

平成24年度 教科「情報」対応教員セミナー

はじめての UML

静岡産業大学情報学部

もくじ

はじめに	2
1 UML 描画ツール	3
2 オブジェクト指向	4
3 オブジェクト (Object)	5
4 クラス (Class)	8
5 UML のダイアグラム	10
(1) クラス図 (Class Diagram)	11
(2) オブジェクト図 (Object Diagram)	15
(3) ユースケース図 (Use Case Diagram)	16
(4) シーケンス図 (Sequence Diagram)	19
(5) コミュニケーション図 (Communication Diagram)	21
(6) ステートマシン図 (State Machine Diagram)	23
総合問題Ⅰ	27
総合問題Ⅱ	29
総合問題Ⅲ	32
参考文献	34

はじめに

私たちの身の回りには、メールやインターネットなどの通信システム、ネット販売やネットオークションなどの販売システム、鉄道の運行やタクシーの配車などの交通システム、更には、政治のリーダー決める選挙システムなど様々なシステムがあります。これらのシステムの仕組みや動作を説明する場合、言語で説明するよりも、図で説明する方が簡単に伝えることができる場合があります。

しかし、これらの図を描いていると、それぞれのシステムに都合の良い書き方の図になってしまい、異なる分野の相手や異なる国の人には伝わらない場合が出てきてしまいます。

そこで、それまでにばらばらだった図の書き方を精選して、国際的に統一したのが、UML (Unified Modeling Language : ユー・エム・エル : 統一モデリング言語) です。

UMLは様々なシステムを表現できますが、特にオブジェクト指向のプログラミングでシステムを開発する場合に適しているため、Java や C++ のプログラミングの普及と共に利用者也増加してきました。最近のシステム開発部門では、開発者はUMLを描くだけで、あとはUMLに従ってプログラマーがプログラムを完成する方式が増えています。

さて、UMLの重要性は何となくご理解いただけたと思いますが、いざUMLを学ぼうとすると、図の種類の多さと、オブジェクト指向の理解が、初心者の方の障害になります。

そこで、この講座ではなるべく簡単に身近にあるシステムを題材にしてUMLの理解を深め、最終的に簡単なUMLを描けるようになることを目標にしました。みなさん、目標に向かってがんばりましょう。

1 UML 描画ツール

UML の描画にはツールを使うと便利です。描画ツールには、フリーウェア、や シェアウェアがあり、更に UML から Java のソースコードを自動生成する高機能・高額なものもあります。

ここでは、Excel があれば学校でも簡単に使用できる、アドインのフリーウェアを使用します。

ソフト名 Excel で UML

配布サイト Vector <http://www.vector.co.jp/soft/winnt/business/se422771.html>

組み込み方法

- ① ソフト名で検索して、ファイルをダウンロードする。
- ② ダウンロードファイルを解凍して、“ExcelDeUML.xla” を用意する。
- ③ Excel を起動する。
- ④ (Office ボタンがあるバージョン) Office ボタンから [Office のオプション] → [アドイン] → [設定]
(Office ボタンが無いバージョン) メニューバーから [ツール] → [アドイン]
- ⑤ [アドイン] ダイアログの [参照] ボタンをクリックする。
- ⑥ 展開したディレクトリの “ExcelDeUML.xla” を選択する。
- ⑦ [OK] ボタンをクリックする。
- ⑧ メニューバーの [アドイン] のタブを選択すると、UML の各図のボタンが表示される。

使用法

- ① 描画する位置のセルを選択する。(位置はあとで自由に変更できる。)
- ② [アドイン] のメニューから図の種類を選び、さらにプルダウンメニューから要素を選びクリックする。
- ③ 図に文字を入力する。

