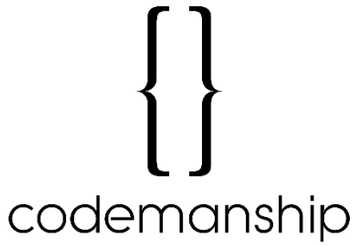


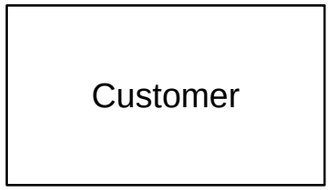
UML

Sequence Diagrams

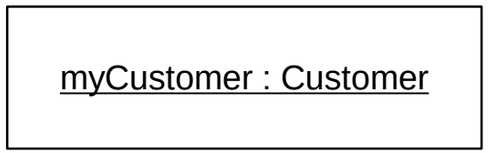
Contents

- Objects in UML
- Timelines
- Messages
  - Synchronous
  - Asynchronous
  - Iteration
- Static method calls in sequence diagrams

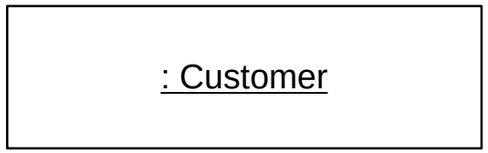
Objects in UML



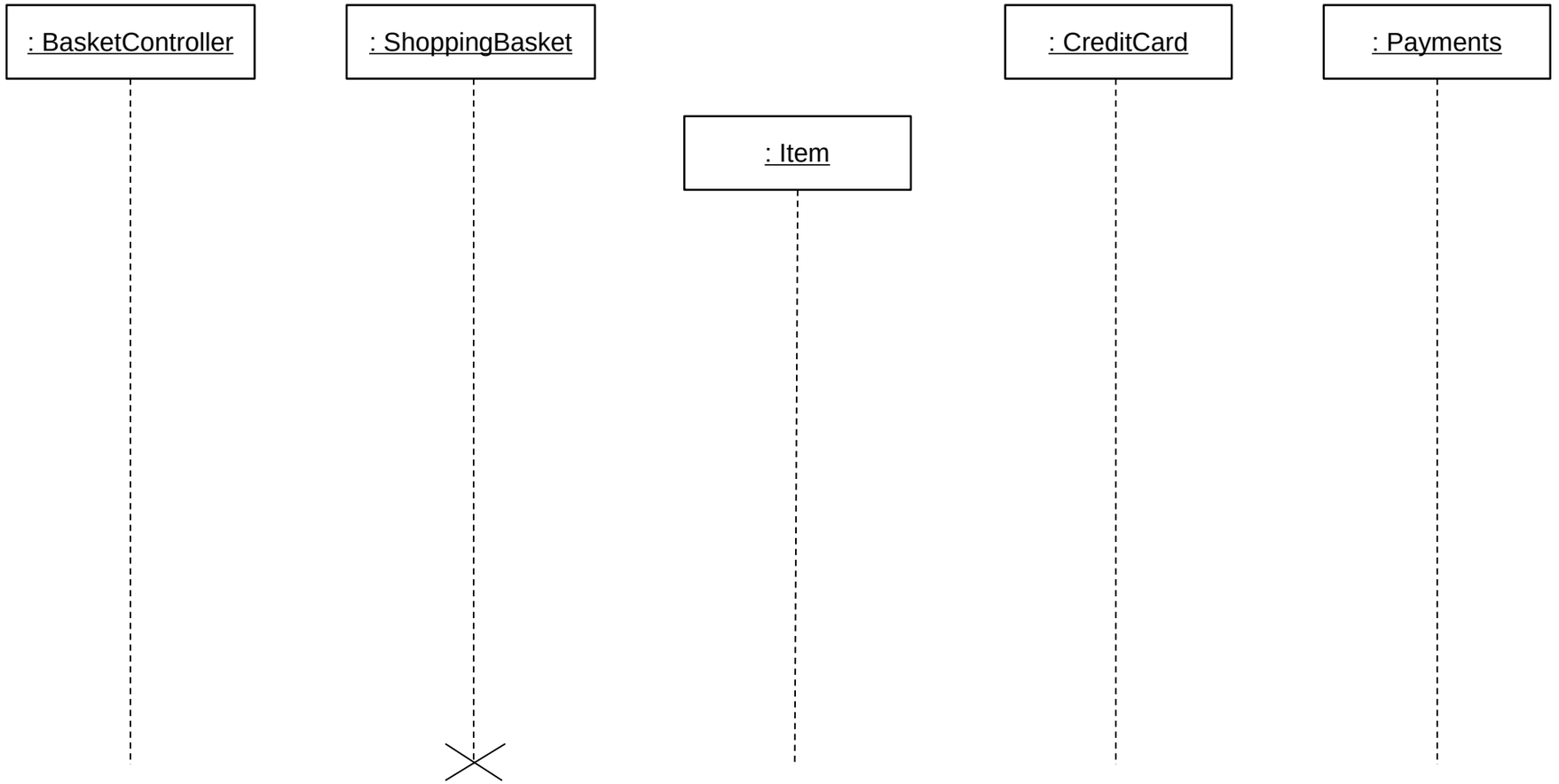

```
public class Customer {

}
```




```
private final Customer myCustomer;
```



