

図書館の貸出问题

佐原伸

2005年11月22日

1 問題

以下のトランザクションが発生する図書館の貸出管理処理の VDM++ モデルである。

1. 本の貸出および返却
2. 図書館への本の登録および削除
3. 特定の作者の本、あるいは特定の分野の本に関するリストの作成
4. 特定の利用者が借りている本のリストの作成
5. 特定の本を最後に借りた人の検索

利用者は図書館の職員と一般の利用者で、職員は（１）～（５）のすべてを行えるが、一般の利用者は（４）の機能のうち、自分で借りている本のリストの作成しか行えない。

2 制約条件

1. 図書館のすべての本は、貸出可能かすでに貸し出されているかのどちらかである。
2. ある本が貸出可能であり、かつ貸し出されているということはない。
3. 利用者は一回にある決まった冊数以上の本を借り出すことはできない。

2.1 利用者クラス

システムの利用者を表す抽象クラス。

class 利用者クラス is subclass of 人クラス

values

- = "\$Header: /home/vdmtools/cvsroot/toolbox/examples-jp/pp/library/AbstractUser.vpp, v

20:16 vdmtools Exp \$"

instance variables

public *ID* : char*;

public 図書館 : 図書館クラス;

貸出集合 : 貸出クラス-set := {};

operations

public

借りる : *Date* × 本実体クラス × 職員クラス \xrightarrow{o} 貸出クラス

借りる (*today*, 借用本, 操作者) \triangle

(dcl ある貸出 : 貸出クラス := 操作者.貸出を作成する (*today*, self, 借用本
);

貸出集合に追加する(ある貸出);

return ある貸出

)

pre 借用本.貸出可能 () \wedge

self.借用可能 (*today*)

post

let ある貸出 = 操作者.貸出を作成する (*today*, self, 借用本) in

借用本.貸出可能 () = false \wedge

ある貸出.借用者を得る () = self;

public

```

返す :  $Date \times \text{本実体クラス} \xrightarrow{o} ()$ 
返す (返却日, 借用本)  $\triangleq$ 
    let {貸出} = {貸出 | 貸出  $\in$  貸出集合} .
        貸出.本実体 = 借用本  $\wedge$  貸出.返却日 = nil } in
    貸出.貸出を終わる(返却日, 借用本)
pre 借用本.貸出可能() = false  $\wedge$ 
    借用本  $\in$  {貸出.本実体 | 貸出  $\in$  貸出集合}

post
self.貸出集合 =  $\overleftarrow{\text{貸出集合}} \setminus$ 
    {貸出 | 貸出  $\in$  貸出集合} .
        貸出.本実体 = 借用本  $\wedge$  貸出.返却日 = nil }  $\wedge$ 
    借用本.貸出可能() = true;

public
借用可能 :  $Date \xrightarrow{o} \mathbb{B}$ 
借用可能 (貸出日)  $\triangleq$ 
    if 貸出集合 = {}
    then return true
    else return  $\forall$  貸出  $\in$  貸出集合 .
        貸出.返却日 = nil  $\vee$ 
        (貸出.返却日  $\neq$  nil  $\wedge$ 
        貸出日.  $\leq$  (貸出.返却日));

public
借りている本 : 利用者クラス  $\xrightarrow{o}$  本クラス-set
借りている本 (操作員)  $\triangleq$ 
    return {貸出.本実体.本 | 貸出  $\in$  self.貸出集合  $\cdot$  貸出.返却日 = nil }
pre 操作員.職員()  $\vee$  self = 操作員 ;

public
借用数 : ()  $\xrightarrow{o} \mathbb{N}$ 
借用数 ()  $\triangleq$ 
    return card self.借りている本実体 (self);

public

```

```

借りている本実体 : 利用者クラス  $\xrightarrow{o}$  本クラス-set
借りている本実体 (操作員)  $\triangle$ 
    return {貸出.本実体 | 貸出  $\in$  self.貸出集合  $\cdot$  貸出.返却日 = nil }
pre 操作員.職員 ()  $\vee$  self = 操作員 ;

public
    職員 : ()  $\xrightarrow{o}$   $\mathbb{B}$ 
    職員 ()  $\triangle$ 
        is subclass responsibility;

public
    ID を設定する : char*  $\xrightarrow{o}$  ()
    ID を設定する (ある ID)  $\triangle$ 
        ID := ある ID;

public
    図書館を設定する : 図書館クラス  $\xrightarrow{o}$  ()
    図書館を設定する (ある図書館)  $\triangle$ 
        図書館 := ある図書館;

public
    貸出集合に追加する : 貸出クラス  $\xrightarrow{o}$  ()
    貸出集合に追加する (ある貸出)  $\triangle$ 
        貸出集合 := 貸出集合  $\cup$  {ある貸出}
end 利用者クラス

Test Suite :      vdm.tc
Class :          利用者クラス