2 オブジェクト指向

オブジェクトとは

オブジェクトとは直訳すると「もの」です。「もの」は触ることができます。例えば、「キーボード」「マウス」「机」「椅子」「ポチ（犬）」「タマ（猫）」などです。

しかし、「犬」「猫」「人」などは、抽象的な概念であり、触れることができないのでオブジェクトではありません。同じような例として、英語の定冠詞「the」と不定冠詞「a an」を思い浮かべてください。「the apple」は、実体のある「そのリンゴ」でオブジェクトですが、「an apple」は、「リンゴ：赤くて（青いもある）、丸くて、へたがあって、食べるとおいしい、・・・」という概念なのでオブジェクトではありません。（注意：実際の英語では、このように単純な例ばかりではありません。）

オブジェクト指向とは

オブジェクトは触れることができる「もの」であることは分かりましたが、「そのキーボードに

従来、ソフトウェアは、「キーボードの入力機能」「文字列の判別機能」「図形の描画機能」などの機能の集まりとして設計されてきました。機能は触ることができない非オブジェクトなので、「キーボードの入力機能は、英数字？ローマ字？多言語？」など、動作の範囲があいまいになりがちで、作成者によって機能の定義が異なるため、せっかくプログラムを作成しても他のシステムへの流用ができませんでした。

そこで、「キーボードの入力機能：英数字入力、ローマ字入力」のように機能の範囲を明確にして「こと」オブジェクトとして扱えるようにすることで、既存のプログラムを部品『「もの」オブジェクト』のように組み合わせて複雑なシステムを作成できるようになりました。

オブジェクト指向とUML

オブジェクトは具体的な「もの」「こと」なので、図つまりUMLで表すことができます。オブジェクト指向とUMLは切り離せない関係なので、UMLを理解する第1歩として、オブジェクトの表現から学んでいきましょう。

3 オブジェクト (Object)

ここでは、オブジェクトをもう少し厳密に定義していきましょう。

(1) オブジェクトは属性 (Property) を持っている。



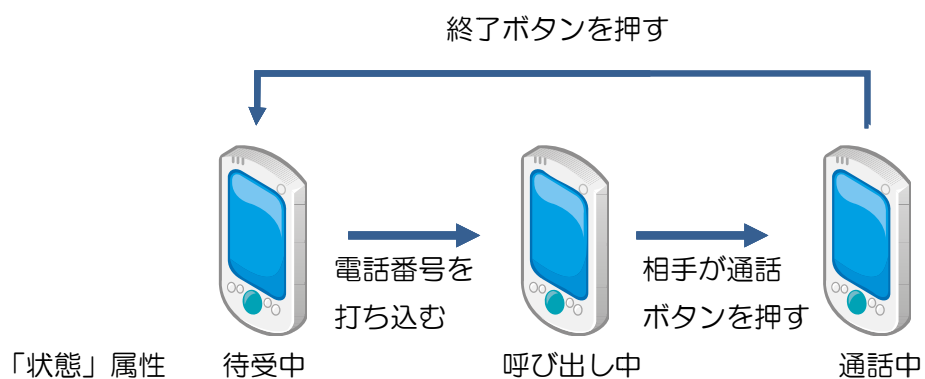
・オブジェクトは具体的に触れることができる「もの」なので固有の属性 (属性名+値) を持っています。

