オブジェクト指向設計　予習範囲・暗記事項・予習ノート・練習問題　for 講義 No.6

・暗記事項を参考にして予習ノートの作成
・予習範囲の熟読
・暗記事項の暗記
・教科書の演習問題を理解
上記を済ませてから練習問題に取り組んでください。
最終ページに解答を載せています。

予習範囲
3.5 節です.
これまでの確認テストも復習として 1〜2 点の配点で出しますので勉強しておいてください.

暗記事項
【暗記事項 1】から【暗記事項 10】までは，すべて、「A: 正しい」が正答となります。予習確認テストでは表現を変えて「B: 間違っている」が答えになることもあります.
ただ覚えるだけでなく，教科書中の当該事項を説明した文章を読んで理解につとめてください。

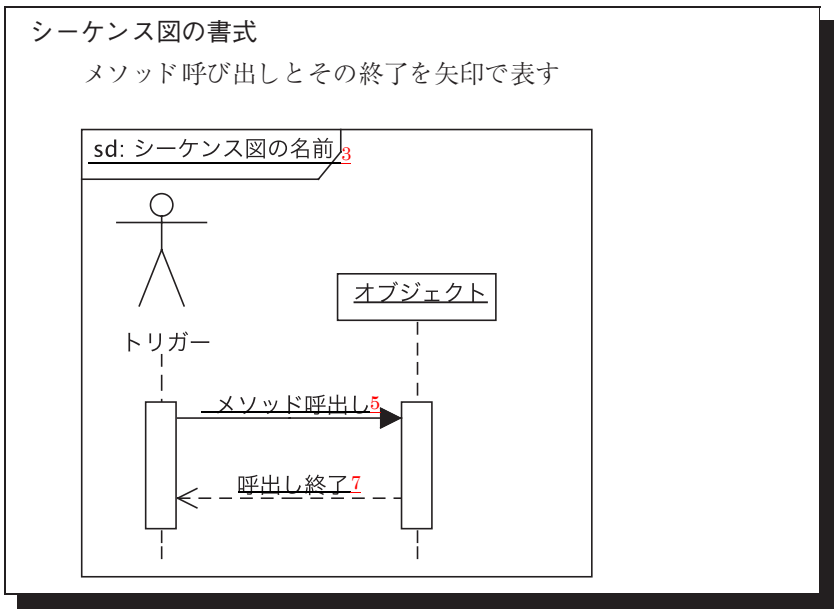
【暗記事項 1】シーケンス図はオブジェクト同士のメッセージのやり取りを表現する図である.
【暗記事項 2】シーケンス図では，時系列に沿って表現する ．
【暗記事項 3】シーケンス図では左上に SD:と書いてから図の名前を書く ．
【暗記事項 4】シーケンス図のメッセージ（呼出し）では，線の上側に「メソッド名（引数リスト)」を書く.
【暗記事項 5】シーケンス図のメッセージ（呼出し）では，呼び出し元から呼び出し先へ線を引く.
【暗記事項 6】シーケンス図のメッセージ（戻り値）では，線の上側に
「戻り値の型＝メソッド名（引数リスト）: 戻り値」を書く ．
【暗記事項 7】シーケンス図のメッセージ（戻り値）では，呼び出し先から呼び出し元へ線を引く ．
【暗記事項 8】シーケンス図の自己メッセージとは，自クラスのメソッドを呼び出すことを表す.
【暗記事項 9】シーケンス図でライフラインの生成とはコンストラクタ呼出しを表している.
【暗記事項 10】シーケンス図でライフラインの停止とはオブジェクトの消滅を表している.

3.5節　シーケンス図

シーケンス図

・オブジェクト同士の　　（メソッド呼出し）のやり取りを表現する図[1]

・オブジェクト間の相互作用，特に，オブジェクト間のメソッド呼出しによる制御の遷移を呼出しの　　に沿って表現する[2]



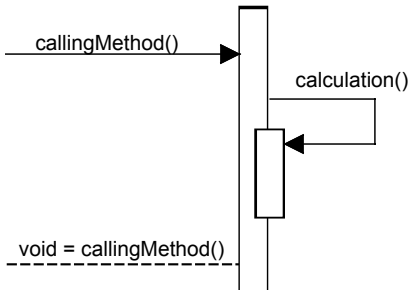
・メソッド呼び出しの書式：線の上側に「　　（　　）」を書く[4]

・メソッド終了時の書式：　線の上側に「　　（　　）：　　」を書く[6]

練習問題では戻り値は省略しています.

自己メッセージ

　　のメソッドを呼び出すことを表す[8]



calculation()が自己メッセージ

ライフライン

・生成：　　の呼出しを表す[9]

・停止：　　の消滅を表す[10]

練習問題

確認テストも，この練習問題と同じ方法で，解答してください．最終ページに答えを載せています．

解答が複数ある場合は、ハイフン で繋いで答えること。

例: A と B と C を解答したい場合，A-B-C と解答欄に記入する。答えがない場合は-1 と解答すること。

行番号を解答するとき、左詰めの 0 は取ること

例: 001 行目を解答するときは 1 を解答すること

System.out.println()（改行あり）は main メソッドにしかありません。

他の場所では、System.out.print()（改行なし）を使っています

行番号を振っていない空行に続いて、行番号を振りなおしているソースコードが続く場合は、

別の Java ファイル（クラス）であることを示しています。

本練習問題の答え方

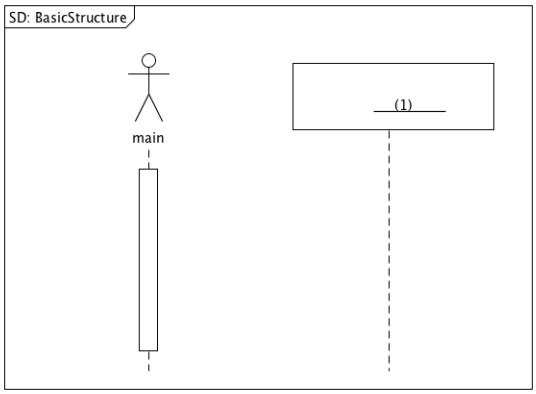
以下のソースコードについて【練習 01】から【練習 11】を解答しなさい

```
01 public class BasicStructure {
02   private int data;
03   public BasicStructure(int data) {
04     this.data = data;
05           System.out.print(“0”);
06   }
07   public void printData() {
08     System.out.print(data);
09   }
10 }

21 public class TestDrive04_3 {
22   public static void main(String[] args) {
23     BasicStructure basicStructure = new BasicStructure(3);
24     basicStructure.printData();
25     basicStructure = new BasicStructure(2);
26     System.out.println("a");
27     basicStructure.printData();
28     System.out.println("b");
29   }
30 }
```

【練習 01】出力の一行目を答えなさい　　030a

【練習 02】出力の二行目を答えなさい　　2b

右のシーケンス図を埋めなさい



複数のオブジェクトがある場合は，先に生成されたオブジェクトが左に来るように解答してください

System.out.println(), System.out.print()は矢印を引く必要はありません

矢印のついた線を記述する場合は，始点（矢印のない方）を左側に，終点（矢印のある方）を右に書くようにしてください．オブジェクト生成ならびにメソッド呼び出しを表す線はハイフン『-』を，メソッド処理の終了を表す線はドット『.』を用いてください．

アクターはライフライン（0）としています．

【練習 03】(1)のライフライン　　basicStructure:BasicStructure

【練習 04】一つ目の矢印のついた線・記号　　0-1: BasicStructure(3)

【練習 05】二つ目の矢印のついた線・記号　　0-1: printData()

解答の解釈：アクター（＝ライフライン(0)）からライフライン(1)まで矢印のついた線を**実線**-で引く．コロン:の後ろは矢印の上のメッセージ

【練習 06】三つ目の矢印のついた線・記号　　1.0: void=printData()

解答の解釈：ライフライン(1)からアクター（＝ライフライン(0)）まで矢印のついた線を**点線**.で引く．コロン:の後ろは矢印の上のメッセージ，以下同様

【練習 07】四つ目の矢印のついた線・記号　　1: x

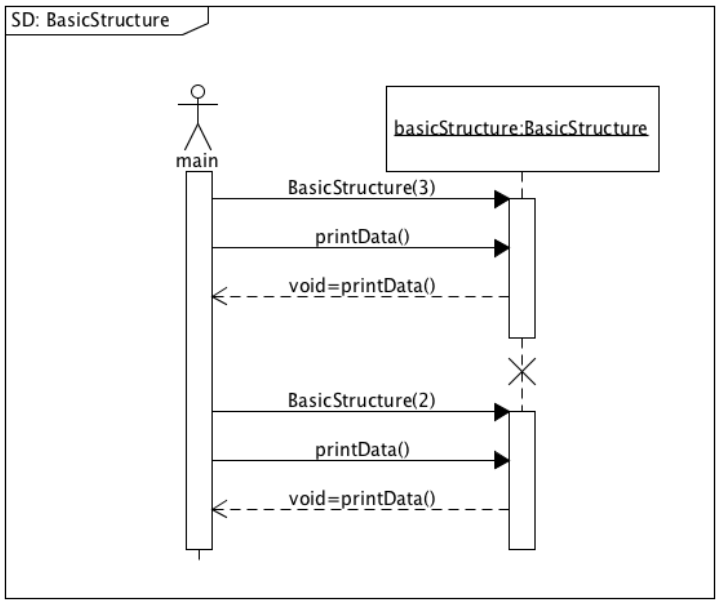
解答の解釈：25 行目で参照型変数 basicStructure は新しいオブジェクトを参照しているので，23 行目の右辺で生成されたライフライン(1)のオブジェクトは消滅 x している．消滅を解答するときは『小文字のエックス』を解答してください．

【練習 08】五つ目の矢印のついた線・記号　　0-1: BasicStructure(2)

【練習 09】六つ目の矢印のついた線・記号　　0-1: printData()

【練習 10】七つ目の矢印のついた線・記号　　1.0: void=printData()

【練習 11】八つ目の矢印のついた線・記号　　-1　　解答すべき矢印がないときは-1 とすること



以下のソースコードについて【練習 12】から【練習 19】を解答しなさい

```
01 public class ConstructorOnly {
02   public ConstructorOnly() {
03     System.out.print("1");
04   }
05 }

11 public class TestDrive04_1 {
12   public static void main(String[] args) {
13     System.out.println("a");
14     ConstructorOnly constructorOnly;
15     System.out.println("b");
16     constructorOnly = new ConstructorOnly();
17     System.out.println("c");
18     ConstructorOnly co = new ConstructorOnly();
19     System.out.println("d");
20   }
21 }
```