```

Name	#Calls	Coverage
利用者クラス ‘職員	0	0%
利用者クラス ‘返す	6	98%
利用者クラス ‘借りる	35	✓
利用者クラス ‘借用数	18	✓
利用者クラス ‘借用可能	35	✓
利用者クラス ‘借りている本	12	✓
利用者クラス ‘ID を設定する	32	✓

Name	#Calls	Coverage
利用者クラス‘借りている本実体	18	✓
利用者クラス‘図書館を設定する	32	✓
利用者クラス‘貸出集合に追加する	35	✓
Total Coverage		98%

2.2 AllT

全テストケースのテスト

```
class AllT
```

```
operations
```

```
public
```

```
  run : ()  $\xrightarrow{o}$   $\mathbb{B}^*$ 
```

```
  run ()  $\triangleq$ 
```

```
    return [
```

```
      new 図書館貸出管理クラス T ().run ()]
```

```
end AllT
```

2.3 著者クラス

本の著者の情報を持つ

```
class 著者クラス is subclass of 人クラス
```

```
values
```

```
    - = "$Header: /home/vdmtools/cvsroot/toolbox/examples-jp/pp/library/Author.vpp, v 1.1 2000/01/16 vdmtools Exp $"
```

```
20:16 vdmtools Exp $"
```

```
instance variables
```

```
    public 著書:[本クラス]:= nil ;
```

```
end 著者クラス
```

2.4 本クラス

題名・著者が同じ本を一つと考える場合の、抽象的論理的概念としての「本」。題名・著者を答え、本の実体クラスのインスタンスを生成・管理している主体。

class 本クラス

values

- = "\$Header: /home/vdmttools/cvsroot/toolbox/examples-jp/pp/library/Book.vpp, v 1.1 200

20:16 vdmttools Exp \$"

instance variables

public 題名 : char*;

public 著者 : 著者クラス;

public 分野集合 : 分野クラス-set;

operations

public

初期化する : $\text{char}^* \times \text{著者クラス} \times \text{分野クラス-set} \xrightarrow{o} ()$

初期化する (ある題名, ある著者, ある分野集合) \triangle

(題名 := ある題名;

著者 := ある著者;

分野集合 := ある分野集合

)

end 本クラス

2.5 本実体クラス

個々の本の情報を持つ物理的な概念としての「本」。蔵書管理の対象。

class 本実体クラス is subclass of 本クラス

values

- = "\$Header: /home/vdmtools/cvsroot/toolbox/examples-jp/pp/library/BookInstance.vpp, "

20:16 vdmtools Exp \$"

instance variables

貸出中 : \mathbb{B} := false;

public ID : char*;

public 本 : 本クラス;

貸出 : [貸出クラス] := nil ;

operations

public

初期化する : $\text{char}^* \times \text{本クラス} \xrightarrow{o} ()$

初期化する (ある ID, ある本) \triangle

(ID := ある ID;

本 := ある本

);

public

貸し出す : 貸出クラス \xrightarrow{o} 本実体クラス

貸し出す (ある貸出) \triangle

(self.貸出不可にする();

貸出 := ある貸出;

return self

);

public

```

貸出を終わる : ()  $\xrightarrow{o}$  ()
貸出を終わる ()  $\triangleq$ 
  ( 貸出可能にする();
    貸出 := nil
  );

public
  貸出可能 : ()  $\xrightarrow{o}$   $\mathbb{B}$ 
  貸出可能 ()  $\triangleq$ 
    return  $\neg$  貸出中;

public
  借用者を得る : ()  $\xrightarrow{o}$  [利用者クラス]
  借用者を得る ()  $\triangleq$ 
    if 貸出 = nil
    then return nil
    else return 貸出.借用者を得る ();

public
  貸出可能にする : ()  $\xrightarrow{o}$  ()
  貸出可能にする ()  $\triangleq$ 
    貸出中 := false;

public
  貸出不可にする : ()  $\xrightarrow{o}$  ()
  貸出不可にする ()  $\triangleq$ 
    貸出中 := true

end 本実体クラス

Test Suite :    vdm.tc
Class :        本実体クラス

```

Name	#Calls	Coverage
本実体クラス ‘貸し出す	70	✓
本実体クラス ‘貸出可能	82	✓
本実体クラス ‘初期化する	48	✓
本実体クラス ‘借用者を得る	15	✓

Name	#Calls	Coverage
本実体クラス ‘貸出を終わる	6	✓
本実体クラス ‘貸出不可にする	70	✓
本実体クラス ‘貸出可能にする	6	✓
Total Coverage		100%

2.6 貸出クラス

貸出行為を処理・管理する主体。

class 貸出クラス

values

- = "\$Header: /home/vdmtools/cvsroot/toolbox/examples-jp/pp/library/LentOut.vpp, v 1.1