(2) オブジェクトは状態を持つことができる。



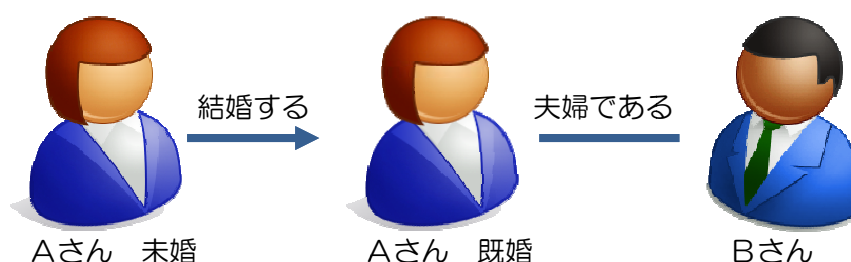
・オブジェクトは変化します。その変化は「状態」という属性で表現できます。

(3) 状態はイベントで変化する。



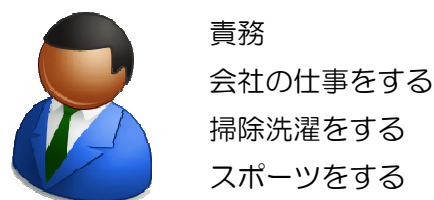
・状態を変化させるきっかけをイベントと言います。

(4) 状態は関連で表わせる。



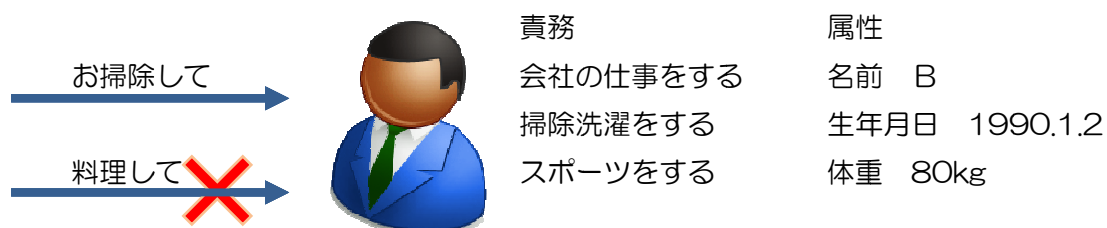
- ・「状態」の属性から考えると、Aさんは、結婚というイベントにより、状態の属性が未婚から既婚に変化したと言えます。属性の変化以外にも、「AさんとBさんは夫婦関係である」と言う関連を示すことで、Aさんが既婚である状態を表現することができます。

(5) オブジェクトは責務を持っている。



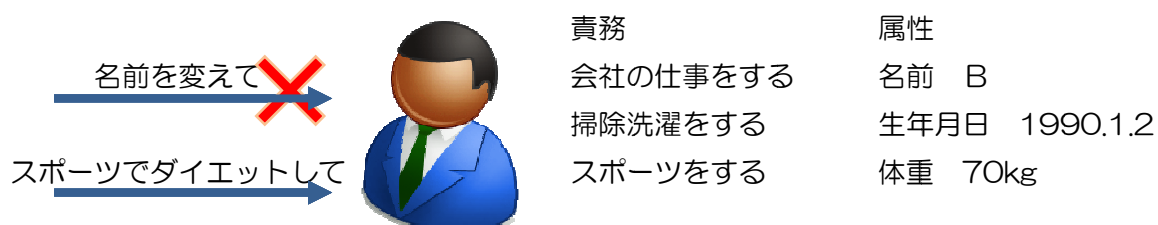
- ・オブジェクトにはそれぞれの役割があります。これを「責務」と言います。

(6) 責務を実行させるにはメッセージを送る。



- ・メッセージを送っても、責務がなければ実行しません。

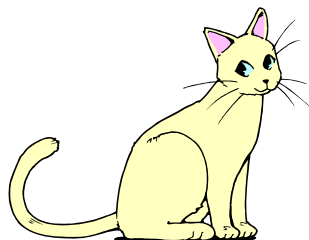
(7) 属性や状態は外部から直接操作できない。状態と責務は一体化している。



- ・名前属性は、外部から直接変更できません。体重の状態の属性は、スポーツの責務により変化します。このように、オブジェクトでは状態と責務が一体化しており、これをカプセル化と呼びます。

