

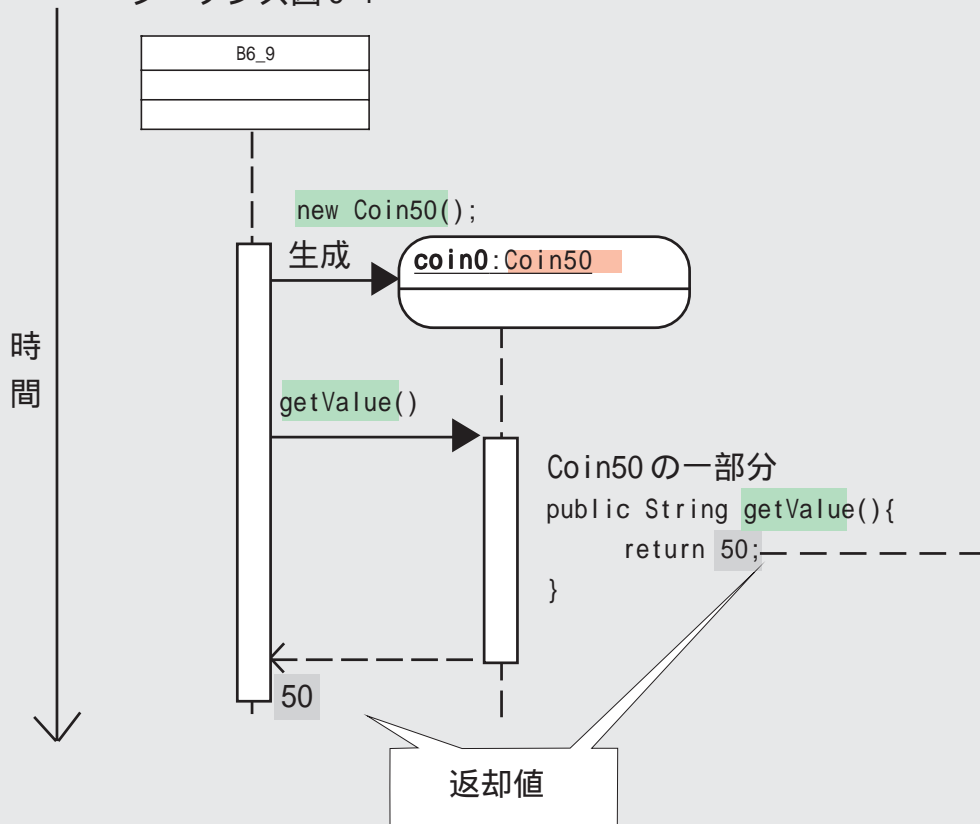
本見開きページの使い方：初めての人はず、右ページの の説明を読みましよう。 のクラス図の解説と用語の説明が記述されています。

E6.1 オブジェクトの生成とメソッドの呼び出し シーケンス図

プログラムB6_9のCoin50オブジェクトの生成とgetValueメソッドの実行

```
public class B6_9{  
    public static void main(String[] args){  
        Coin50 coin0 = new Coin50();  
        System.out.println(coin0.getValue());  
    }  
}
```

シーケンス図 6-1



プログラムの説明

説明	
メソッドの呼び出し	<p>シーケンス図は縦軸が時間軸、横軸が各オブジェクトのメソッドの呼び出しのタイミングを表しています。</p> <p>プログラムB6_9のメインメソッドにおけるオブジェクトの生成とメソッドの呼び出しはシーケンス図6-1で表現できます。</p> <p>破線の矢印は返却時の値とタイミングを示しています。</p> <p>各オブジェクトのメソッドの呼び出しは、シーケンス図にすることで理解しやすくなります。</p> <p>復習になりますが、メソッドにメッセージを伝えるには</p> <p>変数名.メッセージ（例：<code>coin0.getValue()</code>）と記述する必要がありました。</p> <p>シーケンス図を見てください。この記述は必ず生成されたオブジェクトへのメッセージになります。</p>

本見開きページの使い方：初めての人はず、右ページの の説明を読みましよう。 のプログラムの解説と用語の説明が記述されています。

E6.2 Wallet クラス プログラム

プログラム B6_10

```
public class B6_10{  
    public static void main(String[] args){  
        Wallet wallet = new Wallet();  
        wallet.add(new Coin50());  
        wallet.add(new Coin50());  
        System.out.println(wallet.size());  
    }  
}
```

Wallet
<code>arrayList:ArrayList<Value> = new ArrayList<Value>()</code>
<code>Wallet() add(value:Value):void remove(number:int):Value size():int getSum():int</code>