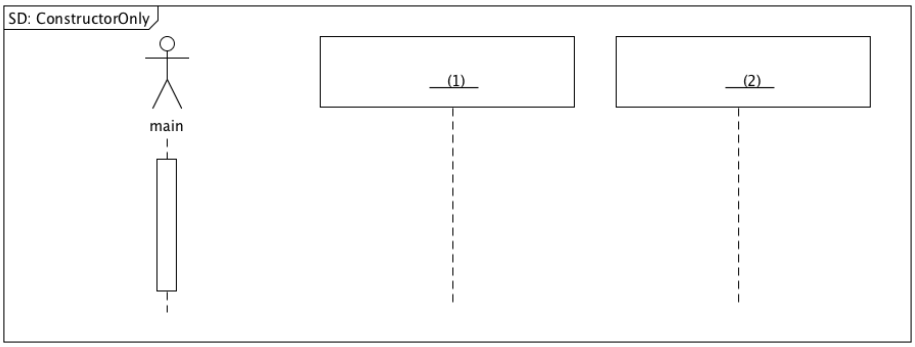
【練習 12】出力の一行目を答えなさい

【練習 13】出力の二行目を答えなさい

【練習 14】出力の三行目を答えなさい

【練習 15】出力の四行目を答えなさい

右のシーケンス図を埋めなさい



【練習 16】(1)のライフライン

【練習 17】(2)のライフライン

【練習 18】一つ目の矢印のついた線・記号

【練習 19】二つ目の矢印のついた線・記号

以下のソースコードについて【練習 20】から【練習 26】を解答しなさい

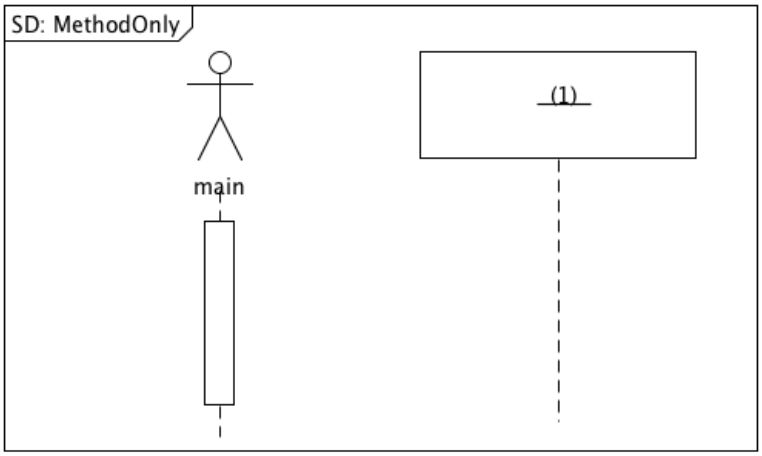
```
01 public class MethodOnly {
02   public void method() {
03     System.out.print("1");
04   }
05 }

11 public class TestDrive04_2 {
12   public static void main(String[] args) {
13     System.out.println("a");
14     MethodOnly methodOnly = new MethodOnly();
15     System.out.println("b");
16     methodOnly.method();
17     System.out.println("c");
18   }
19 }
```

【練習 20】出力の一行目を答えなさい

【練習 21】出力の二行目を答えなさい

【練習 22】出力の三行目を答えなさい

右のシーケンス図を埋めなさい



【練習 23】(1)のライフライン

【練習 24】一つ目の矢印のついた線・記号

【練習 25】二つ目の矢印のついた線・記号

【練習 26】三つ目の矢印のついた線・記号

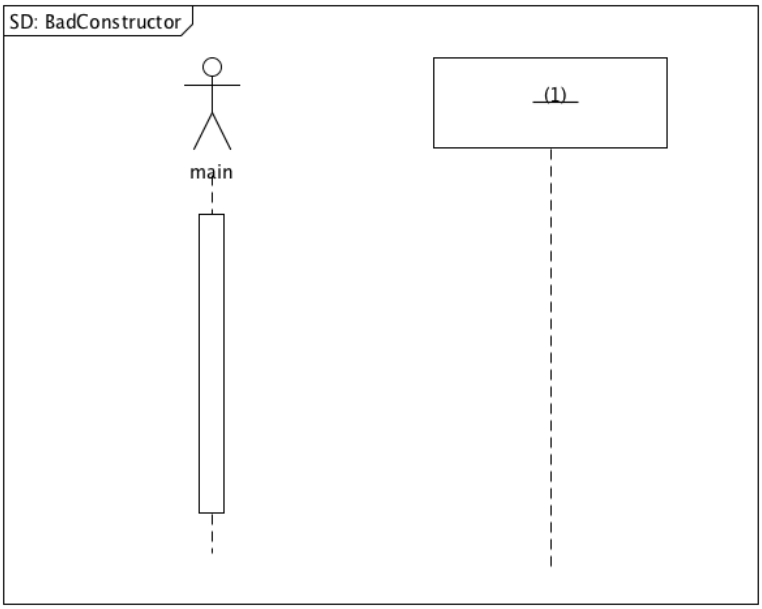
以下のソースコードについて【練習 27】から【練習 36】を解答しなさい

```
01 public class BadConstructor {
02   private int data;
03   public BadConstructor(int data) {
04     data = data;
05   }
06   public void print() {
07     System.out.print(data);
08   }
09 }

11 public class TestDrive04_3b {
12   public static void main(String[] args) {
13     BadConstructor badConstructor = new BadConstructor(2);
14     badConstructor.print();
15     badConstructor = new BadConstructor(3);
16     System.out.println("a");
17     badConstructor.print();
18     System.out.println("b");
19   }
20 }
```

【練習 27】出力の一行目を答えなさい

【練習 28】出力の二行目を答えなさい

右のシーケンス図を埋めなさい



【練習 29】(1)のライフライン

【練習 30】一つ目の矢印のついた線・記号

【練習 31】二つ目の矢印のついた線・記号

【練習 32】三つ目の矢印のついた線・記号

【練習 33】四つ目の矢印のついた線・記号

【練習 34】五つ目の矢印のついた線・記号

【練習 35】六つ目の矢印のついた線・記号

【練習 36】七つ目の矢印のついた線・記号

以下のソースコードについて【練習 37】から【練習 44】を解答しなさい

```
01 public class MethodWithReturn {
02   public int method() {
03     System.out.print("1");
04     return 2;
05   }
06 }

11 public class TestDrive04_4 {
12   public static void main(String[] args) {
13     System.out.println("a");
14     MethodWithReturn methodWithReturn = new MethodWithReturn();
15     System.out.println("b");
16     int a = methodWithReturn.method();
17     System.out.println("c");
18     System.out.println(a);           // System.out.println(“a”)でないことに注意！
19   }
20 }
```

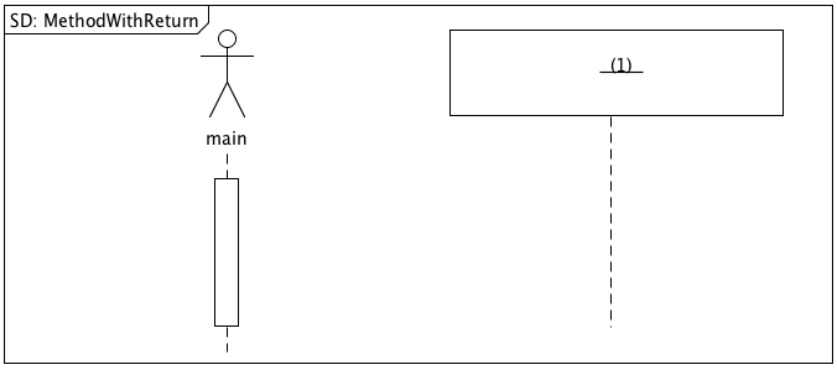
【練習 37】出力の一行目を答えなさい

【練習 38】出力の二行目を答えなさい

【練習 39】出力の三行目を答えなさい

【練習 40】出力の四行目を答えなさい

右のシーケンス図を埋めなさい



【練習 41】(1)のライフライン

【練習 42】一つ目の矢印のついた線・記号

【練習 43】二つ目の矢印のついた線・記号

【練習 44】三つ目の矢印のついた線・記号

以下のソースコードについて【練習 45】から【練習 54】を解答しなさい

```
001  public class Dog {
002      public Dog(String name){
003          System.out.print("0");
004      }
005      public void walk(){
006          System.out.print("1");
007      }
008  }

011  public class Bird {
012      public Bird(){
013          System.out.print("2");
014      }
015      public void fly(){
016          System.out.print("3");
017      }
018  }

021  public class Main {
022      public static void main(String[] args) {
023          Dog dog = new Dog("ポチ");
024          dog.walk();
025          System.out.println("a");
026
027          Bird bird = new Bird();
028          bird.fly();
029          System.out.println("b");
030      }
031  }
```