練習問題

- (1) ねこに3つの属性を持たせて、オブジェクトにしてください。また、責務も3つ持たせてください。



属性（属性名＋値）

例 品種 雑種

1

2

3

責務

例 えさを食べる

1

2

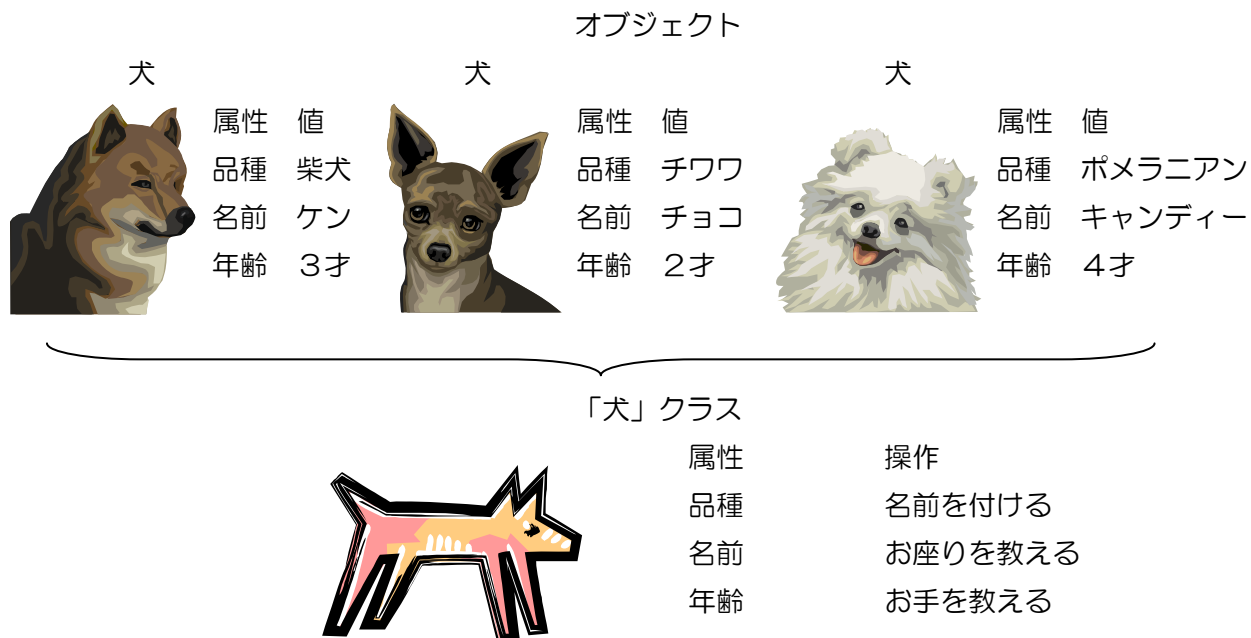
3

- (2) (1) のねこに、メッセージを送り状態を変化させます。3つの責務から一つを選び、その責務に適したメッセージを記述してください。

4 クラス (Class)

クラスは同じ属性を持つオブジェクトをまとめて抽象的に表現するものです。

(1) クラスは属性（属性名）と操作を持っている。



- ・複数の「犬」オブジェクトを見ていくと、共通の属性があることに気づき、「犬」という抽象的な概念、つまり「犬」クラスが形成されます。したがって、「犬」クラスの属性は属性名だけで、値はありません。
- ・クラスの責務は「操作」になります。操作は外部からのメッセージにより、属性値を設定したり、クラスで共通な動作をします。

(2) クラスからインスタンス（オブジェクト）を作る。



- ・「犬」クラスの属性に値を与えていき、具体的な「犬」にしたのが「インスタンス」(Instance)です。

インスタンスは具体的な「もの」なのでオブジェクトと同じ意味です。

オブジェクト = インスタンス

具体的な「もの」は、すべてオブジェクトですが、インスタンスはクラスを基として作られているので、オブジェクトを示す場合はクラスを利用した表現をします。

(3) クラスを説明する方法には内包 (Intention) と外延 (Extension) がある。

「犬」とはなんですか？

内包



ほ乳類、4本足
古代からペット
品種、寿命
えさ、・・・

外延



ケン



キャンディー



チョコ



ちびたん

- ・「犬」クラス、つまり「犬」の概念を説明する方法は2つあります。

- ① 犬の定義を説明する方法で、「内包」と言います。
- ② 犬の具体例を列挙して行く方法で、「外延」と言います。

練習問題

(1) 「自動車」という抽象概念を属性と操作で定義して、「自動車」クラスを作ってください。

「自動車」クラス



属性

例 メーカー

1

2

3

操作

例 前進

1

2

3

(2) (1) の「自動車」クラスからインスタンスを2つ作成してください。(作成するインスタンスとイラストは関係ありません。)