Timelines

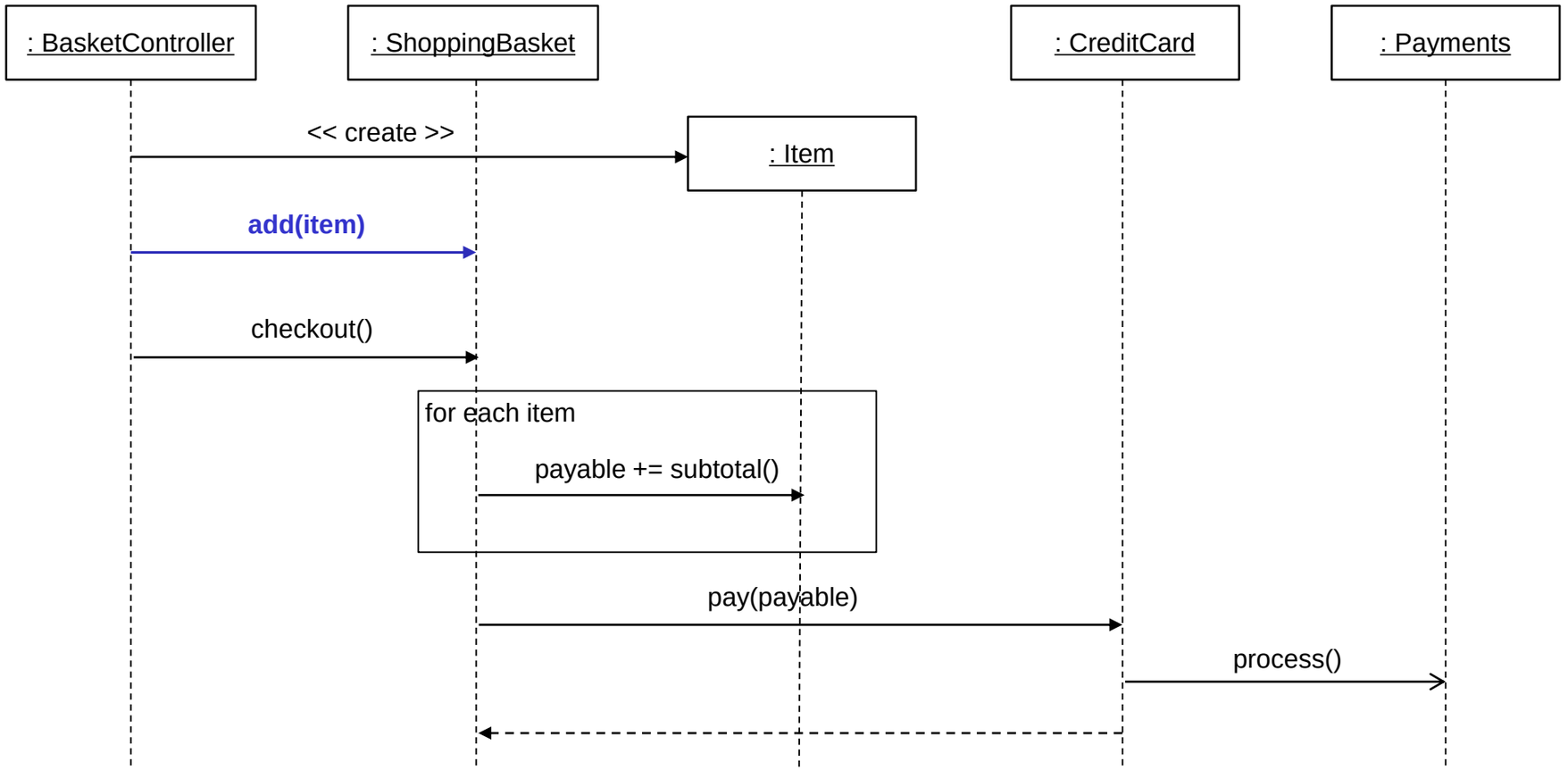

: BasketController
: ShoppingBasket
: Item
: CreditCard
: Payments

```java
public class BasketController {

    private final ShoppingBasket shoppingBasket;
    private final CreditCard card;

    public BasketController(ShoppingBasket shoppingBasket, CreditCard card) {
        this.shoppingBasket = shoppingBasket;
        this.card = card;
    }

    public void addItem(Product product, int quantity){
        shoppingBasket.add(new Item(product, quantity));
    }

    public void checkout(){
        shoppingBasket.checkout(card);
    }
}
```