【練習 45】出力の一行目を答えなさい

【練習 46】出力の二行目を答えなさい

下のシーケンス図を埋めなさい

【練習 47】(1)のライフライン

【練習 48】(2)のライフライン

【練習 49】一つ目の矢印のついた線・記号

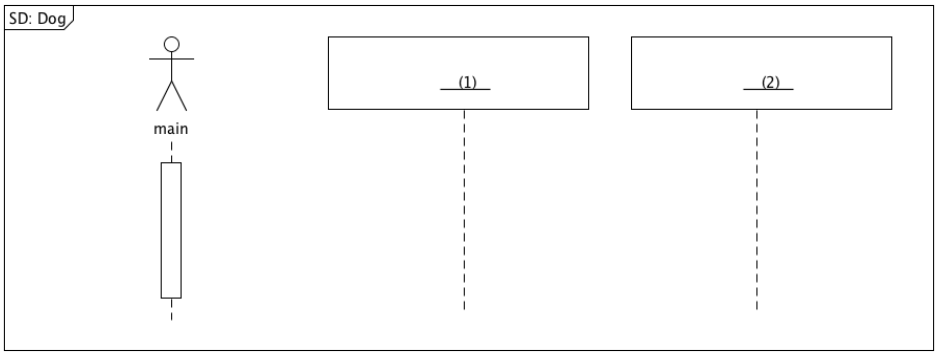
【練習 50】二つ目の矢印のついた線・記号

【練習 51】三つ目の矢印のついた線・記号

【練習 52】四つ目の矢印のついた線・記号

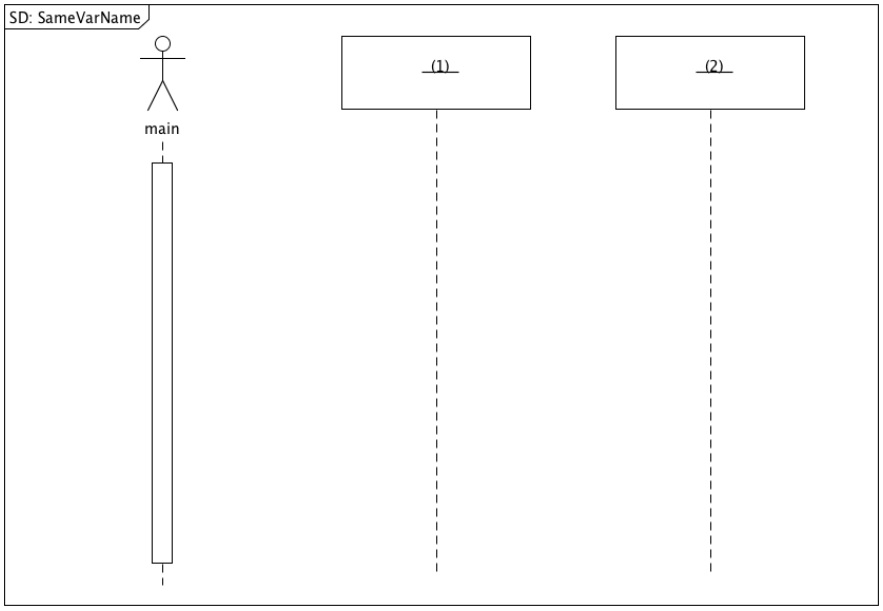
【練習 53】五つ目の矢印のついた線・記号

【練習 54】六つ目の矢印のついた線・記号



以下のソースコードについて【練習 55】から【練習 73】を解答しなさい

```
001  public class SameVarName {
002      private int var;
003      public SameVarName(int var) {
004          System.out.print("z");
005          this.var = var;
006      }
007      public void multiply(int var) {
008          System.out.print("y");
009          this.var *= var;
010      }
011      public int calc() {
012          System.out.print("x");
013          int var = 2;
014          var += this.var;
015          return var;
016      }
017      public int getVar() {
018          System.out.print("w");
019          return var;
020      }
021  }
```




```
031  public class TestDrive04_7 {
032      public static void main(String[] args) {
033          int var = 2;
034          System.out.print("a");
035          SameVarName sameVarName1 = new SameVarName(var);
036          SameVarName sameVarName2 = new SameVarName(var*2);
037          System.out.println("b");
038
039          sameVarName1.multiply(var);
040          sameVarName2.multiply(var);
041          var = sameVarName2.calc();
042          sameVarName2.multiply(var);
043          int var2 = sameVarName2.getVar();
044          System.out.println("c");
045          System.out.println(sameVarName1.calc());
046          System.out.println(var);
047          System.out.println(var2);
048      }
049  }
```

【練習 55】出力の一行目を答えなさい

【練習 56】出力の二行目を答えなさい

【練習 57】出力の三行目を答えなさい

【練習 58】出力の四行目を答えなさい

【練習 59】出力の五行目を答えなさい

右上のシーケンス図を埋めなさい

【練習 60】(1)のライフライン

【練習 61】(2)のライフライン

【練習 62】一つ目の矢印のついた線・記号

【練習 63】二つ目の矢印のついた線・記号

【練習 64】三つ目の矢印のついた線・記号

【練習 65】四つ目の矢印のついた線・記号

【練習 66】五つ目の矢印のついた線・記号

【練習 67】六つ目の矢印のついた線・記号

【練習 68】七つ目の矢印のついた線・記号

【練習 69】八つ目の矢印のついた線・記号

【練習 70】九つ目の矢印のついた線・記号

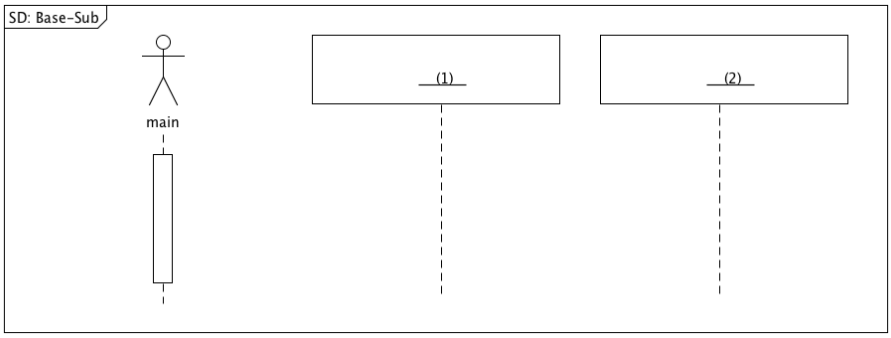
【練習 71】10 個目の矢印のついた線・記号

【練習 72】11 個目の矢印のついた線・記号

【練習 73】12 個目の矢印のついた線・記号

以下のソースコードについて【練習 74】から【練習 73】を解答しなさい

```
001  public class Base {
002      private int x;
003
004      public Base(int x) {
005          this.x = x;
006          System.out.print("0");
007      }
008      public int methodA(int k) {
009          System.out.print("1");
010          return k*x;
011      }
012      public int methodB(int k) {
013          System.out.print("2");
014          int a = methodA(k);
015          System.out.print("3");
016          return a*x*2;
017      }
018  }

021  public class Sub {
022      private int x;
023
024      public Sub(int x) {
025          this.x = x;
026          System.out.print("3");
027      }
028      public int methodA() {
029          System.out.print("4");
030          return x*x;
031      }
032      public int methodB(int k) {
033          System.out.print("5");
034          return x*k;
035      }
036  }

041  public class Main {
042      public static void main(String[] args) {
043          Base base = new Base(3);
044          base.methodB(1);
045          System.out.println("a");
046
047          Sub sub = new Sub(10);
048          sub.methodA();
049          sub.methodB(2);
050          System.out.println("b");
051      }
052  }
```



【練習 74】出力の一行目を答えなさい

【練習 75】出力の二行目を答えなさい

右のシーケンス図を埋めなさい

【練習 76】(1)のライフライン

【練習 77】(2)のライフライン

【練習 78】一つ目の矢印のついた線・記号

【練習 79】二つ目の矢印のついた線・記号

【練習 80】三つ目の矢印のついた線・記号

14 行目の自クラスのメソッド methodA()の呼び出しです.

1-1: methodA(k)　と解答して下さい

【練習 81】四つ目の矢印のついた線・記号

上の自クラスのメソッド呼び出しの終了した時点

（この【練習 81】に相当します）で,

1.1: int=methodA(k)と解答して下さい

【練習 82】五つ目の矢印のついた線・記号

【練習 83】六つ目の矢印のついた線・記号

【練習 84】七つ目の矢印のついた線・記号

【練習 85】八つ目の矢印のついた線・記号

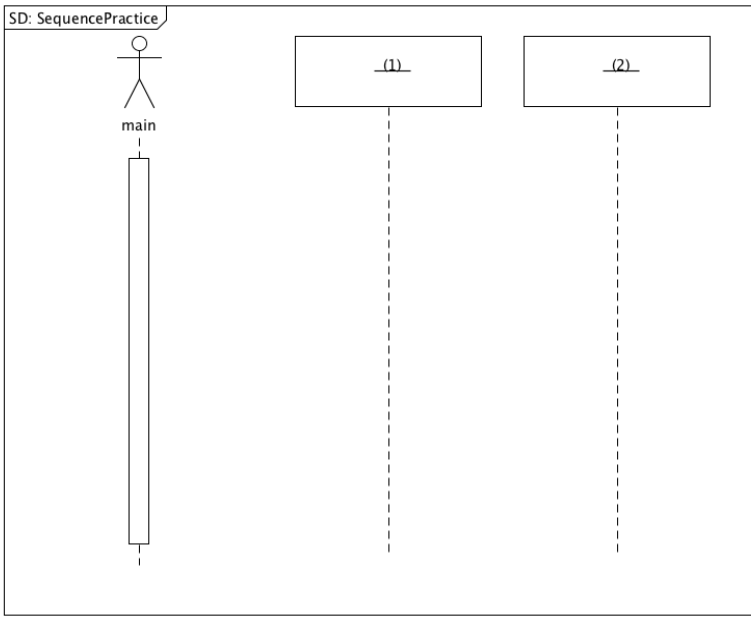
【練習 86】九つ目の矢印のついた線・記号

【練習 87】10 個目の矢印のついた線・記号

これ以降のシーケンス図の問題は中間テストのための練習問題である．中間試験では，シーケンス図の他，これまでの確認テスト，授業中の演習問題から出題される．

以下のソースコードについて【練習 88】から【練習 102】を解答しなさい

```
001   public class SequencePractice1 {
002         public SequencePractice1() {
003               System.out.print("0");
004         }
005         public void meet() {
006               System.out.print("1");
007               hello();
008               System.out.print("2");
009         }
010         public void hello() {
011               System.out.print("3");
012               SequencePractice2 sp = new SequencePractice2();
013               System.out.print("4");
014               sp.talk();
015               System.out.print("5");
016         }
017         public void bye() {
018               System.out.print("6");
019         }
020   }

031   public class SequencePractice2 {
032         public SequencePractice2() {
033               System.out.print("y");
034         }
035         public void talk() {
036               System.out.print("z");
037         }
038   }

041   public class Main {
042         public static void main(String[] args) {
043               SequencePractice1 sp = new SequencePractice1();
044               sp.meet();
045               System.out.println("a");
046               sp.bye();
047               System.out.println("b");
048         }
049
050   }
```



【練習 88】出力の一行目を答えなさい

【練習 89】出力の二行目を答えなさい

右上のシーケンス図を埋めなさい

【練習 90】(1)のライフライン

【練習 91】(2)のライフライン

【練習 92】一つ目の矢印のついた線・記号

【練習 93】二つ目の矢印のついた線・記号

【練習 94】三つ目の矢印のついた線・記号

【練習 95】四つ目の矢印のついた線・記号

【練習 96】五つ目の矢印のついた線・記号

【練習 97】六つ目の矢印のついた線・記号

【練習 98】七つ目の矢印のついた線・記号

2: x と解答してください

(【練習 95】で作成したオブジェクトが消滅するため)