属性

1

2

3



属性

1

2

3

5 UMLのダイアグラム（図）

UML 2.0 では、13 種類のダイアグラムが定義されています。また、13 のダイアグラムは、「構造」を表記する図と、「振る舞い」を表記する図の2種類に分類されます。

構造に関する表記

- ・クラス図（Class Diagram）クラス間の構造を示す。
- ・オブジェクト図（Object Diagram）オブジェクト間の構造を示す。
- ・パッケージ図（Package Diagram）クラスなどをグループ化し整理された関係を表現します。
- ・コンポジット構造図（Composite Structure Diagram）クラスやコンポーネントの内部構造を示す。
- ・コンポーネント図（Component Diagram）コンポーネントの内部構造とにコンポーネント間の構造を示す。
- ・配置図（Deployment Diagram）オブジェクトの物理的な配置を示す。

振る舞いに関する表記

- ・ユースケース図（Use Case Diagram）ユーザーなどからのシステムの使われ方を示す。
- ・アクティビティ図（Activity Diagram）システムの実行における処理の流れや状態遷移を示す。
- ・ステートマシン図（State Machine Diagram）オブジェクトの状態遷移を示す。
- ・シーケンス図（Sequence Diagram）オブジェクト間の応答を時系列で示す。
- ・コミュニケーション図（Communication Diagram）オブジェクト間の関連と応答を視覚的に示す。
- ・相互作用概要図（Interaction Overview Diagram）相互作用図を要素として、より広域な処理の流れを示す。
- ・タイミング図（Timing Diagram）オブジェクトの状態遷移を時系列で示す。

たくさんの図がありますが、ここでは基本的な図である、クラス図、オブジェクト図、ユースケース図、コミュニケーション図、ステートマシン図について詳しく見ていきましょう。

(1) クラス図 (Class Diagram)

○ クラスの表記

・クラス名

【パッケージ：クラス名】

パッケージは省略可 抽象クラスはイタリックで記入します。

・属性

【可視性 名前 : 形＝初期値 {制約条件}】

名前以外は省略可能です。

クラス名
属性
操作

可視性	意味
+	public : 全てにおいて参照可能
-	private : 自クラスでのみ参照可能
#	protected : 自クラス及びその派生クラスにおいて参照可能
~	package : 同パッケージ内で参照可能

・操作

【可視性 名前 (引数の名前 : 引数の型) : 戻り値の型】

操作名のあとに () を付けます。名前以外は省略可能です。

○ クラス間関係の表現

関係	線形
関連 (association)	—————
集約 (aggregation)	◇—————
コンポジション (composition)	◆—————
依存 (dependency)	←-----
汎化 (generalization)	◁—————
実現 (realization)	◁-----

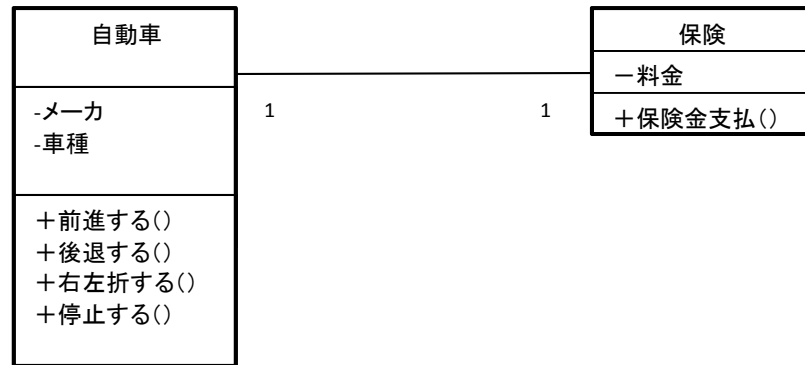
○ 多重度の表現

表記	意味
1	1 個
0..n *	0 以上
1..n	1 以上
3..7	3 から 7

○ クラス間関係の例

・ 関連 (Association)

