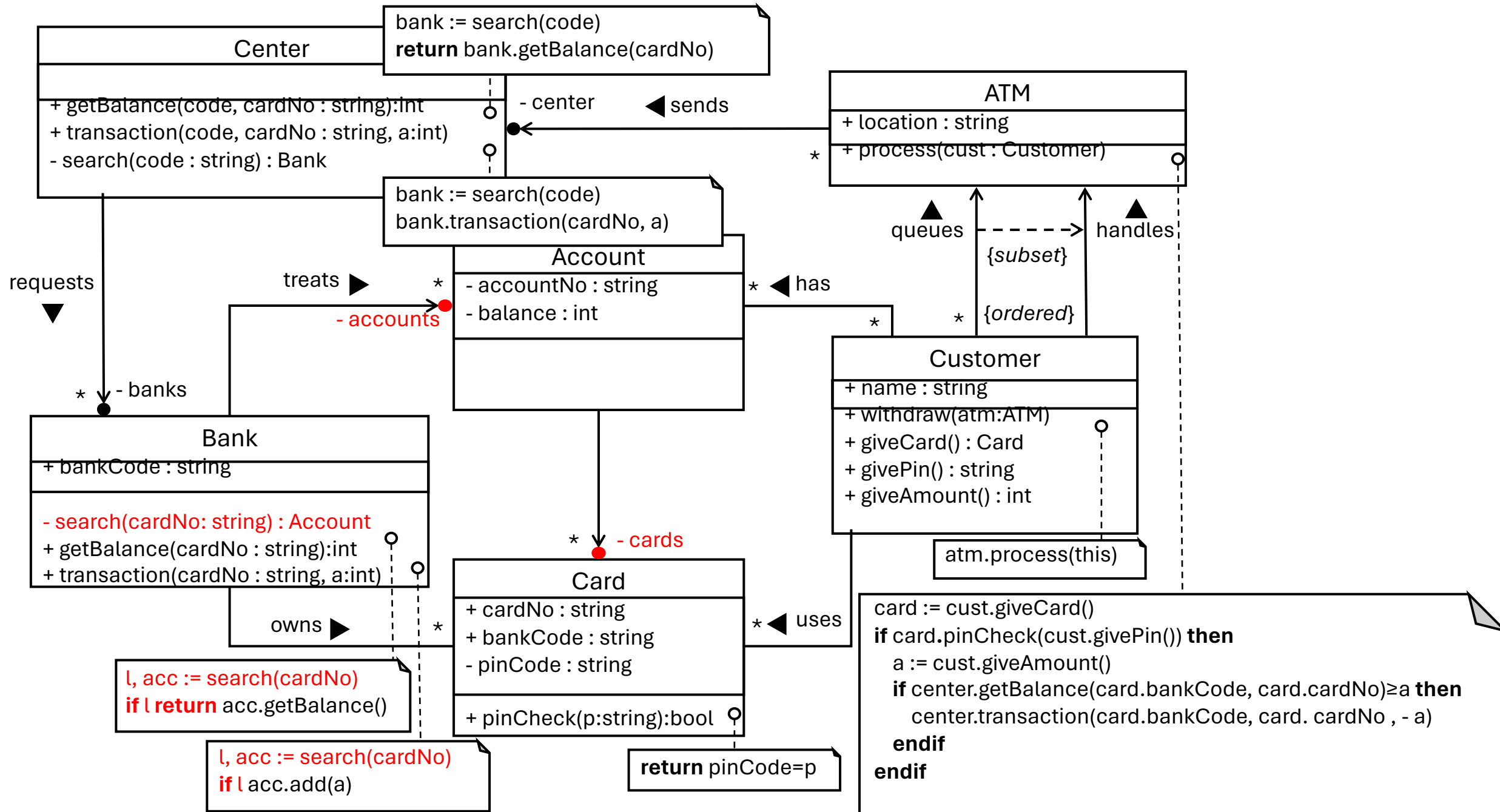


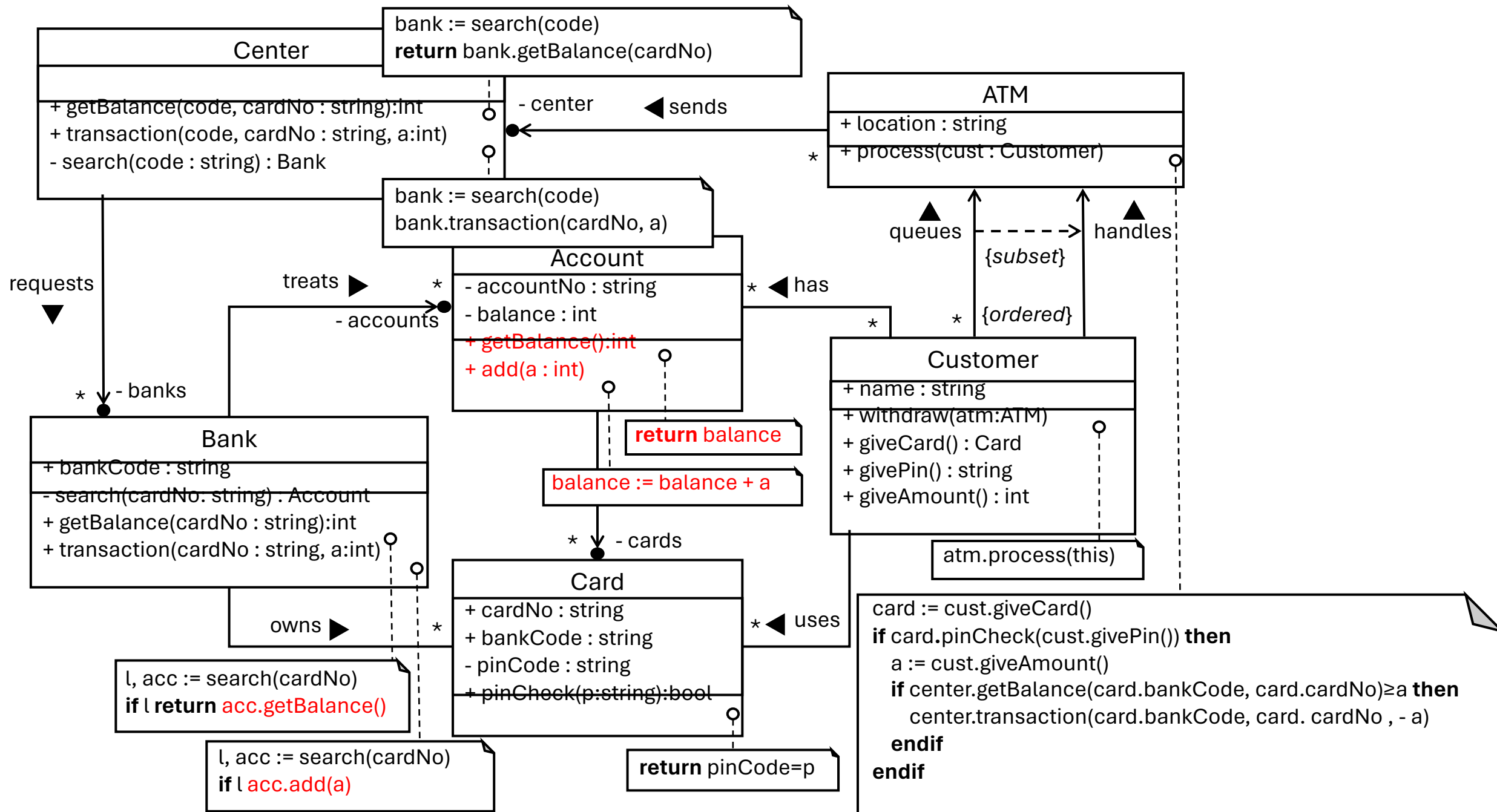


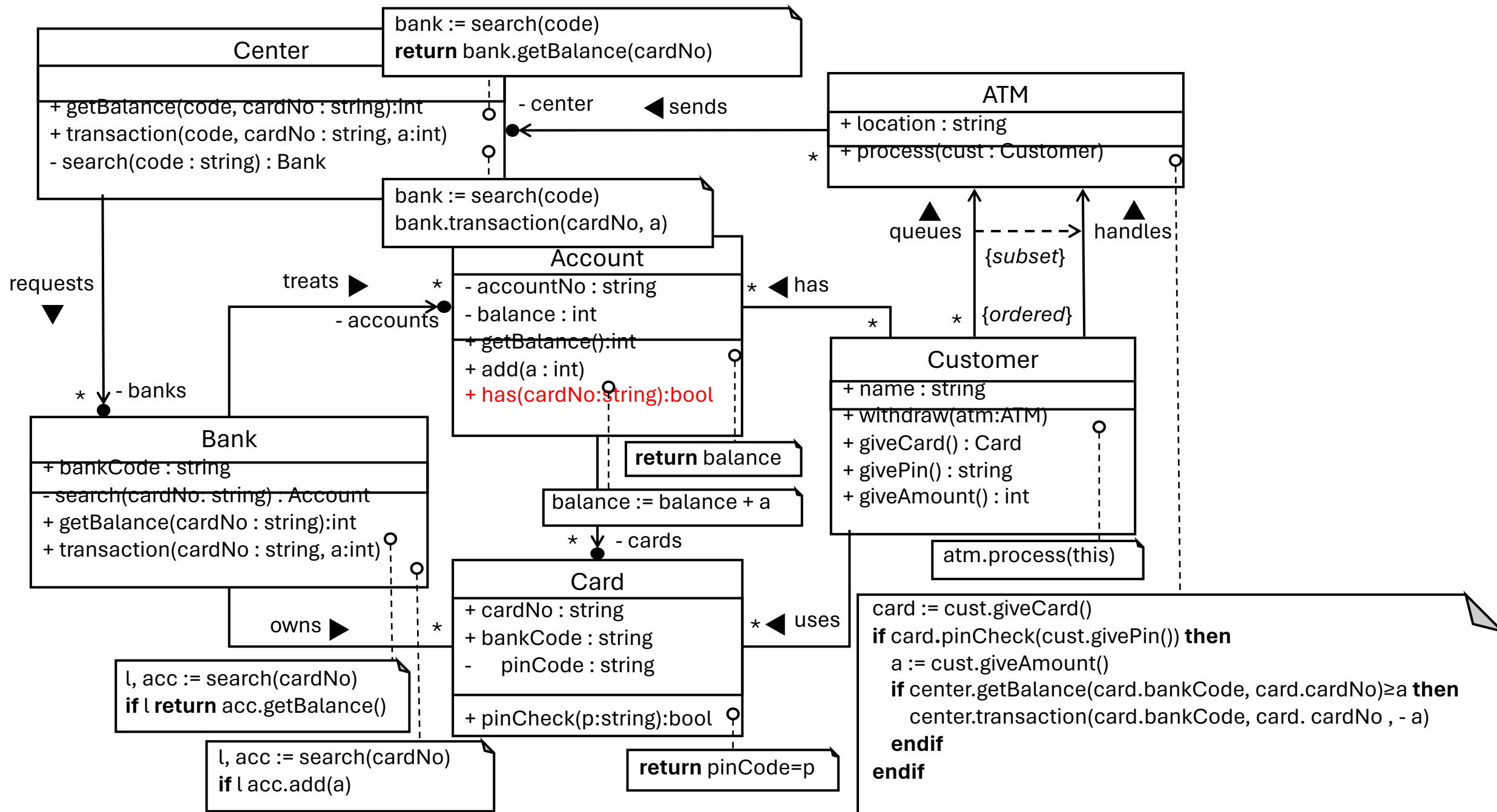