【練習 99】八つ目の矢印のついた線・記号

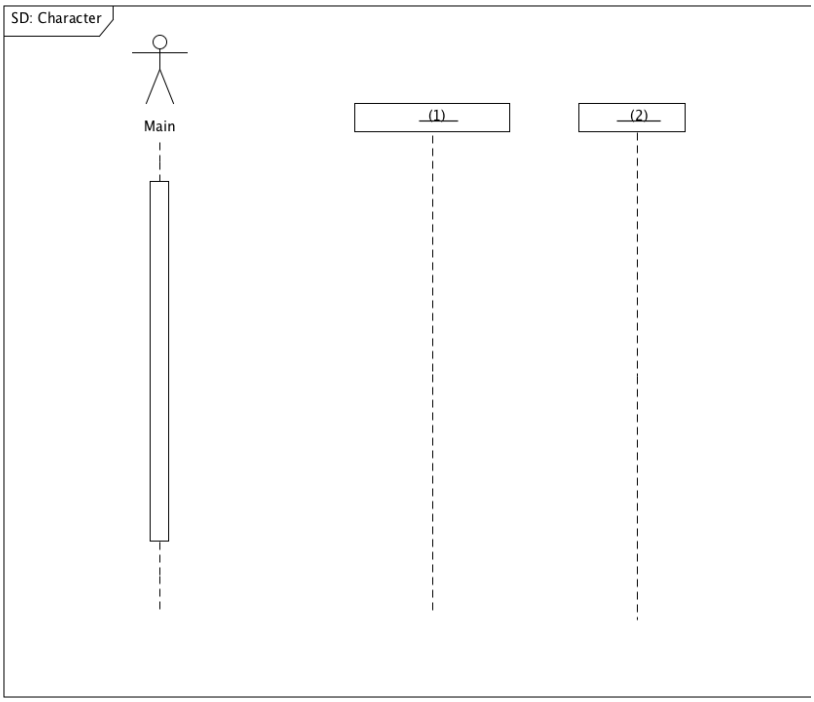
【練習 100】九つ目の矢印のついた線・記号

【練習 101】10 個目の矢印のついた線・記号

【練習 102】11 個目の矢印のついた線・記号

以下のソースコードについて【練習 103】から【練習 110】を解答しなさい

```
001  public class Armor {
002      private int dif;
003
004      public Armor(int dif){
005          this.dif = dif;
006      }
007      public int hitDamage(int power){
008          int damage = power - (dif / 2 + 5);
009          return damage;
010      }
011
012      public int getDif(){
013          return dif;
014      }
015  }

021  public class Character {
022      private String name;
023      private int HP;
024      private Armor armor;
025
026      public Character(String name, int HP){
027          this.name = name;
028          this.HP = HP;
029          armor = new Armor(20);
030      }
031      public void hitDamage(int power){
032          int damage = armor.hitDamage(power);
033          HP -= damage;
034      }
035      public String getName(){
036          return name;
037      }
038      public int getHP(){
039          return HP;
040      }
041  }

051  public class Main {
052      public static void main(String args[]){
053          Character character = new Character("オーンスタイン", 100);
054          int power = 30;
055          System.out.printf("%s の現在の HP は  %d  です。¥n", character.getName(), character.getHP());
056
057          System.out.printf("%s は攻撃をくらった。¥n", character.getName());
058          character.hitDamage(power);
059
060          System.out.printf("%s の現在の HP は  %d  です。¥n", character.getName(), character.getHP());
061      }
062  }
```



右上のシーケンス図を埋めなさい

【練習 103】(1)のライフライン

【練習 104】(2)のライフライン

【練習 105】一つ目の矢印のついた線・記号

【練習 106】二つ目の矢印のついた線・記号

【練習 107】九つ目の矢印のついた線・記号

【練習 108】10 個目の矢印のついた線・記号

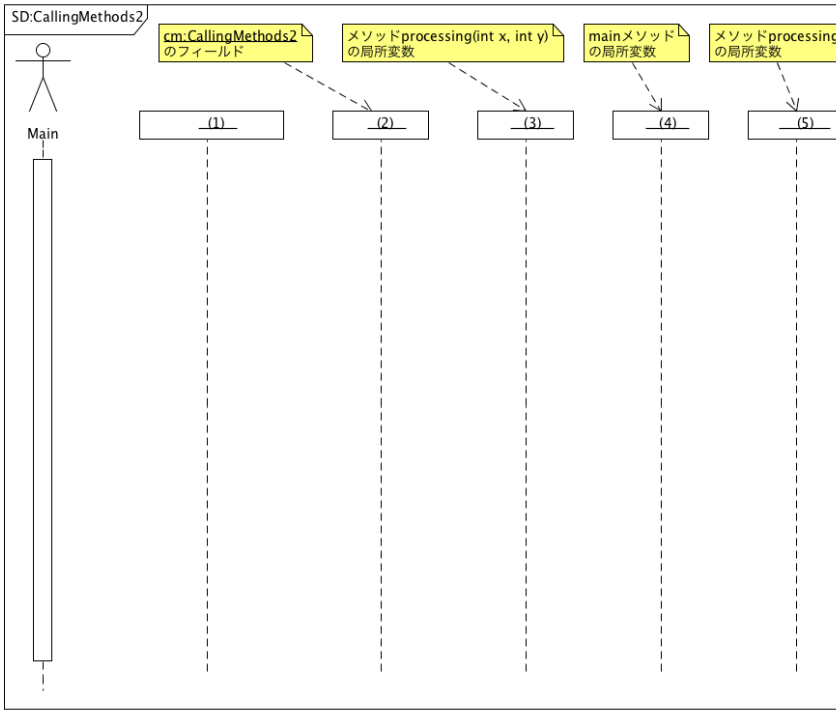
【練習 109】11 個目の矢印のついた線・記号

【練習 110】12 個目の矢印のついた線・記号

以下のソースコードについて【練習 111】から【練習 141】を解答しなさい

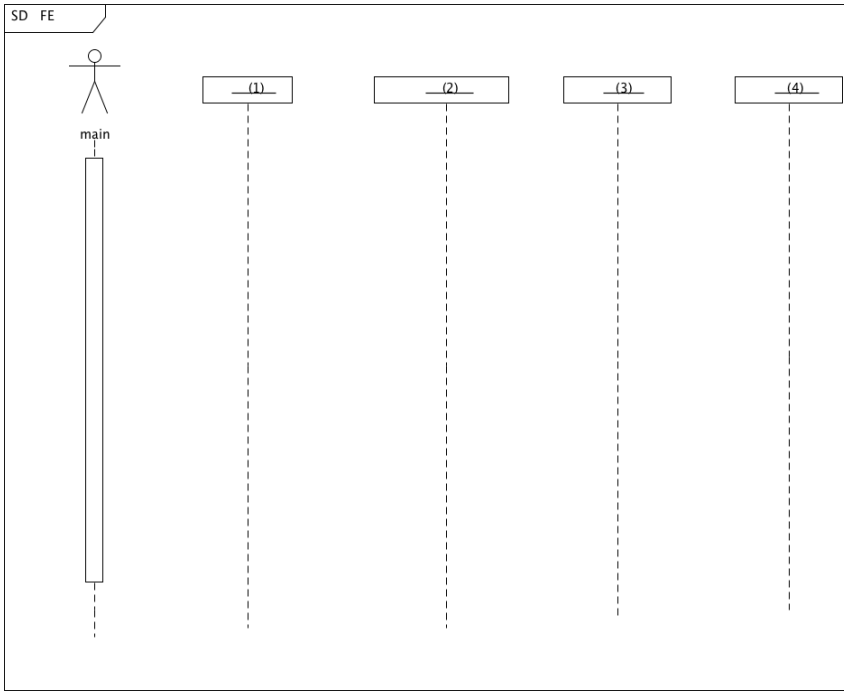
```
001  public class Point {
002      private int x;
003      private int y;
004
005      public Point(int x, int y) {
006          this.x = x;
007          this.y = y;
008          System.out.print("x");
009      }
010      public void move(int x, int y) {
011          this.x += x;
012          this.y += y;
013          System.out.print("y");
014      }
015      public void move() {
016          x = 0;
017          y = 0;
018          System.out.print("z");
019      }
020  }

031  public class CallingMethods2 {
032      private Point point1;
033
034      public CallingMethods2() {
035          point1 = new Point(1,1);
036          System.out.print("0");
037      }
038
039      public void processing() {
040          point1.move(1,2);
041          Point point2;
042          point2 = new Point(2,2);
043          point2.move();
044          System.out.print("1");
045      }
046
047      public void processing(int x, int y) {
048          Point point3;
049          point3 = new Point(x,y);
050          point1.move();
051          point3.move(x,y);
052          System.out.print("2");
053      }
054  }

061  public class Main {
062      public static void main(String[] args) {
063          CallingMethods2 cm = new CallingMethods2();
064          System.out.println("a");
065          cm.processing(4,5);
066          System.out.println("b");
067          Point point = new Point(3,3);
068          System.out.println("c");
069          cm.processing();
070          System.out.println("d");
071          point.move();
072          System.out.println("e");
073      }
074  }
```

【練習 111】出力の一行目を答えなさい
【練習 112】出力の二行目を答えなさい
【練習 113】出力の三行目を答えなさい
【練習 114】出力の四行目を答えなさい
【練習 115】出力の五行目を答えなさい
右のシーケンス図を埋めなさい
【練習 116】(1)のライフライン
【練習 117】(2)のライフライン
【練習 118】(3)のライフライン
【練習 119】(4)のライフライン
【練習 120】(5)のライフライン
【練習 121】一つ目の矢印のついた線・記号
【練習 122】二つ目の矢印のついた線・記号
【練習 123】三つ目の矢印のついた線・記号
【練習 124】四つ目の矢印のついた線・記号
【練習 125】五つ目の矢印のついた線・記号
【練習 126】六つ目の矢印のついた線・記号
【練習 127】七つ目の矢印のついた線・記号
【練習 128】八つ目の矢印のついた線・記号
【練習 129】九つ目の矢印のついた線・記号
【練習 130】10 個目の矢印のついた線・記号
【練習 131】11 個目の矢印のついた線・記号
【練習 132】12 個目の矢印のついた線・記号
【練習 133】13 個目の矢印のついた線・記号
【練習 134】14 個目の矢印のついた線・記号
【練習 135】15 個目の矢印のついた線・記号
【練習 136】16 個目の矢印のついた線・記号
【練習 137】17 個目の矢印のついた線・記号
【練習 138】18 個目の矢印のついた線・記号
【練習 139】19 個目の矢印のついた線・記号
【練習 140】20 個目の矢印のついた線・記号
【練習 141】21 個目の矢印のついた線・記号