クラス間の結びつきは、関連で表現します。



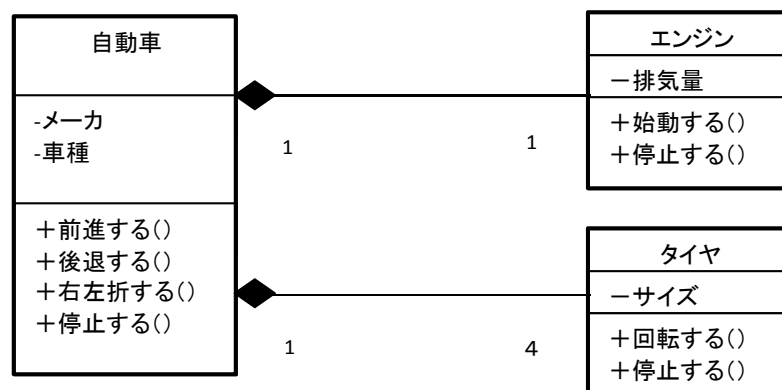
・ 集約 (Aggregation)

多数ある乗員のクラスをまとめ表現します。また、双方の間に「全体一部分」の関係があります。



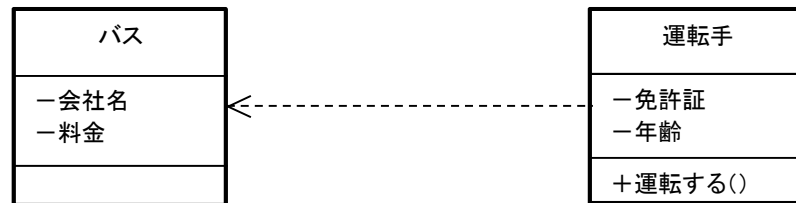
・ コンポジション (Composition)

そのクラスが無いと完成しないような、集約よりも結びつきが強い関係を表現します。



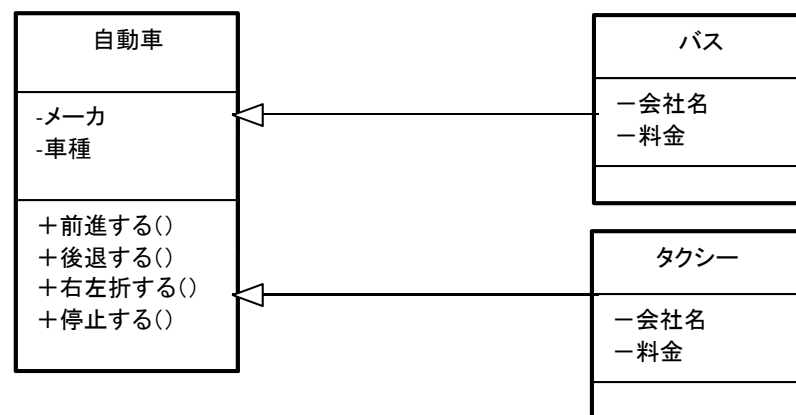
・依存 (Dependency)

相手の状態やイベントに影響を与える関係を表現します。



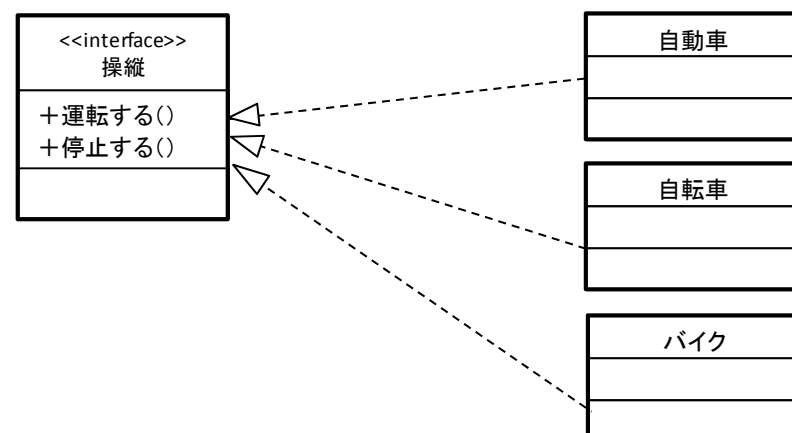
・汎化 (Generalization)