20:16 vdmtools Exp \$"

instance variables

public 借用者 : 利用者クラス;

public 本実体 : 本実体クラス;

public 貸出日 : *Date*;

public 返却日 : [*Date*] := nil ;

inv if 返却日 \neq nil

then self.貸出日. \leq (返却日)

else true

operations

public

初期化する : $Date \times \text{利用者クラス} \times \text{本実体クラス} \xrightarrow{o} ()$

初期化する (ある貸出日, ある借用者, ある本実体) \triangle

(借用者 := ある借用者;

本実体 := ある本実体.貸し出す (self);

貸出日 := ある貸出日

);

public

貸出を終わる : $Date \times \text{本実体クラス} \xrightarrow{o} ()$

貸出を終わる (返却した日, 借用本) \triangle

(借用本.貸出を終わる();

返却日 := 返却した日

)

pre 返却日 = nil ;

public

```

借用者を得る : ()  $\xrightarrow{o}$  [利用者クラス]
借用者を得る ()  $\triangleq$ 
  if 返却日  $\neq$  nil
  then return nil
  else return 借用者
end 貸出クラス
Test Suite :      vdm.tc
Class :         貸出クラス

```

Name	#Calls	Coverage
貸出クラス ‘初期化する	70	✓
貸出クラス ‘借用者を得る	50	✓
貸出クラス ‘貸出を終わる	6	✓
Total Coverage		100%

2.7 職員クラス

貸出・返却を行う主体である。司書クラスは、貸出クラスのインスタンスの生成・管理を行う。

class 職員クラス is subclass of 利用者クラス

values

- = "\$Header: /home/vdmtools/cvsroot/toolbox/examples-jp/pp/library/Libarian.vpp, v 1.1

20:16 vdmtools Exp \$"

operations

public

貸出を作成する : $Date \times \text{利用者クラス} \times \text{本実体クラス} \xrightarrow{o} \text{貸出クラス}$

貸出を作成する (ある貸出日, ある借用者, ある本実体) \triangle

```
( dcl ある貸出 : 貸出クラス := new 貸出クラス ();  
  ある貸出.初期化する (ある貸出日, ある借用者, ある本実体);  
  return ある貸出  
);
```

public

本を貸す : $Date \times \text{本実体クラス} \times \text{一般利用者クラス} \xrightarrow{o} \text{貸出クラス}$

本を貸す (貸出日, ある本実体, ある利用者) \triangle

```
( let 貸出 = ある利用者.借りる (貸出日, ある本実体, self) in  
  return 貸出  
)
```

pre ある本実体 \in 図書館.蔵書 \wedge

ある本実体.貸出可能 () \wedge

ある利用者 \in 図書館.利用者 \wedge

ある利用者.借用可能 (貸出日) \wedge

図書館.貸出可能 (ある利用者.借用数 ());

public

```

本を返す :  $Date \times \text{本実体クラス} \xrightarrow{o} \text{図書館クラス}$ 
本を返す(返却日, 借用本)  $\triangleq$ 
    ( 借用本.借用者を得る ( ) .返す(返却日, 借用本);
      return 図書館
    )
pre 借用本  $\in$  図書館.蔵書  $\wedge$ 
    借用本.貸出可能 ( ) = false  $\wedge$ 
    借用本.借用者を得る ( )  $\neq \text{nil}$  ;
public
    職員 : ( )  $\xrightarrow{o} \mathbb{B}$ 
    職員 ( )  $\triangleq$ 
        return true
end 職員クラス
Test Suite :      vdm.tc
Class :          職員クラス

```

Name	#Calls	Coverage
職員クラス ‘職員	39	✓
職員クラス ‘本を貸す	36	✓
職員クラス ‘本を返す	6	✓
職員クラス ‘貸出を作成する	70	✓
Total Coverage		100%

2.8 図書館クラス

システムに関するリソース（人と本）の管理を行う主体であり、運用ルールも持つ。図書館クラスは、図書館のインスタンスを生成・管理するとともに、利用者・本・司書・著者・分野の各クラスのインスタンスの生成・管理を行う。

class 図書館クラス

values

- = "\$Header: /home/vdmttools/cvsroot/toolbox/examples-jp/pp/library/Library.vpp, v 1.1 2

20:16 vdmttools Exp \$"

instance variables

最大貸出数 : $N_1 := 3$;

public 蔵書 : 本実体クラス-set := {};

public 利用者 : 利用者クラス-set := {};

operations

public

蔵書を追加する : 本実体クラス \times 職員クラス \xrightarrow{o} 図書館クラス

蔵書を追加する (ある本実体, 操作者) \triangle

(蔵書 := 蔵書 \cup {ある本実体};

return self

)

pre 操作者.職員() \wedge

ある本実体 \notin 蔵書

post

ある本実体 \in 蔵書;

public

利用者を追加する : 利用者クラス \times 職員クラス \xrightarrow{o} 図書館クラス

利用者を追加する (ある利用者, 操作者) \triangle

(利用者 := 利用者 \cup {ある利用者};