以下のソースコードについて【練習 142】から【練習 158】を解答しなさい

```
001  public class Status {
002      private int hp;
003      private int attack;
004
005      public   Status(int hp, int attack) {
006          this.hp = hp;
007          this.attack = attack;
008      }
009      public void levelup(int hp,int attack) {
010          this.hp += hp;
011          this.attack += attack;
012      }
013      public void dead(){
014          hp = 0;
015          attack = 0;
016      }
017  }

021  public class FE {
022      private Status magician;
023
024      public FE(){
025          magician = new Status(14,6);
026      }
027
028      public void statusProcessing(){
029          magician.levelup(1,1);
030          Status knight;
031          knight = new Status(18,9);
032          knight.dead();
033      }
034  }

041  public class Main06 {
042      public static void main(String[] args) {
043          FE feIf = new FE();
044          feIf.statusProcessing();
045
046          Status farmer = new Status(10,3);
047          farmer.dead();
048      }
049  }
```



右上のシーケンス図を埋めなさい

【練習 142】(1)のライフライン
【練習 143】(2)のライフライン
【練習 144】(3)のライフライン
【練習 145】(4)のライフライン
【練習 146】一つ目の矢印のついた線・記号
【練習 147】二つ目の矢印のついた線・記号
【練習 148】三つ目の矢印のついた線・記号
【練習 149】四つ目の矢印のついた線・記号
【練習 150】五つ目の矢印のついた線・記号
【練習 151】六つ目の矢印のついた線・記号
【練習 152】七つ目の矢印のついた線・記号
【練習 153】八つ目の矢印のついた線・記号
【練習 154】九つ目の矢印のついた線・記号
【練習 155】10 個目の矢印のついた線・記号
【練習 156】11 個目の矢印のついた線・記号
【練習 157】12 個目の矢印のついた線・記号
【練習 158】13 個目の矢印のついた線・記号

以下のソースコードについて【練習 159】から【練習 185】を解答しなさい

```
001  public class Status {
002      private String status;
004      public Status(String status) {
005          this.status = status;
006      }
007      public String getStatus() {
008          return this.status;
009      }
010      public void setStatus(String item) {
011          switch (item) {                          // swith 文の条件式に String が許されるのは Java1.7 以降
013          case "Default":
014              this.status = "ちび";
015              break;
016          case "Mushroom":
017              if(status.equals("ちび")) status = "スーパー";
018              break;
019          case "FireFlower":
020              this.status = "ファイアー";
021              break;
022          case "Feather":
023              this.status = "マント";
024              break;
025          default:
026              System.out.println(item+"は定義されていません");
027          }
028      }
029  }

041  public class Mario {
042      private Status status;
043      private String agentName;
045      public Mario(String name) {              // this("ちび", name);と書いて Mario(String, String)を
047                                               //呼び出した方がすっきりすることが多い
048          status = new Status("ちび");
049          agentName = name;
050      }
051      public Mario(String status, String name) {
052          this.status = new Status(status);
053          agentName = name;
054      }
055      public void giveItem(String item) {
056          status.setStatus(item);
057      }
058      public void checkStatus() {
059          System.out.println(status.getStatus() + agentName + "です");
060      }
061      public void takeDamage() {
062          status.setStatus("Default");
063      }
064  }

071  public class Main {
072      public static void main(String[] args) {
073          Mario mario1 = new Mario("マリオ");
074          Mario mario2 = new Mario("スーパー", "マリオ");
075          mario1.checkStatus();
076          mario1.giveItem("Mushroom");
077          Mario luigi = new Mario("ルイージ");
078          mario2.checkStatus();
079          luigi.giveItem("Feather");
```

```
080                 luigi.takeDamage();
081         }
082   }
```

【練習 159】(1)のライフライン
【練習 160】(2)のライフライン
【練習 161】(3)のライフライン
【練習 162】(4)のライフライン
【練習 163】(5)のライフライン
【練習 164】(6)のライフライン
【練習 165】一つ目の矢印のついた線・記号
【練習 166】二つ目の矢印のついた線・記号
【練習 167】三つ目の矢印のついた線・記号
【練習 168】四つ目の矢印のついた線・記号
【練習 169】五つ目の矢印のついた線・記号
【練習 170】六つ目の矢印のついた線・記号
【練習 171】七つ目の矢印のついた線・記号
【練習 172】八つ目の矢印のついた線・記号
【練習 173】九つ目の矢印のついた線・記号
【練習 174】10 個目の矢印のついた線・記号
【練習 175】11 個目の矢印のついた線・記号
【練習 176】12 個目の矢印のついた線・記号
【練習 177】13 個目の矢印のついた線・記号
【練習 178】14 個目の矢印のついた線・記号
【練習 179】15 個目の矢印のついた線・記号
【練習 180】16 個目の矢印のついた線・記号
【練習 181】17 個目の矢印のついた線・記号
【練習 182】18 個目の矢印のついた線・記号
【練習 183】19 個目の矢印のついた線・記号
【練習 184】20 個目の矢印のついた線・記号
【練習 185】21 個目の矢印のついた線・記号
【練習 186】22 個目の矢印のついた線・記号
【練習 187】23 個目の矢印のついた線・記号
【練習 188】24 個目の矢印のついた線・記号
【練習 189】25 個目の矢印のついた線・記号
【練習 190】26 個目の矢印のついた線・記号

練習問題の解答

【練習 12】a

【練習 13】b

【練習 14】1c

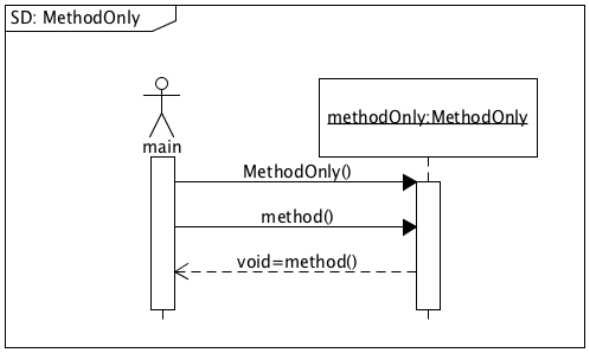
【練習 15】1d

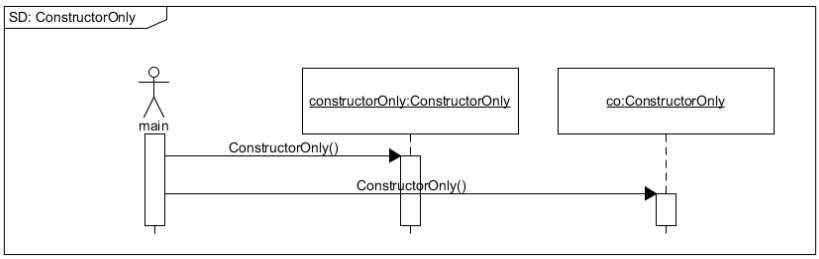
【練習 16】constructorOnly: ConstructorOnly

【練習 17】co: ConstructorOnly

【練習 18】0-1: ConstructorOnly()

【練習 19】0-2: ConstructorOnly()





オブジェクト生成（コンストラクタ呼び出し）には，処理の終了を表す戻りの矢印のついた点線はいらない

【練習 20】a

【練習 21】b

【練習 22】1c

【練習 23】methodOnly: MethodOnly

【練習 24】0-1: MethodOnly()

【練習 25】0-1: method()

【練習 26】1.0: void=method()

クラス MethodOnly にはコンストラクタが定義されていない．コストラクタが一つも定義されていない場合，『引数なしで処理がまったくないデフォルトコンストラクタが定義されている』とコンパイラは解釈する．つまり，問題文のソースコードは以下のようなソースコードとして解釈されていることに留意．

```
01    public class MethodOnly {
01a       public MethodOnly() {
01b       }
02         public void method() {
03           System.out.print("1");
04         }
05    }
```

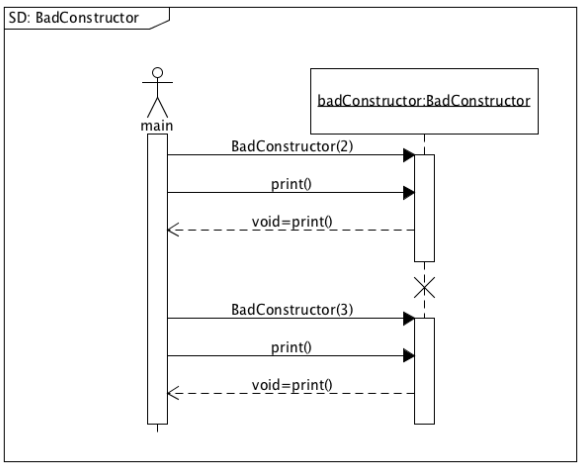
【練習 27】0a

【練習 28】0b

【練習 29】badConstructor: BadConstructor

【練習 30】0-1: BadConstructor(2)

【練習 31】0-1: print()



【練習 32】1.0: void=print()

【練習 33】1: x

【練習 34】0-1: BadConstructor(3)

【練習 35】0-1: print()

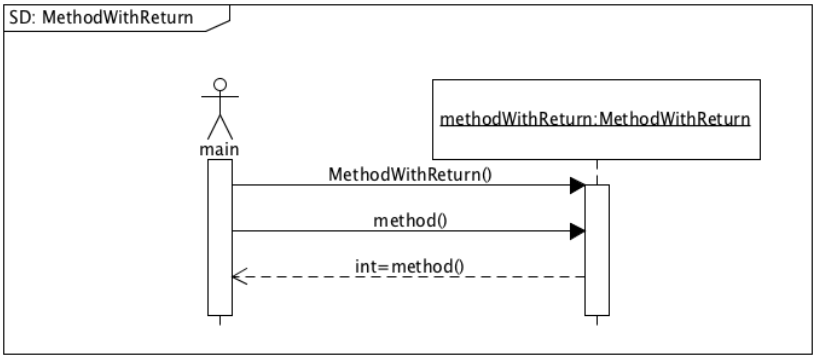
【練習 36】1.0: void=print()

【練習 37】a

【練習 38】b

【練習 39】1c

【練習 40】2



【練習 41】methodWithReturn: MethodWithReturn

【練習 42】0-1: MethodWithReturn()

【練習 43】0-1: method()

【練習 44】1.0: int=method()

問題のソースコードのように戻り値の値が明白な場合は,「1.0: int=method() :2 」のように値を付与しても良い. しかし，本講義では，いかなる場合においても戻り値の値を省略すること.