実行結果

B6_10 の実行結果

```
> java B6_10  
2
```

課題の説明

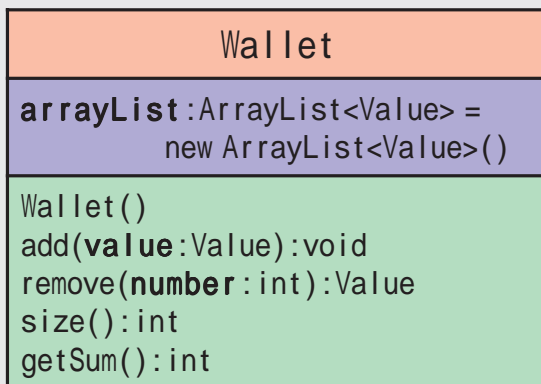
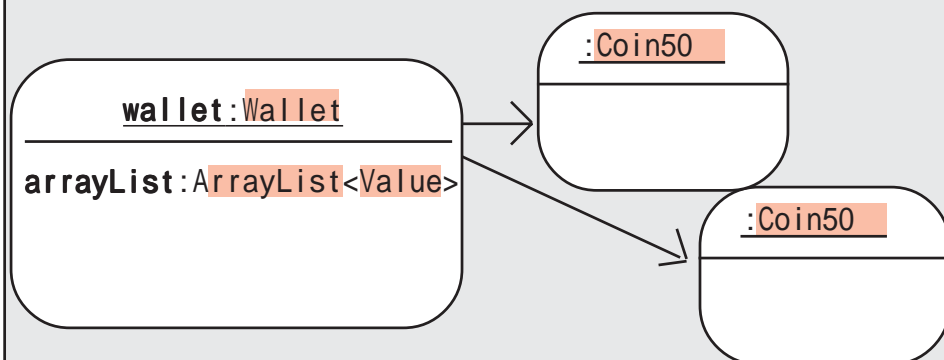
説明	
Walletクラス	<p>Valueインタフェースを実装した貨幣 (Coin) オブジェクトを入れるWalletオブジェクトのクラス図があります。</p> <p>それぞれの振る舞いのAPIは</p> <ul style="list-style-type: none">• add(Value):void 引数のValueインタフェースを実装したオブジェクトを追加します。• remove(int):Value Valueインタフェースを実装したオブジェクトを取り出します。ただし、Walletオブジェクト (財布) からは無くなります。引数の整数の値はオブジェクトの価値です。Coin50オブジェクトを取り出す場合は50が引数になります。• size():int Valueオブジェクトの数を返却します。• getSum():int Valueオブジェクトの価値の合計を返却します。 <p>となります。</p>

本見開きページの使い方：初めての方はまず、右ページの の説明を読みましょう。 のプログラムの解説と用語の説明が記述されています。

E6.3 演習 E6-1

プログラム

プログラム B6_11 の の時点のオブジェクト図



課題の説明

説明	
今日の課題	<p>オブジェクト図をみてください。</p> <p>Valueインタフェースを実装した貨幣 (Coin) オブジェクトを入れるWalletオブジェクトのクラス図があります。</p> <p>基本問題: Walletクラス図を参考にクラスのソースコードを機械的に導出してください。</p> <p>応用問題: Walletクラスを完成させてください。</p> <ul style="list-style-type: none">• add(Value):void 引数のValueインタフェースを実装したオブジェクトを追加します。• remove(int):Value Valueインタフェースを実装したオブジェクトを取り出します。ただし、Walletオブジェクト (財布) からは無くなります。引数の整数の値はオブジェクトの価値です。Coin50オブジェクトを取り出す場合は50が引数になります。• size():int Valueオブジェクトの数を返却します。• getSum():int Valueオブジェクトの価値の合計を返却します。
<p>プログラム B6_11</p> <pre>public class B6_11{ public static void main(String[] args){ Wallet wallet = new Wallet(); wallet.add(new Coin50()); wallet.add(new Coin50()); System.out.println(wallet.getSum()); } }</pre>	