return self

)


```

pre  操作者.職員 () ∧
    ある利用者 ∉ 利用者

post
ある利用者 ∈ 利用者;

public
本を検索する : (著者クラス | 分野クラス)  $\xrightarrow{o}$  本クラス-set
本を検索する (検索キー)  $\triangle$ 
    return {ある蔵書.本 | ある蔵書 ∈ 蔵書 ·
        if isofclass (著者クラス, 検索キー)
        then ある蔵書.本.著者 = 検索キー
        else 検索キー ∈ ある蔵書.本.分野集合 };

public
貸出可能 :  $\mathbb{N} \xrightarrow{o} \mathbb{B}$ 
貸出可能 (貸出冊数)  $\triangle$ 
    return 貸出冊数 < 最大貸出数;

public
最後の利用者 : 本実体クラス × 利用者クラス  $\xrightarrow{o}$  [利用者クラス]
最後の利用者 (ある本実体, 操作者)  $\triangle$ 
    return ある本実体.借用者を得る ()
pre  操作者.職員 () ∧
    ある本実体 ∈ 蔵書 ;

public
本を作成する :  $\text{char}^* \times \text{著者クラス} \times \text{分野クラス-set} \xrightarrow{o} \text{本クラス}$ 
本を作成する (ある題名, ある著者, ある分野集合)  $\triangle$ 
    (
        dcl 本 : 本クラス := new 本クラス ();
        本.初期化する(ある題名, ある著者, ある分野集合);
        return 本
    );

public

```

本実体を作成する : $\text{char}^* \times \text{本クラス} \xrightarrow{o} \text{本実体クラス}$

本実体を作成する (ある ID , ある本) \triangle

```
(    dcl 本実体 : 本実体クラス := new 本実体クラス ();  
      本実体.初期化する(ある  $ID$ , ある本);  
      return 本実体  
);
```

public

分野を作成する : $\text{char}^* \xrightarrow{o} \text{分野クラス}$

分野を作成する (ある分野名) \triangle

```
(    dcl 分野 : 分野クラス := new 分野クラス ();  
      分野.初期化する(ある分野名);  
      return 分野  
);
```

public

著者を作成する : $\text{char}^* \times \text{char}^* \xrightarrow{o} \text{著者クラス}$

著者を作成する (ある名前, ある電話番号) \triangle

```
(    dcl 著者 : 著者クラス := new 著者クラス ();  
      著者.初期化する(ある名前, ある電話番号);  
      return 著者  
);
```

public

職員を作成する : $\text{図書館クラス} \times \text{char}^* \times \text{char}^* \times \text{char}^* \xrightarrow{o} \text{職員クラス}$

職員を作成する (ある図書館, ある ID , ある名前, ある電話番号) \triangle

```
(    dcl 職員 : 職員クラス := new 職員クラス ();  
      職員. $ID$  を設定する(ある  $ID$ );  
      職員.初期化する(ある名前, ある電話番号);  
      職員.図書館を設定する(ある図書館);  
      return 職員  
);
```

public static

利用者を作成する : 図書館クラス \times char* \times char* \times char* \xrightarrow{o} 利用者クラス

利用者を作成する (ある図書館, ある ID, ある名前, ある電話番号) \triangle

```
( dcl 一般利用者 : 一般利用者クラス := new 一般利用者クラス ();  
  一般利用者.ID を設定する(ある ID);  
  一般利用者.初期化する(ある名前, ある電話番号);  
  一般利用者.図書館を設定する(ある図書館);  
  return 一般利用者  
)
```

end 図書館クラス

Test Suite : vdm.tc

Class : 図書館クラス

Name	#Calls	Coverage
図書館クラス ‘貸出可能	18	✓
図書館クラス ‘最後の利用者	6	✓
図書館クラス ‘本を作成する	36	✓
図書館クラス ‘本を検索する	10	94%
図書館クラス ‘分野を作成する	38	✓
図書館クラス ‘職員を作成する	12	✓
図書館クラス ‘著者を作成する	24	✓
図書館クラス ‘蔵書を追加する	48	✓
図書館クラス ‘利用者を作成する	20	✓
図書館クラス ‘利用者を追加する	20	✓
図書館クラス ‘本実体を作成する	48	✓
Total Coverage		99%

```

class 人クラス
instance variables
    public 名前 : char*;
    public 電話番号 : char*;

operations
public
    初期化する :  $\text{char}^* \times \text{char}^* \xrightarrow{o} ()$ 
    初期化する (ある名前, ある電話番号)  $\triangle$ 
        (   名前 := ある名前;
           電話番号 := ある電話番号
        )
end 人クラス