【練習 45】01a

【練習 46】23b

【練習 47】dog: Dog

【練習 48】bird: Bird

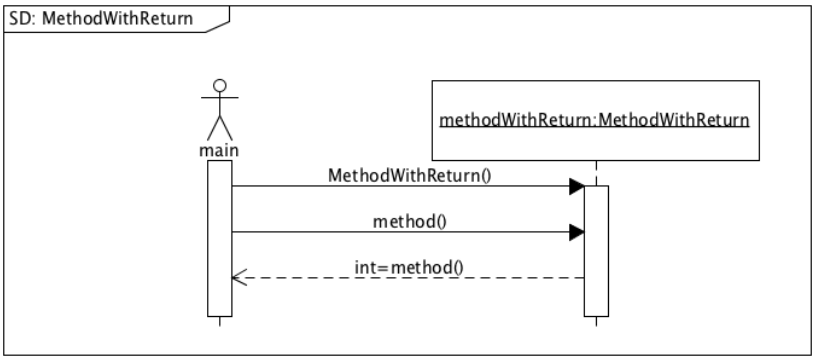
【練習 49】0-1: Dog(“ポチ”)

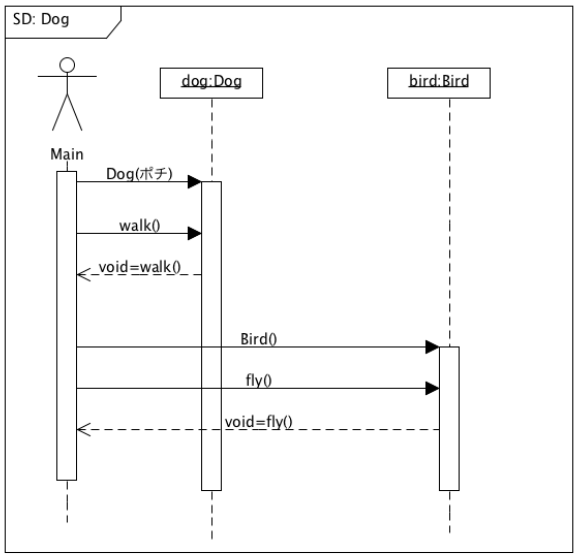
【練習 50】0-1: walk()

【練習 51】1.0: void=walk()

【練習 52】0-2: Bird()

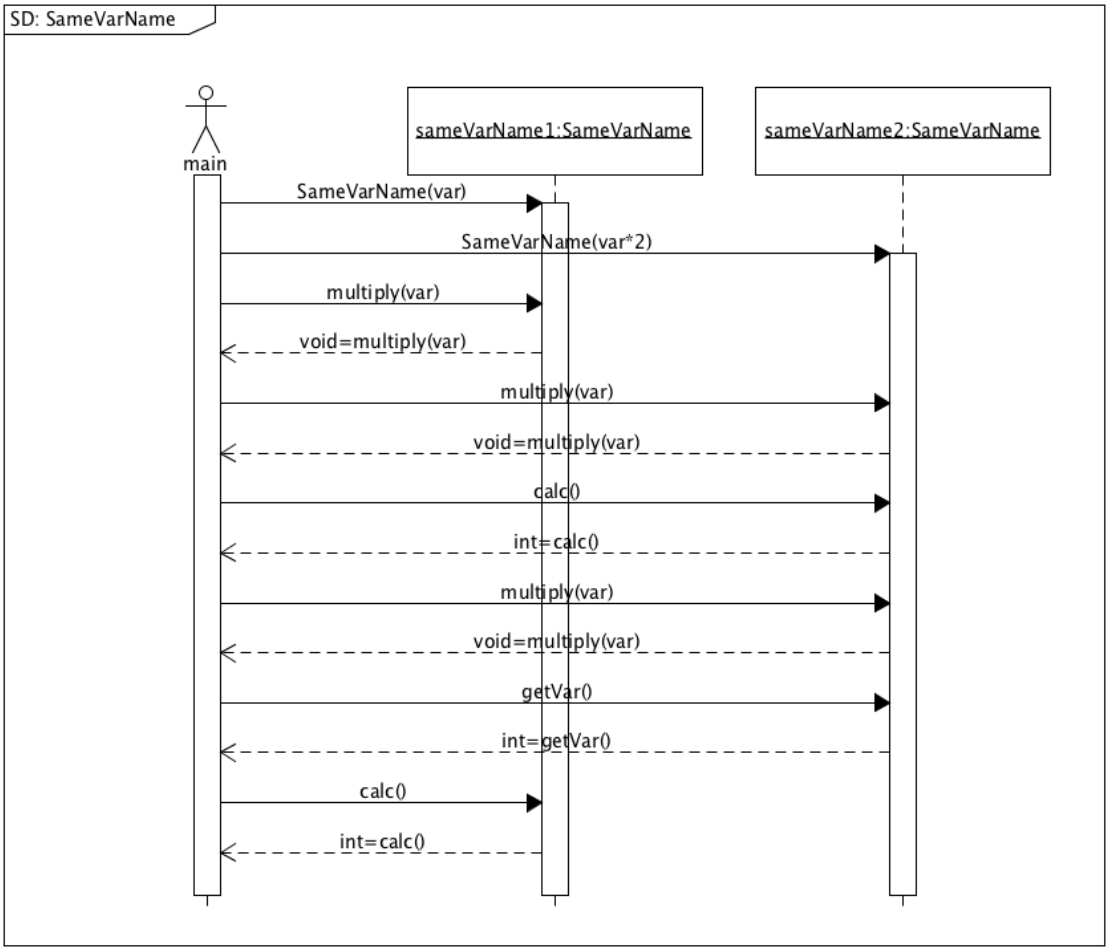
【練習 53】0-2: fly()





【練習 54】2.0: void=fly()

【練習 55】azzb



【練習 56】yyxywc

【練習 57】x6

【練習 58】10

【練習 59】80

【練習 60】sameVarName1: SameVarName

【練習 61】sameVarName2: SameVarName

【練習 62】0-1: SameVarName(var)

【練習 63】0-2: SameVarName(var*2)

【練習 64】0-1: multiply(var)

【練習 65】1.0: void=multiply(var)

【練習 66】0-2: multiply(var)

【練習 67】2.0: void=multiply(var)

【練習 68】0-2: calc()

【練習 69】2.0: int=calc()

【練習 70】0-2: multiply(var)

【練習 71】2.0: void=multiply(var)

【練習 72】0-2: getVar()

【練習 73】2.0: int=getVar()

【練習 74】0213a

【練習 75】345b

【練習 76】base: Base

【練習 77】sub: Sub

【練習 78】0-1: Base(3)

【練習 79】0-1: methodB(1)

【練習 80】1-1: methodA(k)

【練習 81】1.1: int=methodA(k)

【練習 82】1.0: int=methodB(1)

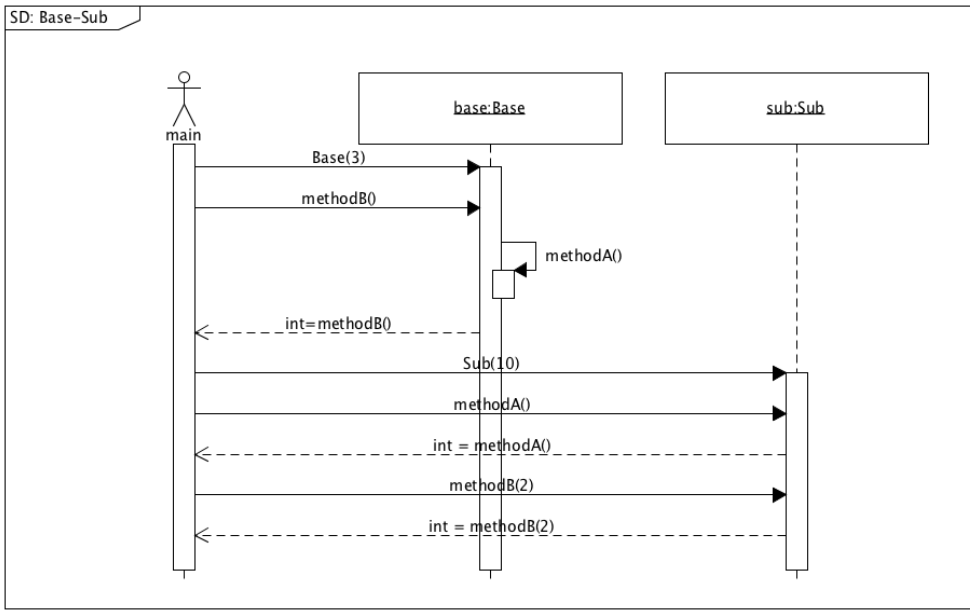
【練習 83】0-2: Sub(10)

【練習 84】0-2: methodA()

【練習 85】2.0: int=methodA()

【練習 86】0-2: methodB(2)

【練習 87】2.0: int=methodB(2)



これ以降のシーケンス図の問題は中間テストのための練習問題である．中間試験では，シーケンス図の他，これまでの確認テスト，授業中の演習問題から出題される．

【練習 88】013y4z52a

【練習 89】6b

【練習 90】sp: SequencePractice1

【練習 91】sp: SequencePractice2

【練習 92】0-1: SequencePractice1()

【練習 93】0-1: meet()

【練習 94】1-1: hello()

【練習 95】1-2: SequencePractice2()

【練習 96】1-2: talk()

【練習 97】2.1: void=talk()