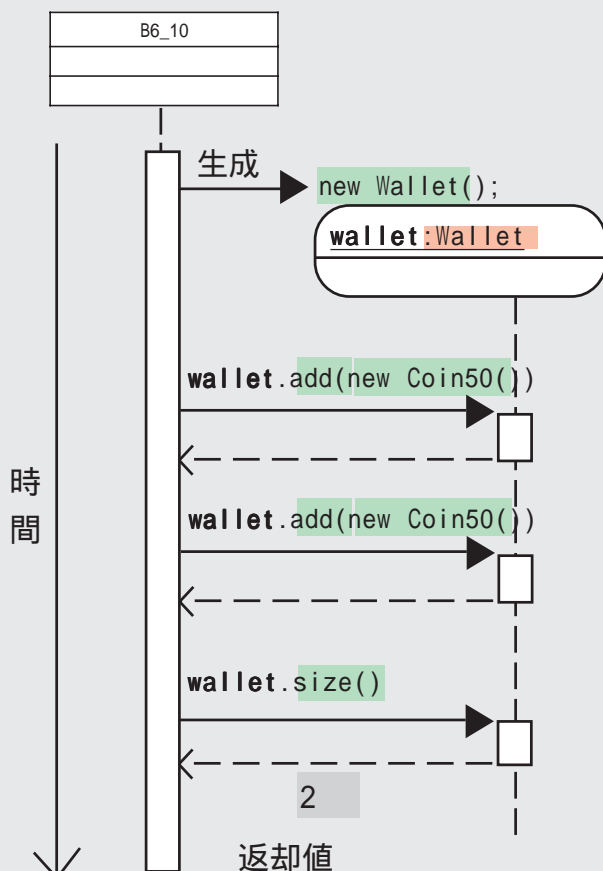
本見開きページの使い方：初めての人はず、右ページの の説明を読みましよう。 のクラス図の解説と用語の説明が記述されています。

E6.4 オブジェクトの生成とメソッドの呼び出し

シーケンス図

```
public class B6_10{  
    public static void main(String[] args){  
        Wallet wallet = new Wallet();  
        wallet.add(new Coin50());  
        wallet.add(new Coin50());  
        System.out.println(wallet.size());  
    }  
}
```