```

2.9 分野クラス

本の分野を表す。

```
class 分野クラス
```

```
values
```

```
    - = "$Header: /home/vdmtools/cvsroot/toolbox/examples-jp/pp/library/Subject.vpp, v 1.1 2000-01-16 16:16:16 vdmtools Exp $"
```

```
20 : 16 vdmtools Exp $"
```

```
instance variables
```

```
    public 分野名 : char*;
```

```
operations
```

```
public
```

```
    初期化する : char*  $\xrightarrow{o}$  ()
```

```
    初期化する (ある分野名)  $\triangle$ 
```

```
        ( 分野名 := ある分野名  
          )
```

```
end 分野クラス
```

2.10 一般利用者クラス

蔵書を借りる主体であり、セキュリティ上の理由で、システムに登録されていなければならない。

class 一般利用者クラス is subclass of 利用者クラス

values

- = "\$Header: /home/vdmttools/cvsroot/toolbox/examples-jp/pp/library/User.vpp, v 1.1 200

20:16 vdmttools Exp \$"

operations

public

職員: $() \xrightarrow{o} \mathbb{B}$

職員 $() \triangle$

return false

end 一般利用者クラス

2.11 図書館貸出管理クラス T

図書館貸出管理クラスの回帰テストを行う

class 図書館貸出管理クラス T is subclass of $TestDriver$

values

- = "\$Header: /home/vdmtools/cvsroot/toolbox/examples-jp/pp/library/testCase.vpp, v 1.2

53:28 vdmtools Exp \$"

functions

$tests : () \rightarrow TestCase^*$

$tests () \triangleq$

[
new 図書館貸出管理クラス $T01 ()$,
new 図書館貸出管理クラス $T02 ()$,
new 図書館貸出管理クラス $T03 ()$,
new 図書館貸出管理クラス $T04 ()$,
new 図書館貸出管理クラス $T05 ()$,
new 図書館貸出管理クラス $T06 ()$]

end 図書館貸出管理クラス T

class 図書館貸出管理クラス $T01$ is subclass of $TestCase$

operations

protected

$test : () \xrightarrow{o} \mathbb{B}$

$test () \triangleq$

let $today = new SBCalendar ().基準日 ()$,
図書館 = new 図書館クラス (),
主題 1 = 図書館.分野を作成する ("ソフトウェア"),
主題 2 = 図書館.分野を作成する ("小説"),
主題 3 = 図書館.分野を作成する ("戦争"),
著者 1 = 図書館.著者を作成する ("佐原伸", "0422-55-7724"),
著者 2 = 図書館.著者を作成する ("井上ひさし", "不明"),
本 1 = 図書館.本を作成する ("デザインパターン", 著者 1, {主題 1}),
本 2 = 図書館.本を作成する ("紙屋町桜ホテル", 著者 2, {主題 2, 主題 3}),

```

    本 3 = 図書館.本を作成する ("夢の裂け目", 著者 2, {主題 2, 主題 3}),
    本実体 1 = 図書館.本実体を作成する ("101", 本 1),
    本実体 2 = 図書館.本実体を作成する ("102", 本 1),
    本実体 3 = 図書館.本実体を作成する ("103", 本 2),
    本実体 4 = 図書館.本実体を作成する ("104", 本 3),
    職員 1 = 図書館.職員を作成する (図書館, "11", "宮崎寅蔵", "不明"),
    利用者 1 = 図書館.利用者を作成する (図書館, "10", "チェ・ゲバラ", "不明"),
  ),

```

```

    利用者 2 = 図書館.利用者を作成する (図書館, "12", "胡志明", "不明"),
    図書館 0 = 図書館.利用者を追加する (利用者 1, 職員 1),
    図書館 1 = 図書館 0.利用者を追加する (利用者 2, 職員 1),
    図書館 2 = 図書館 1.蔵書を追加する (本実体 1, 職員 1),
    図書館 3 = 図書館 2.蔵書を追加する (本実体 2, 職員 1),
    図書館 4 = 図書館 3.蔵書を追加する (本実体 3, 職員 1),
    - = 図書館 4.蔵書を追加する (本実体 4, 職員 1),
    - = 職員 1.本を貸す (today, 本実体 1, 利用者 1),
    - = 職員 1.本を貸す (today, 本実体 3, 利用者 1),
    - = 職員 1.本を貸す (today, 本実体 4, 利用者 1) in
  return 利用者 1.借りている本 (職員 1) = {本 1, 本 2, 本 3} ∧
    利用者 1.借りている本 (利用者 1) = {本 1, 本 2, 本 3} ∧
    利用者 2.借りている本 (職員 1) = {} ∧
    利用者 2.借りている本 (利用者 2) = {};