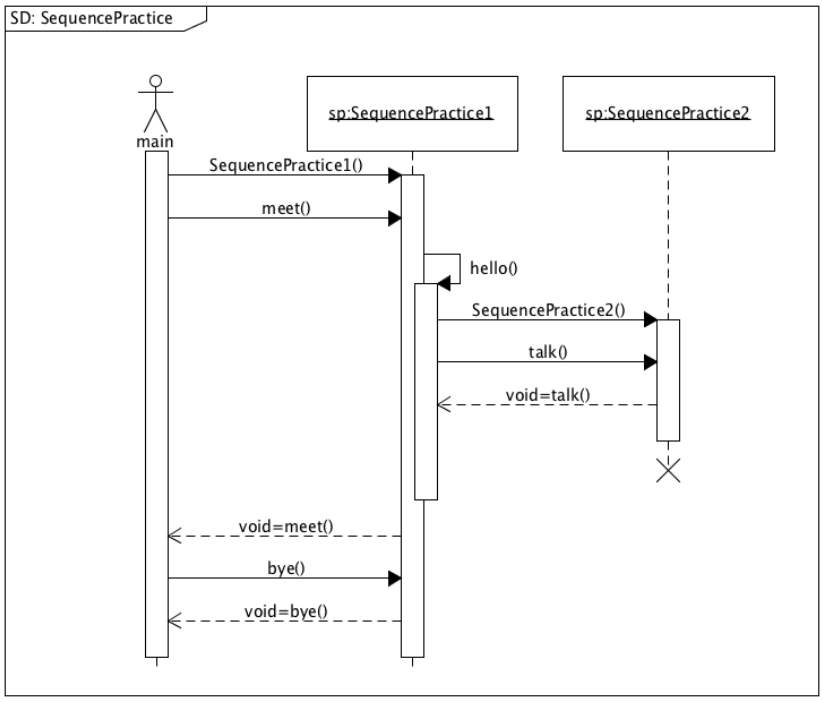
【練習 98】2: x

【練習 99】1.1: void=hello()

【練習 100】1.0: void=meet()

【練習 101】0-1: bye()

【練習 102】1.0: void=bye()



【練習 103】character: Character

【練習 104】armor: Armor

【練習 105】0-1: Character(“オースタイン”, 100)

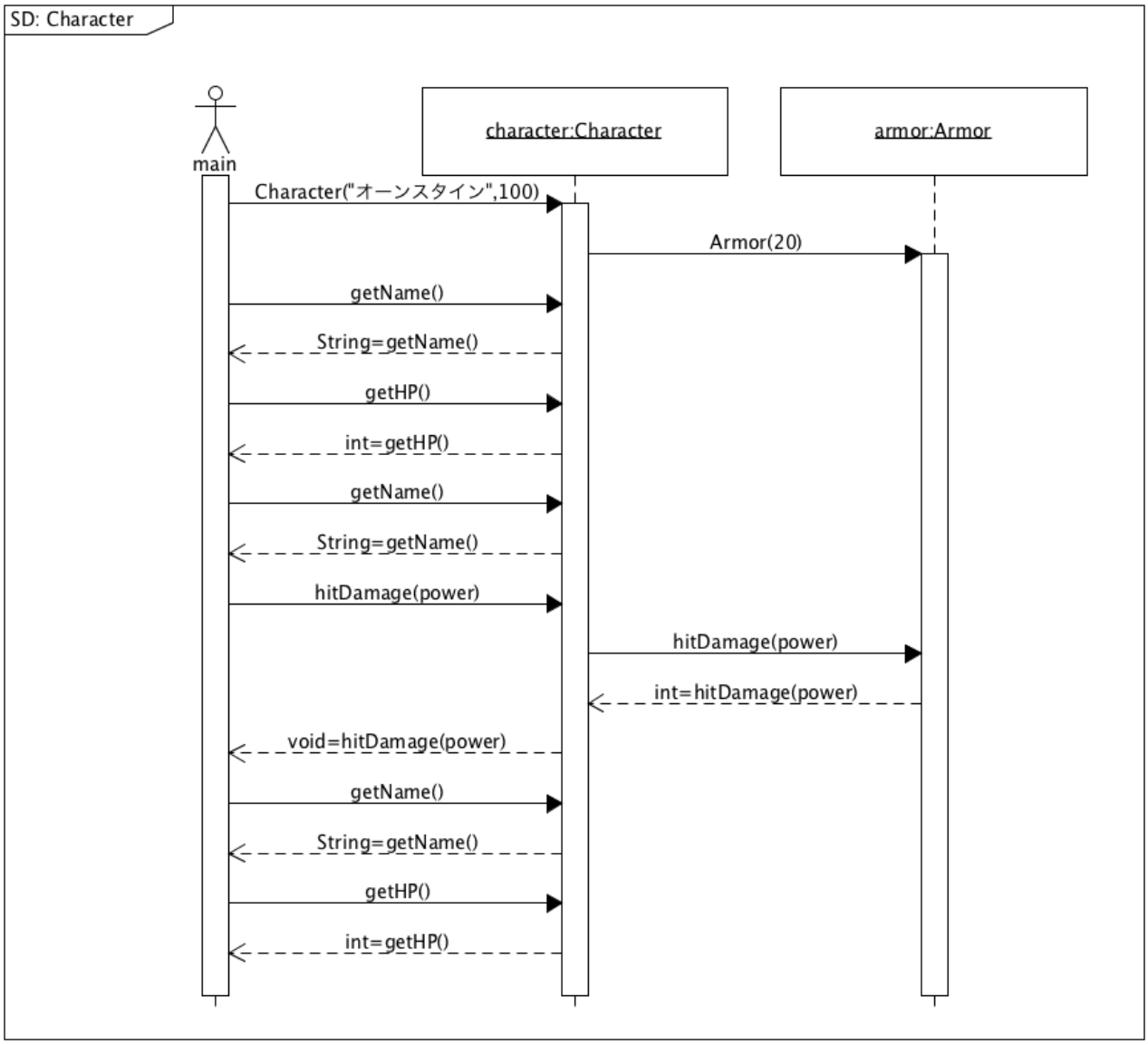
【練習 106】1-2: Armor(20)

【練習 107】0-1: hitDamage(power)

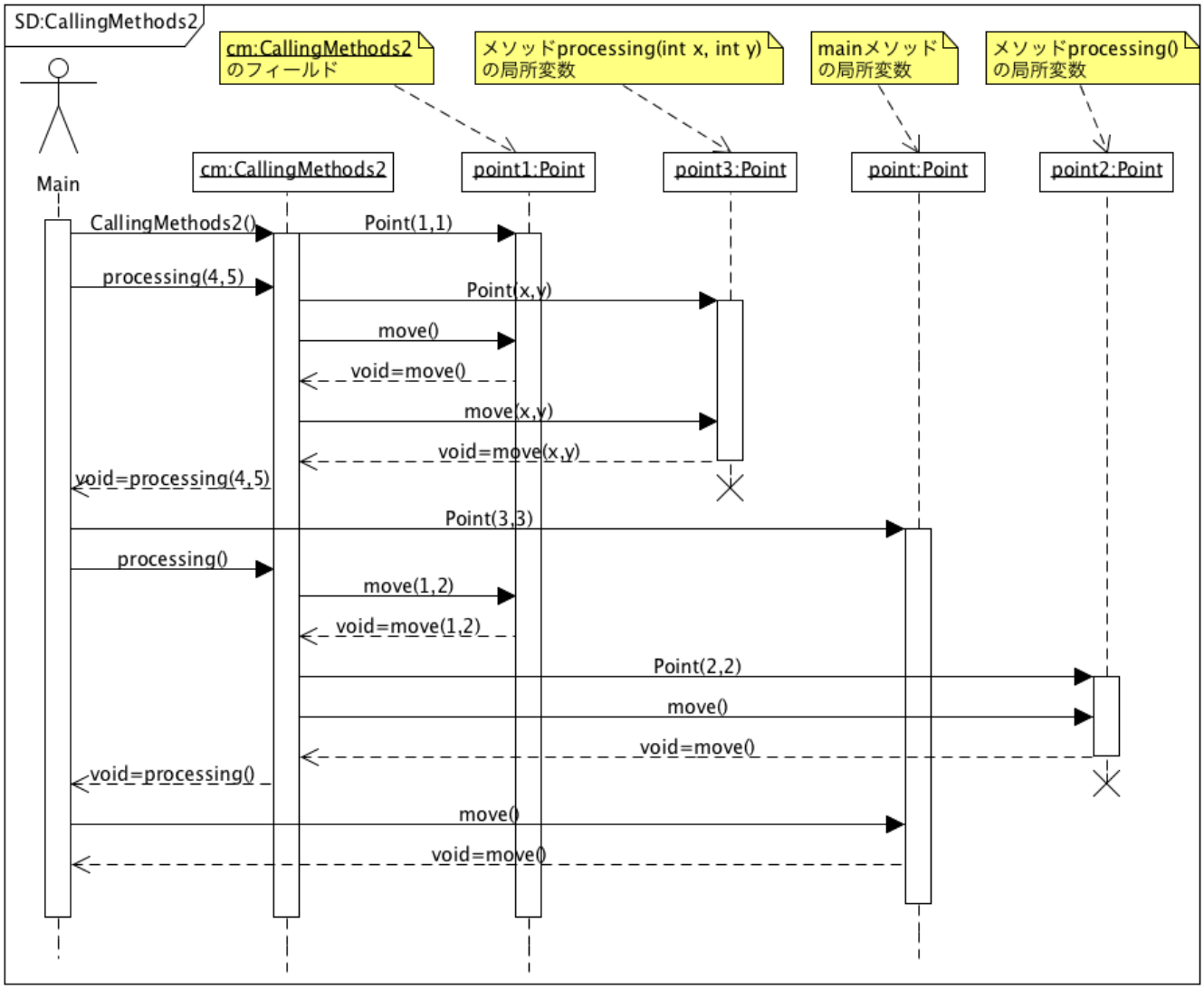
【練習 108】1-2: hitDamage(power)

【練習 109】2.1: int = hitDamage(power)

【練習 110】1.0: void = hitDamage(power)



【練習 111】x0a
【練習 112】xzy2b
【練習 113】xc
【練習 114】yxz1d
【練習 115】ze
【練習 116】cm:CallingMethods2
【練習 117】point1:Point
【練習 118】point3:Point
【練習 119】point:Point
【練習 120】point2:Point
【練習 121】0-1: CallingMethods2()
【練習 122】1-2: Point(1,1)
【練習 123】0-1: processing(4,5)
【練習 124】1-3: Point(x,y)
【練習 125】1-2: move()
【練習 126】2.1: void=move()
【練習 127】1-3: move(x,y)
【練習 128】3.1: void=move(x,y)
【練習 129】3: x
【練習 130】1.0: void=processing(4,5)
【練習 131】0-4: Point(3,3)
【練習 132】0-1: processing()
【練習 133】1-2: move(1,2)
【練習 134】2.1: void=move(1,2)
【練習 135】1-5: Point(2,2)
【練習 136】1-5: move()
【練習 137】5.1: void=move()
【練習 138】5: x
【練習 139】1.0: void=processing()
【練習 140】0-4: move()
【練習 141】4.0: void=move()