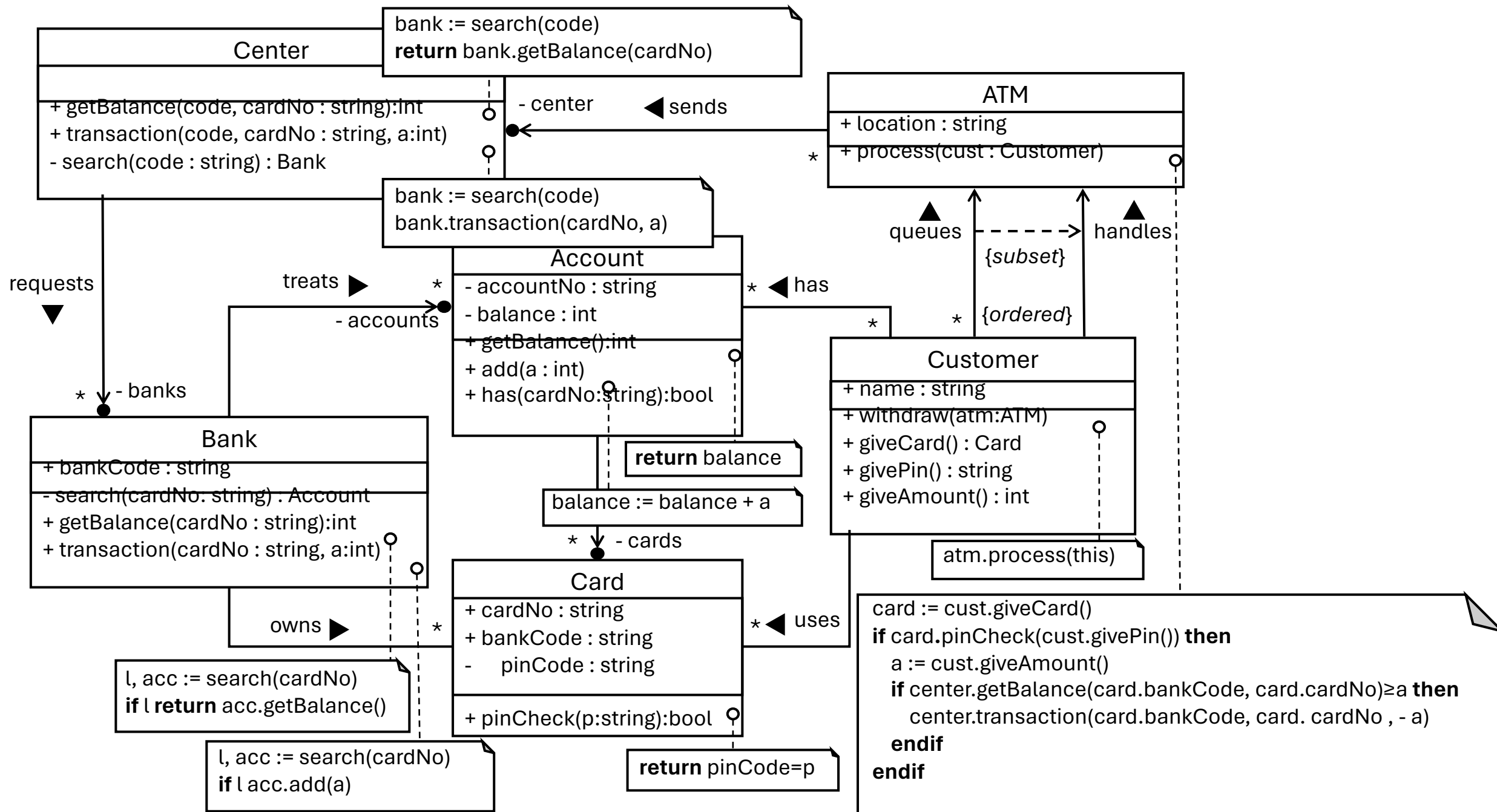




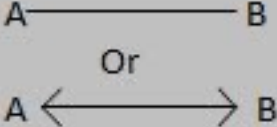













Program Element	Diagram Element	Meaning
Inheritance		B inherits from A
Realization		B implements A
Association		A and B call and access each other's elements
Association(one way)		A can call and access B's element but not vice versa
Aggregation		A has B and B can outlive A
Composition		A has a B and B depends on A