```

protected

準備する : $() \xrightarrow{o} ()$

準備する $() \triangleleft$

テスト名 := "図書館貸出管理クラス T01 : \t 借りている本を求める";

protected

後始末する : $() \xrightarrow{o} ()$

後始末する $() \triangleleft$

return

end 図書館貸出管理クラス T01

class 図書館貸出管理クラス T02 is subclass of *TestCase*

operations

protected

$test : () \xrightarrow{o} \mathbb{B}$

$test () \triangleq$

let $today = new\ SBCalendar ()$.基準日 (),

図書館 = new 図書館クラス (),

主題 1 = 図書館.分野を作成する ("ソフトウェア"),

主題 2 = 図書館.分野を作成する ("小説"),

主題 3 = 図書館.分野を作成する ("戦争"),

主題 4 = 図書館.分野を作成する ("VDM"),

著者 1 = 図書館.著者を作成する ("佐原伸", "0422-55-7724"),

著者 2 = 図書館.著者を作成する ("井上ひさし", "不明"),

本 1 = 図書館.本を作成する ("デザインパターン", 著者 1, {主題 1}),

本 2 = 図書館.本を作成する ("紙屋町桜ホテル", 著者 2, {主題 2, 主題 3}),

本 3 = 図書館.本を作成する ("夢の裂け目", 著者 2, {主題 2, 主題 3}),

本実体 1 = 図書館.本実体を作成する ("101", 本 1),

本実体 2 = 図書館.本実体を作成する ("102", 本 1),

本実体 3 = 図書館.本実体を作成する ("103", 本 2),

本実体 4 = 図書館.本実体を作成する ("104", 本 3),

職員 1 = 図書館.職員を作成する (図書館, "11", "宮崎寅蔵", "不明"),

利用者 1 = 図書館.利用者を作成する (図書館, "10", "チェ・ゲバラ", "不明

"),

図書館 1 = 図書館.利用者を追加する (利用者 1, 職員 1),

図書館 2 = 図書館 1.蔵書を追加する (本実体 1, 職員 1),

図書館 3 = 図書館 2.蔵書を追加する (本実体 2, 職員 1),

図書館 4 = 図書館 3.蔵書を追加する (本実体 3, 職員 1),

図書館 5 = 図書館 4.蔵書を追加する (本実体 4, 職員 1),

- = 職員 1.本を貸す ($today$, 本実体 1, 利用者 1),

```

- = 職員 1.本を貸す (today, 本実体 4, 利用者 1) in
return 図書館 5.本を検索する (著者 1) = {本 1} ∧
      図書館 5.本を検索する (著者 2) = {本 2, 本 3} ∧
      図書館 5.本を検索する (主題 1) = {本 1} ∧
      図書館 5.本を検索する (主題 2) = {本 2, 本 3} ∧
      図書館 5.本を検索する (主題 4) = {};

protected
準備する : ()  $\xrightarrow{o}$  ()
準備する ()  $\triangleq$ 
  テスト名 := "図書館貸出管理クラス T02:\t 本を検索する";

protected
後始末する : ()  $\xrightarrow{o}$  ()
後始末する ()  $\triangleq$ 

return
end 図書館貸出管理クラス T02

class 図書館貸出管理クラス T03 is subclass of TestCase
operations
protected
  test : ()  $\xrightarrow{o}$   $\mathbb{B}$ 
  test ()  $\triangleq$ 
    let today = new SBCalendar ().基準日 (),
        図書館 = new 図書館クラス (),
        主題 1 = 図書館.分野を作成する ("ソフトウェア"),
        主題 2 = 図書館.分野を作成する ("小説"),
        主題 3 = 図書館.分野を作成する ("戦争"),
        著者 1 = 図書館.著者を作成する ("佐原伸", "0422-55-7724"),
        著者 2 = 図書館.著者を作成する ("井上ひさし", "不明"),
        本 1 = 図書館.本を作成する ("デザインパターン", 著者 1, {主題 1}),
        本 2 = 図書館.本を作成する ("紙屋町桜ホテル", 著者 2, {主題 2, 主題 3}),
        本 3 = 図書館.本を作成する ("夢の裂け目", 著者 2, {主題 2, 主題 3}),
        本実体 1 = 図書館.本実体を作成する ("101", 本 1),
        本実体 2 = 図書館.本実体を作成する ("102", 本 1),

```

```

    本実体 3 = 図書館.本実体を作成する ("103", 本 2),
    本実体 4 = 図書館.本実体を作成する ("104", 本 3),
    職員 1 = 図書館.職員を作成する (図書館, "11", "宮崎寅蔵", "不明"),
    利用者 1 = 図書館.利用者を作成する (図書館, "10", "チェ・ゲバラ", "不明"),
  ),

```

```

    図書館 1 = 図書館.利用者を追加する (利用者 1, 職員 1),
    図書館 2 = 図書館 1.蔵書を追加する (本実体 1, 職員 1),
    図書館 3 = 図書館 2.蔵書を追加する (本実体 2, 職員 1),
    図書館 4 = 図書館 3.蔵書を追加する (本実体 3, 職員 1),
    図書館 5 = 図書館 4.蔵書を追加する (本実体 4, 職員 1),
    - = 職員 1.本を貸す (today, 本実体 1, 利用者 1),
    - = 職員 1.本を貸す (today, 本実体 4, 利用者 1) in
  return 図書館 5.最後の利用者 (本実体 3, 職員 1) = nil ∧
    図書館 5.最後の利用者 (本実体 4, 職員 1) = 利用者 1;