【練習 142】felf:FE

【練習 143】magician:Status

【練習 144】knight:Status

【練習 145】farmer:Status

【練習 146】0-1: FE()

【練習 147】1-2: Status(14,6)

【練習 148】0-1: statusProcessing()

【練習 149】1-2: levelup(1,1)

【練習 150】2.1:void=levelup(1,1)

【練習 151】1-3: Status(18,9)

【練習 152】1-3: dead()

【練習 153】3.1: void=dead()

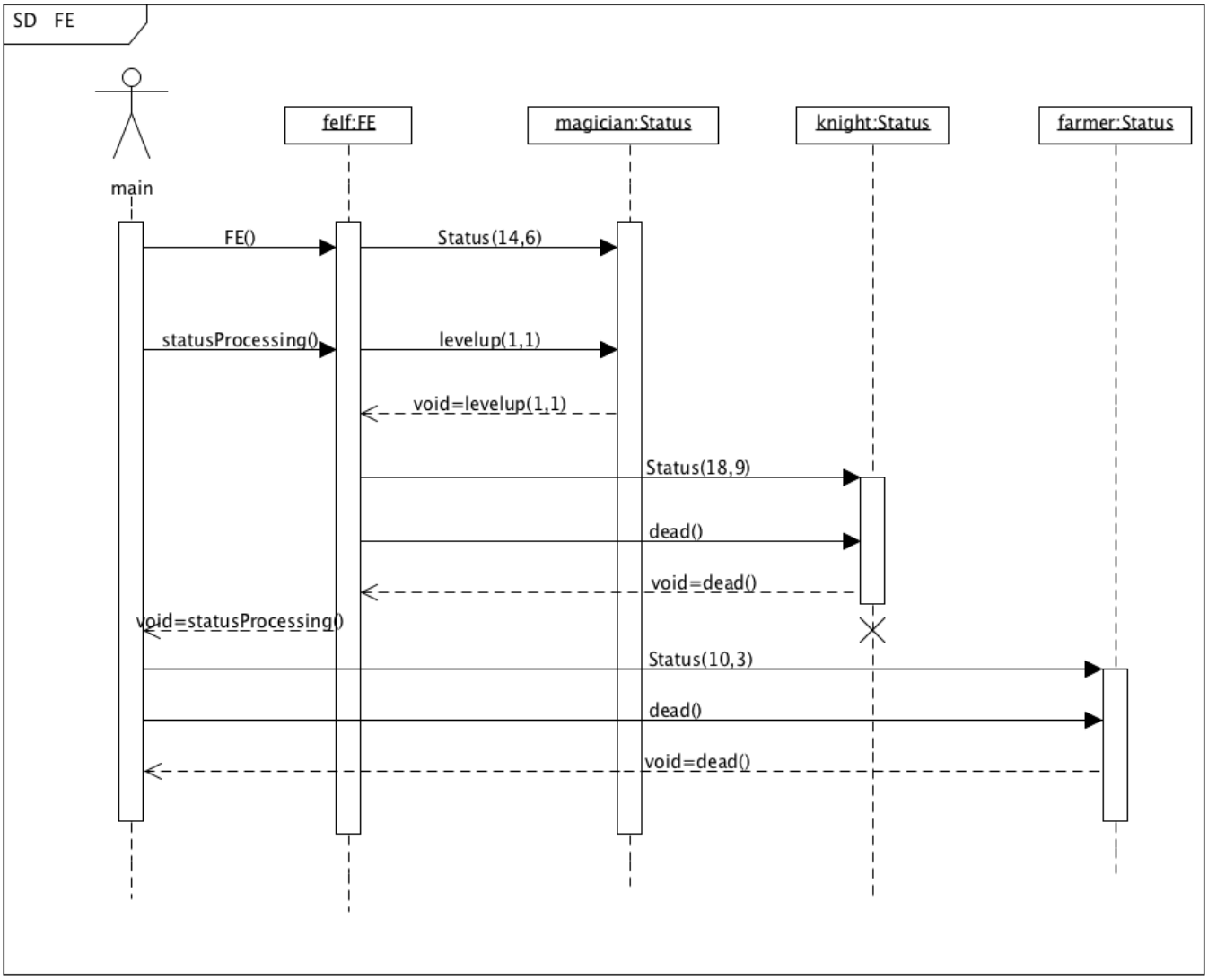
【練習 154】3: x

【練習 155】1.0: void=statusProcessing()

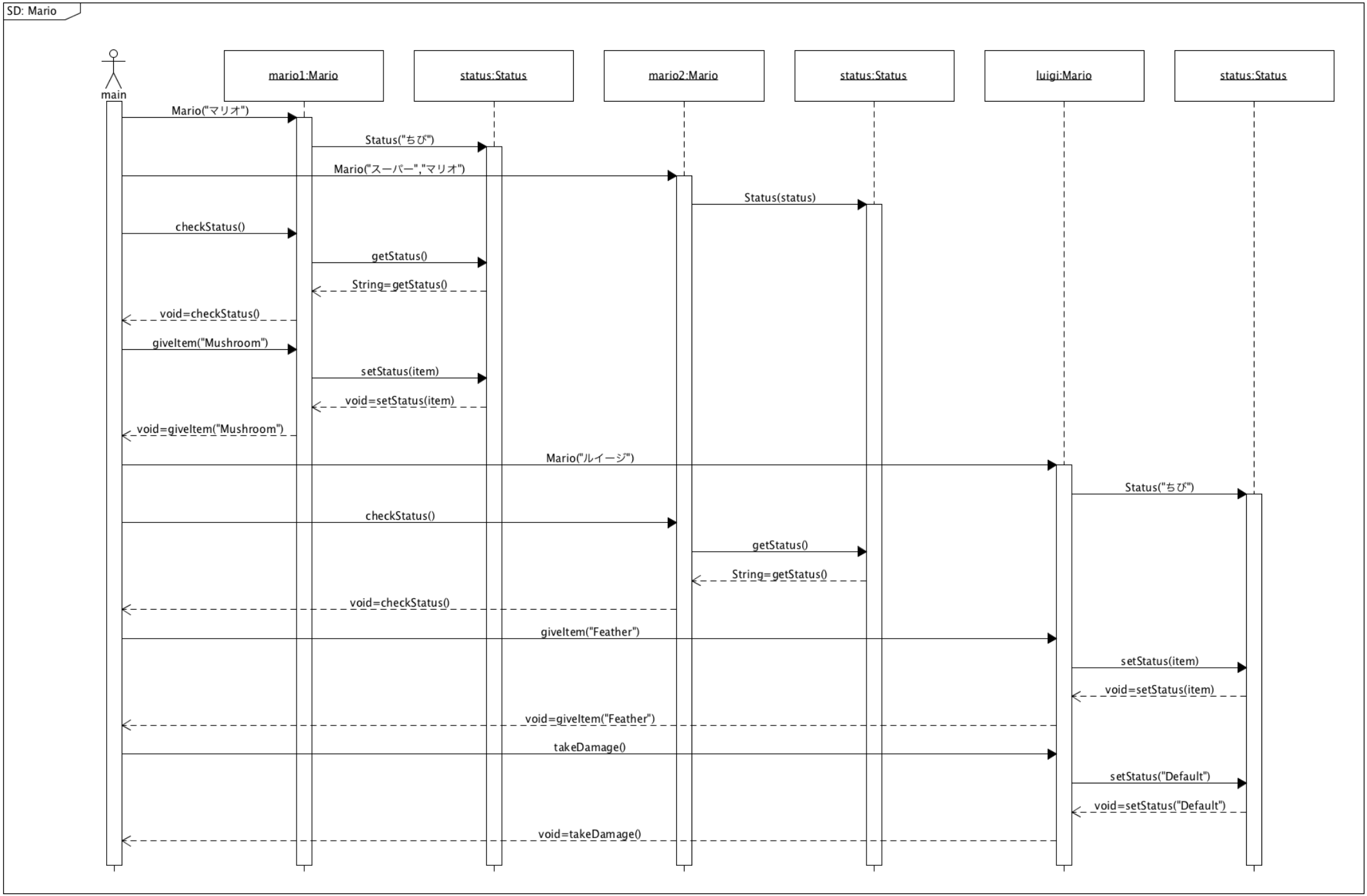
【練習 156】0-4: Status(10,3)

【練習 157】0-4: dead()

【練習 158】4.0: void=dead()



【練習 159】mario1: Mario
【練習 160】status: Status
【練習 161】mario2: Mario
【練習 162】status: Status
【練習 163】luigi: Mario
【練習 164】status: Status
【練習 165】0-1: Mario(“マリオ”)
【練習 166】1-2: Status(“ちび”)
【練習 167】0-3: Mario(“スーパー”, “マリオ”)
【練習 168】3-4: Status(status)
【練習 169】0-1: checkStatus()
【練習 170】1-2: getStatus()
【練習 171】2.1: String=getStatus()
【練習 172】1.0: void=checkStatus()
【練習 173】0-1: giveItem(“Mushroom”)
【練習 174】1-2: setStatus(item)
【練習 175】2.1: void=setStatus(item)
【練習 176】1.0: void=giveItem(“Mushroom”)
【練習 177】0-5: Mario(“ルイージ”)
【練習 178】5-6: Status(“ちび”)
【練習 179】0-3: checkStatus()
【練習 180】3-4: getStatus()
【練習 181】4.3: String=getStatus()
【練習 182】3.0: void=checkStatus()
【練習 183】0-5: giveItem(“Feather”)
【練習 184】5-6: setStatus(item)
【練習 185】6.5: void=setStatus(item)
【練習 186】5.0: void=giveItem(“Feather”)
【練習 187】0-5: takeDamage()
【練習 188】5-6: setStatus(“Default”)
【練習 189】6.5: void=setStatus(“Default”)
【練習 190】5.0: void=takeDamage()


SD: Mario
main
mario1:Mario
status:Status
mario2:Mario
status:Status
luigi:Mario
status:Status
Mario("マリオ")
Status("ちび")
Mario("スーパー","マリオ")
Status(status)
checkStatus()
getStatus()
String=getStatus()
void=checkStatus()
giveItem("Mushroom")
setStatus(item)
void=setStatus(item)
void=giveItem("Mushroom")
Mario("ルイージ")
Status("ちび")
checkStatus()
getStatus()
String=getStatus()
void=checkStatus()
giveItem("Feather")
setStatus(item)
void=setStatus(item)
void=giveItem("Feather")
takeDamage()
setStatus("Default")
void=setStatus("Default")
void=takeDamage()