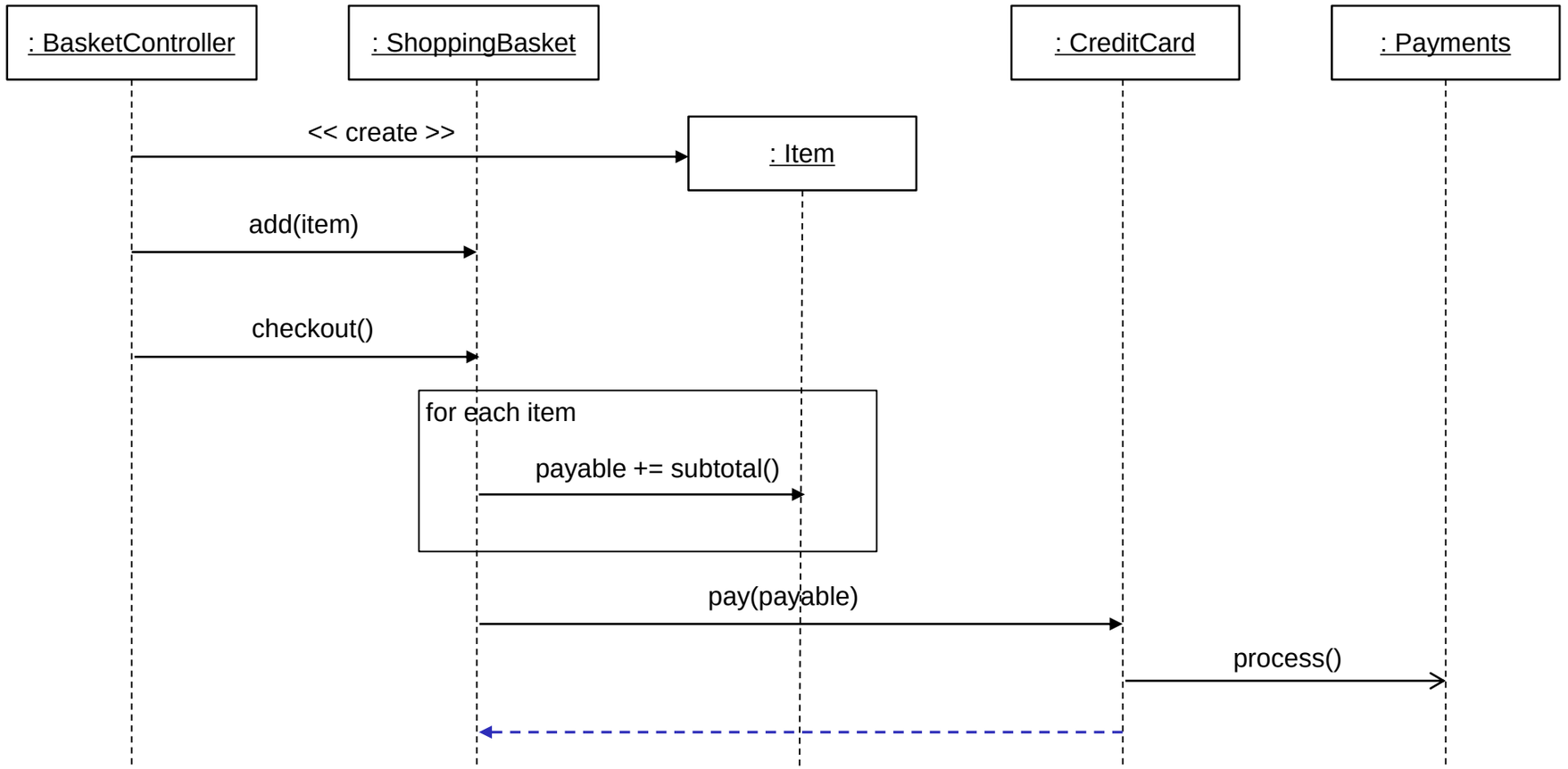
```java
public class ShoppingBasket {

    private List<Item> contents;

    public ShoppingBasket() {
        this.contents = new ArrayList<>();
    }

    public void checkout(CreditCard card) {
        double payable = 0;

        for (Item item: contents) {
            payable += item.subtotal();
        }

        card.pay(payable);
    }

    public void add(Item item){
        contents.add(item);
    }
}
```

```java
public class CreditCard {

    private final Payments payments;
    private final int longNumber;
    private final int expiryMonth;
    private final int expiryYear;
    private final int securityCode;

    public CreditCard(Payments payments, int longNumber, int expiryMonth, int expiryYear, int securityCode) {
        this.payments = payments;
        this.longNumber = longNumber;
        this.expiryMonth = expiryMonth;
        this.expiryYear = expiryYear;
        this.securityCode = securityCode;
    }

    public void pay(double amount){
        payments.process(amount,
                longNumber,
                expiryMonth,
                expiryYear,
                securityCode);
    }
}
```