```

protected

```

  準備する : ()  $\xrightarrow{o}$  ()

```

```

  準備する ()  $\triangle$ 

```

```

  テスト名 := "図書館貸出管理クラス T03 : \t 最後の利用者を求める";

```

protected

```

  後始末する : ()  $\xrightarrow{o}$  ()

```

```

  後始末する ()  $\triangle$ 

```

```

  return

```

end 図書館貸出管理クラス T03

class 図書館貸出管理クラス T04 is subclass of *TestCase*

operations

protected

```

  test : ()  $\xrightarrow{o}$   $\mathbb{B}$ 

```

```

  test ()  $\triangle$ 

```

```

  let today = new SBCalendar ().基準日 (),

```

```

    図書館 = new 図書館クラス (),

```

```

    主題 1 = 図書館.分野を作成する ("ソフトウェア"),

```

```

    主題 2 = 図書館.分野を作成する ("小説"),

```

主題 3 = 図書館.分野を作成する ("戦争"),
 著者 1 = 図書館.著者を作成する ("佐原伸", "0422-55-7724"),
 著者 2 = 図書館.著者を作成する ("井上ひさし", "不明"),
 本 1 = 図書館.本を作成する ("デザインパターン", 著者 1, {主題 1}),
 本 2 = 図書館.本を作成する ("紙屋町桜ホテル", 著者 2, {主題 2, 主題 3}),
 本 3 = 図書館.本を作成する ("夢の裂け目", 著者 2, {主題 2, 主題 3}),
 本実体 1 = 図書館.本実体を作成する ("101", 本 1),
 本実体 2 = 図書館.本実体を作成する ("102", 本 1),
 本実体 3 = 図書館.本実体を作成する ("103", 本 2),
 本実体 4 = 図書館.本実体を作成する ("104", 本 3),
 職員 1 = 図書館.職員を作成する (図書館, "11", "宮崎寅蔵", "不明"),
 利用者 1 = 図書館.利用者を作成する (図書館, "10", "チェ・ゲバラ", "不明"),

利用者 2 = 図書館.利用者を作成する (図書館, "12", "胡志明", "不明"),
 図書館 0 = 図書館.利用者を追加する (利用者 1, 職員 1),
 図書館 1 = 図書館 0.利用者を追加する (利用者 2, 職員 1),
 図書館 2 = 図書館 1.蔵書を追加する (本実体 1, 職員 1),
 図書館 3 = 図書館 2.蔵書を追加する (本実体 2, 職員 1),
 図書館 4 = 図書館 3.蔵書を追加する (本実体 3, 職員 1),
 - = 図書館 4.蔵書を追加する (本実体 4, 職員 1),
 - = 職員 1.本を貸す (*today*, 本実体 1, 利用者 1),
 - = 職員 1.本を貸す (*today*, 本実体 4, 利用者 1),
 図書館 6 = 職員 1.本を返す (*today*, 本実体 1),
 - = 職員 1.本を貸す (*today*, 本実体 1, 利用者 2) in
 return 図書館 6.最後の利用者 (本実体 1, 職員 1) = 利用者 2;

protected

準備する : $() \xrightarrow{o} ()$

準備する $() \triangleleft$

テスト名 := "図書館貸出管理クラス T04:\t 本を返す";

protected

```

    後始末する : ()  $\xrightarrow{o}$  ()
    後始末する ()  $\triangleq$ 

    return
end 図書館貸出管理クラス T04
class 図書館貸出管理クラス T05 is subclass of TestCase
operations
protected
    test : ()  $\xrightarrow{o}$   $\mathbb{B}$ 
    test ()  $\triangleq$ 

    let today = new SBCalendar ().基準日 (),
        図書館 = new 図書館クラス (),
        主題 1 = 図書館.分野を作成する ("ソフトウェア"),
        主題 2 = 図書館.分野を作成する ("小説"),
        主題 3 = 図書館.分野を作成する ("戦争"),
        著者 1 = 図書館.著者を作成する ("佐原伸", "0422-55-7724"),
        著者 2 = 図書館.著者を作成する ("井上ひさし", "不明"),
        本 1 = 図書館.本を作成する ("デザインパターン", 著者 1, {主題 1}),
        本 2 = 図書館.本を作成する ("紙屋町桜ホテル", 著者 2, {主題 2, 主題 3}),
        本 3 = 図書館.本を作成する ("夢の裂け目", 著者 2, {主題 2, 主題 3}),
        本実体 1 = 図書館.本実体を作成する ("101", 本 1),
        本実体 2 = 図書館.本実体を作成する ("102", 本 1),
        本実体 3 = 図書館.本実体を作成する ("103", 本 2),
        本実体 4 = 図書館.本実体を作成する ("104", 本 3),
        職員 1 = 図書館.職員を作成する (図書館, "11", "宮崎寅蔵", "不明"),
        利用者 1 = 図書館.利用者を作成する (図書館, "10", "チェ・ゲバラ", "不明"),

    ),

    利用者 2 = 図書館.利用者を作成する (図書館, "12", "胡志明", "不明"),
    図書館 0 = 図書館.利用者を追加する (利用者 1, 職員 1),
    図書館 1 = 図書館 0.利用者を追加する (利用者 2, 職員 1),
    図書館 2 = 図書館 1.蔵書を追加する (本実体 1, 職員 1),
    図書館 3 = 図書館 2.蔵書を追加する (本実体 2, 職員 1),
    図書館 4 = 図書館 3.蔵書を追加する (本実体 3, 職員 1),