シーケンス図 6-2

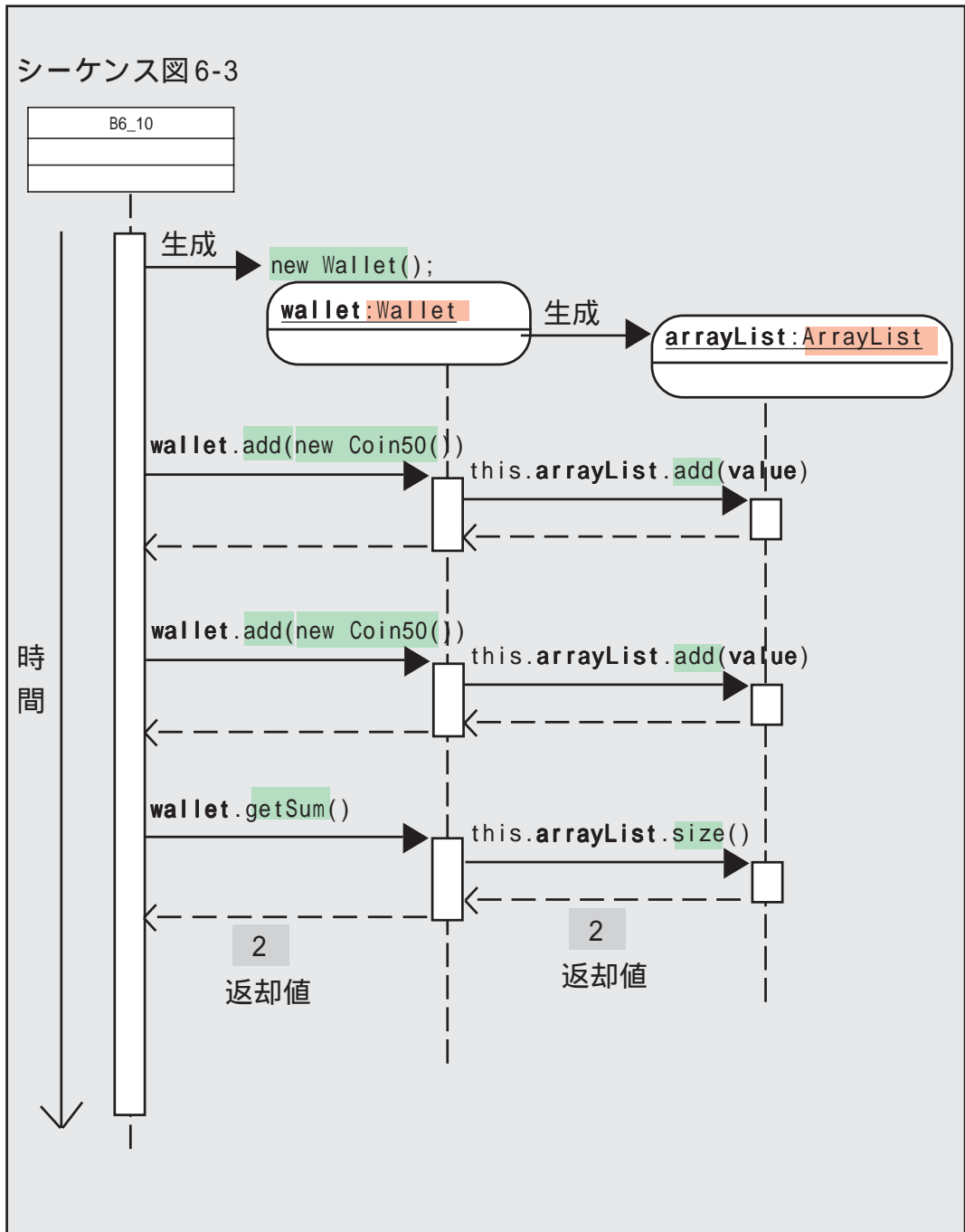


キーワード：シーケンス図

本見開きページの使い方：初めての人はず、右ページの の説明を読みましょう。 のクラス図の解説と用語の説明が記述されています。

E6.5 オブジェクトの生成とメソッドの呼び出し

シーケンス図



プログラム Wallet

```
import java.util.ArrayList;
public class Wallet{
    private ArrayList<Value> arrayList = new ArrayList<Value>();
    public Wallet(){
    }
    public void add(Value value){
        this.arrayList.add(value);
    }
    public Value remove(int number){
        for(int i = 0; i < this.arrayList.size(); i++){
            Value value = this.arrayList.get(i);
            if(value.getValue() == number){
                return this.arrayList.remove(i);
            }
        }
        return null;
    }
    public int size(){
        return this.arrayList.size();
    }
    public int getSum(){
        int sum = 0;
        for(int i = 0; i < this.arrayList.size(); i++){
            Value value = this.arrayList.get(i);
            sum = sum + value.getValue();
        }
        return sum;
    }
}
```

本見開きページの使い方：初めての人はず、右ページの の説明を読みましょう。 のプログラムとクラス図の説明が記述されています。

E6.6 Bookshelf

プログラムとクラス図

プログラム Bookshelf

```
import java.util.ArrayList;

public class Bookshelf{
    private ArrayList<Book> arrayList
        = new ArrayList<Book>();
    public Bookshelf(){
    }
    public void add(Book book){
    }
    public Book get(String isbn){
        for(int i = 0; i < this.arrayList.size(); i++){
            Book book = this.arrayList.get(i);
            if(book.getIsbn().equals(isbn)){
                return book;
            }
        }
        return null;
    }
    public int size(){
        return 0;
    }
}
```

Bookshelf

```
arrayList:ArrayList<Book> =
    new ArrayList<Book>()
```

```
Bookshelf()
add(book:Book):void
get(isbn:String):Book
size():int
```

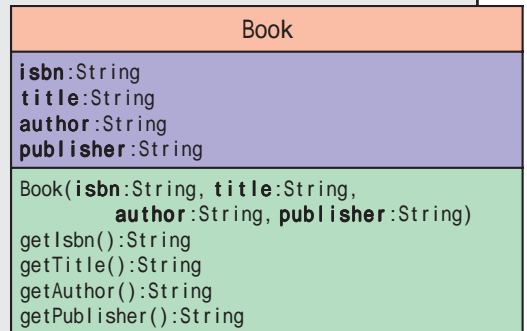
API仕様 Bookshelf

Bookshelf	唯一のコンストラクタです。
add	BookshelfオブジェクトにBookオブジェクトを入れます。
get	Bookshelfオブジェクトに入っているBookオブジェクトを返します。ただし、引数はString型のisbnの値になります。
size	Bookshelfオブジェクトに管理されているBookオブジェクトの個数を返します。

プログラム Book

```
public class Book{
    private String isbn;
    private String title;
    private String author;
    private String publisher;
    public Book(String isbn, String title,
                String author, String publisher){
        this.isbn = isbn;
        this.title = title;
        this.author = author;
        this.publisher = publisher;
    }
    public String getIsbn(){
        return this.isbn;
    }
    public String getTitle(){
        return this.title;
    }
    public String getAuthor(){
        return this.author;
    }
    public String getPublisher(){
        return this.publisher;
    }
}

public class B6_12{
    public static void main(String[] args){
        Bookshelf shelf = new Bookshelf();
        shelf.add(new Book("489471258X", "Java 鉄則 ",
                           "ピアソン", "ドキュメント"));
        shelf.add(new Book("4798105783", "UML 辞書 ",
                           "テクノ", "翔泳社"));
        Book book = shelf.get("4798105783");
        System.out.println(book.getTitle());
    }
}
```



実行結果

```
> java B6_12
```

```
UML 辞書
```