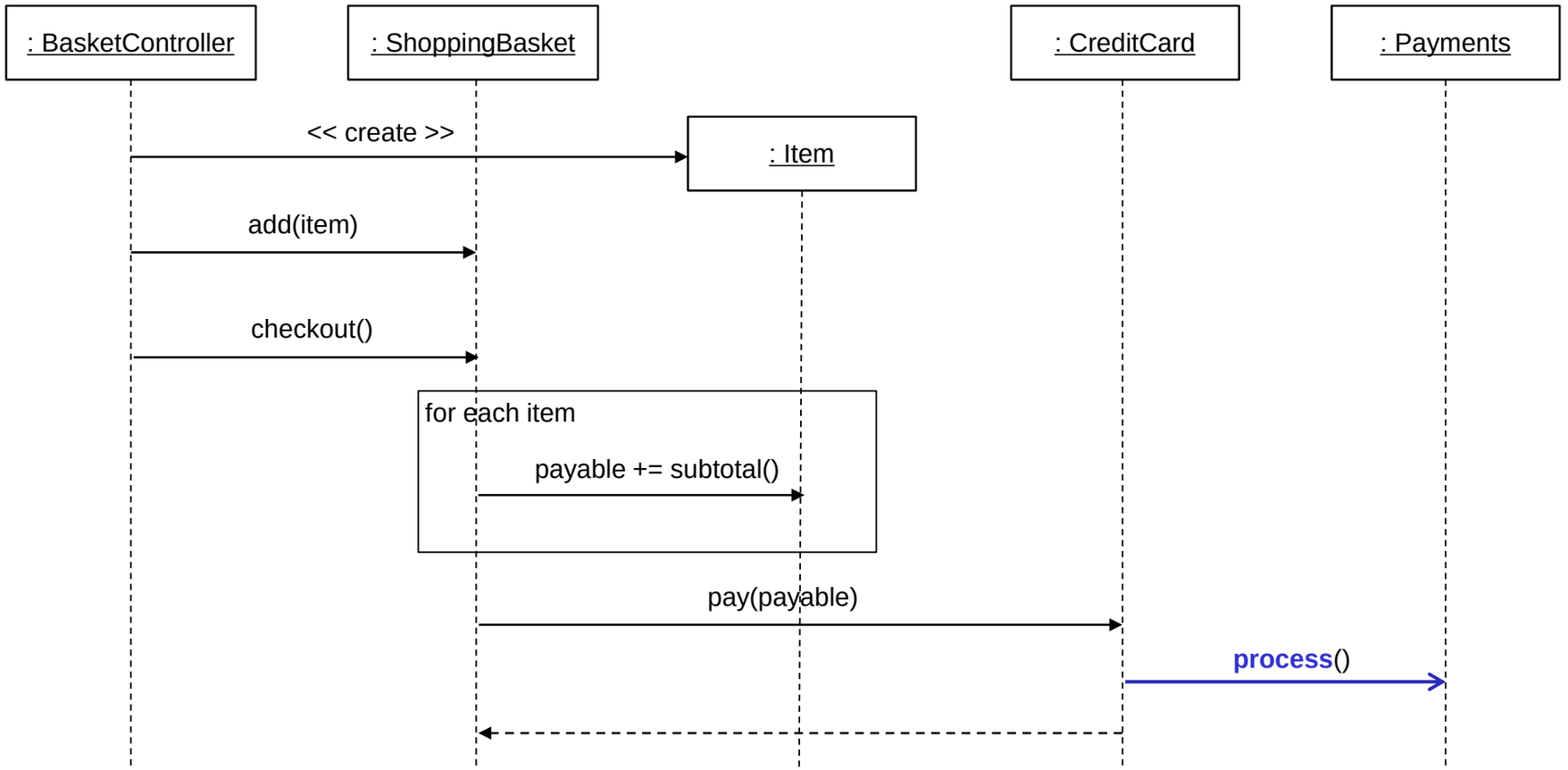
Synchronous Messages



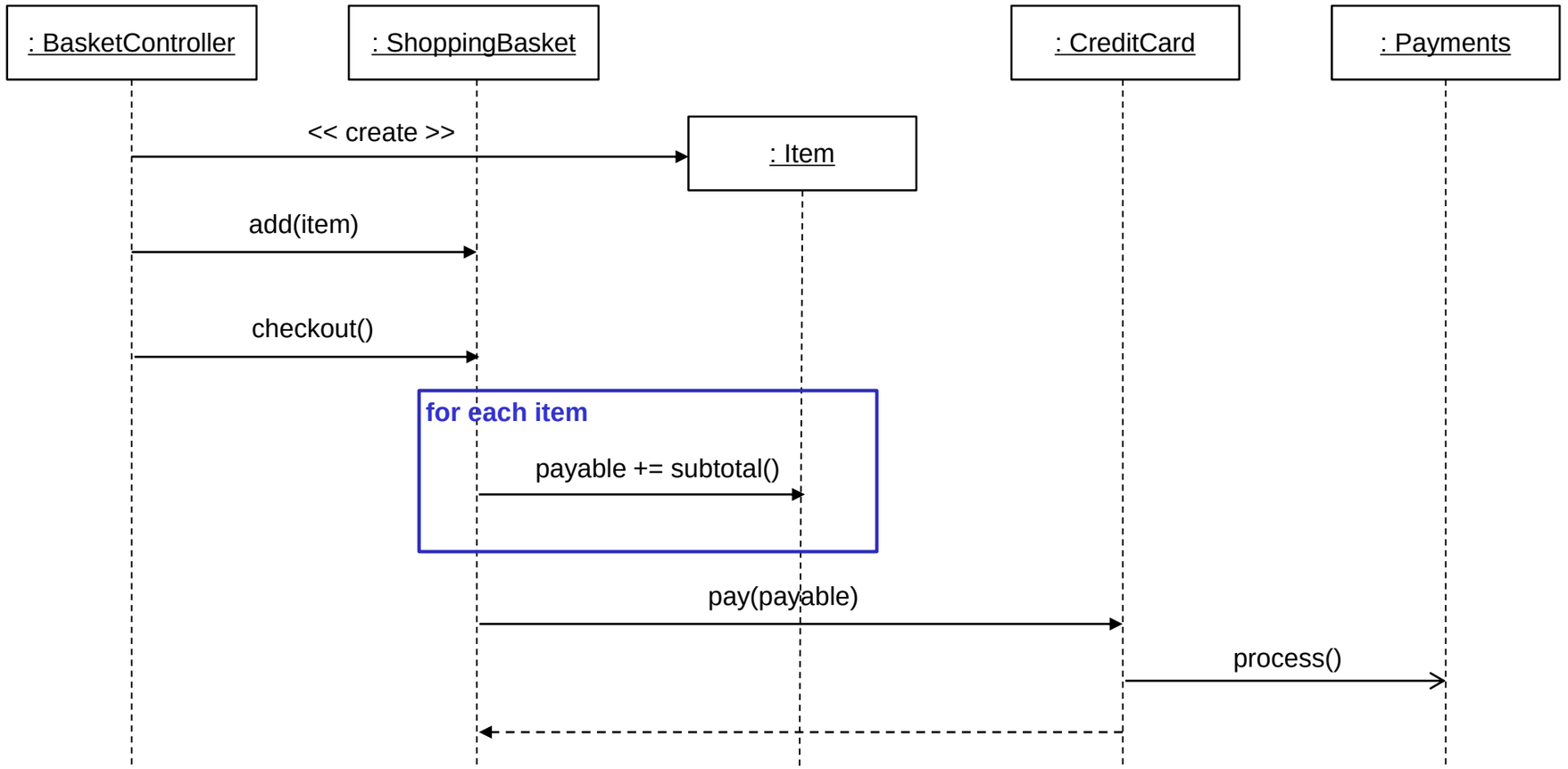
Explicit Returns



Asynchronous Messages



Iteration


: BasketController
: ShoppingBasket
: CreditCard
: Payments
<< create >>
: Item
add(item)
checkout()
for each item
payable += subtotal()
pay(payable)
process()

Calls to Static Methods

```
public void checkout(){
    shoppingBasket.checkout(card);
    Logging.log("BasketController::checkout " + shoppingBasket.toString());
}
```