```

```

- = 図書館 4.蔵書を追加する (本実体 4, 職員 1),
貸出 1 = 職員 1.本を貸す (today, 本実体 1, 利用者 1),
- = 職員 1.本を貸す (today, 本実体 4, 利用者 1),
- = 職員 1.本を返す (today, 本実体 1),
- = 職員 1.本を貸す (today, 本実体 1, 利用者 2) in
return 貸出 1.借用者を得る () = nil ∧
      利用者 2.職員 () = false;

protected
準備する : ()  $\xrightarrow{o}$  ()
準備する ()  $\triangle$ 
      テスト名 := "図書館貸出管理クラス T05 : \t その他の検査";

protected
後始末する : ()  $\xrightarrow{o}$  ()
後始末する ()  $\triangle$ 
      return

end 図書館貸出管理クラス T05

class 図書館貸出管理クラス T06 is subclass of TestCase
operations
protected
      test : ()  $\xrightarrow{o}$   $\mathbb{B}$ 
      test ()  $\triangle$ 
      let today = new SBCalendar ().基準日 (),
          図書館 = new 図書館クラス (),
          主題 1 = 図書館.分野を作成する ("ソフトウェア"),
          主題 2 = 図書館.分野を作成する ("小説"),
          主題 3 = 図書館.分野を作成する ("戦争"),
          著者 1 = 図書館.著者を作成する ("佐原伸", "0422-55-7724"),
          著者 2 = 図書館.著者を作成する ("井上ひさし", "不明"),
          本 1 = 図書館.本を作成する ("デザインパターン", 著者 1, {主題 1}),
          本 2 = 図書館.本を作成する ("紙屋町桜ホテル", 著者 2, {主題 2, 主題 3}),
          本 3 = 図書館.本を作成する ("夢の裂け目", 著者 2, {主題 2, 主題 3}),
          本実体 1 = 図書館.本実体を作成する ("101", 本 1),

```

本実体 2 = 図書館.本実体を作成する ("102", 本 1),
 本実体 3 = 図書館.本実体を作成する ("103", 本 2),
 本実体 4 = 図書館.本実体を作成する ("104", 本 3),
 職員 1 = 図書館.職員を作成する (図書館, "11", "宮崎寅蔵", "不明"),
 利用者 1 = 図書館.利用者を作成する (図書館, "10", "チェ・ゲバラ", "不明"),

利用者 2 = 図書館.利用者を作成する (図書館, "12", "胡志明", "不明"),
 図書館 0 = 図書館.利用者を追加する (利用者 1, 職員 1),
 図書館 1 = 図書館 0.利用者を追加する (利用者 2, 職員 1),
 図書館 2 = 図書館 1.蔵書を追加する (本実体 1, 職員 1),
 図書館 3 = 図書館 2.蔵書を追加する (本実体 2, 職員 1),
 図書館 4 = 図書館 3.蔵書を追加する (本実体 3, 職員 1),
 - = 図書館 4.蔵書を追加する (本実体 4, 職員 1),
 - = 職員 1.本を貸す (*today*, 本実体 1, 利用者 1),
 - = 職員 1.本を貸す (*today*, 本実体 2, 利用者 1),
 - = 職員 1.本を貸す (*today*, 本実体 3, 利用者 1),
 - = 職員 1.本を返す (*today*, 本実体 1),
 - = 職員 1.本を貸す (*today*, 本実体 4, 利用者 1),
 - = 職員 1.本を貸す (*today*, 本実体 1, 利用者 1) in
 return 利用者 1.借りている本 (職員 1) = {本 1, 本 2, 本 3} ∧
 利用者 1.借りている本 (利用者 1) = {本 1, 本 2, 本 3} ∧
 利用者 2.借りている本 (職員 1) = {} ∧
 利用者 2.借りている本 (利用者 2) = {};

protected

準備する : () \xrightarrow{o} ()

準備する () \triangleq

テスト名 := "図書館貸出管理クラス T06 : \t 制約違反の検査";

protected

後始末する : () \xrightarrow{o} ()

後始末する () \triangleq

return

end 図書館貸出管理クラス T06