サブ (下位) クラスがスーパー (上位) クラスの一種である関係を表現します。これは「 is a 」関係とも言います。(例: The bus is a car.)



・実現 (realization)

インターフェースは、外部に公開する操作の型を集めたものです。例では、操縦というインターフェースで、運転と停止の操作ができます。そして、何を運転するのか相手を具象化する関係を表現します。



練習問題

(1) 「自転車」「ハンドル」「フレーム」「タイヤ」「サドル」「ペダル・ギア・チェーン」「ブレーキ」の各クラスを作り、その関係をクラス図に描いてください。

(2) 「たい」「いわし」「あじ」「魚」の各クラスを作り、汎化の関係のクラス図を描いてください。

(2) オブジェクト図 (Object Diagram)

○ オブジェクトの表記

・インスタンス名とクラス名

【インスタンス名：クラス名】

書式は3種類あります。

① 【 ケン 】クラスが定義されているか分からない場合② 【 ケン：犬 】通常はあまり使用しない。③ 【 ：犬 】クラスの定義がある場合

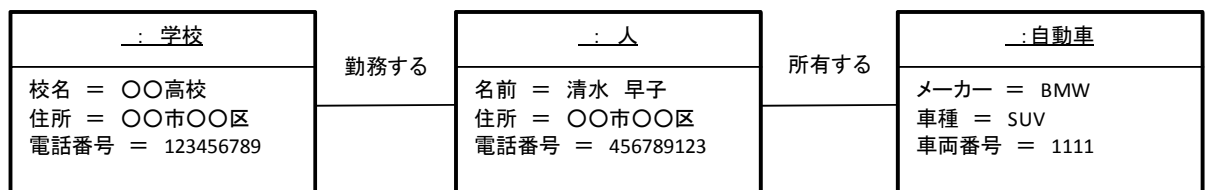
いずれも、アンダーバーを付けます。

・属性名と属性値

【属性名 = 属性値】

インスタンス名： <u>クラス名</u>
属性名 = 属性値

○ オブジェクト間関係の例



練習問題

- (1) 「野球チーム」に所属する「選手」が「携帯電話」を所有しています。各クラスからインスタンスを作り、その関係をオブジェクト図で描いてください。

(3) ユースケース図 (Use Case Diagram)

ユースケースとは、システムが外部に提供する機能です。ユーザーの視点でシステムの機能を把握するための図で、要求分析に利用されます。

○ ユースケース図の表記

- ・ アクター (Actor)

【アクター名】

システムを利用するユーザーです。また、スキャナ、センサー、アクチュエータなどの機器も外部からこのシステムを利用するアクターとして扱います。

- ・ ユースケース (Use case)

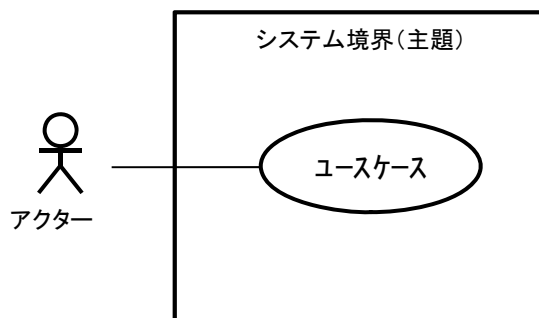
【ユースケース名】

システムに対する具体的な働きかけや命令の内容です。

- ・ システム境界 (主題)

【システム名】

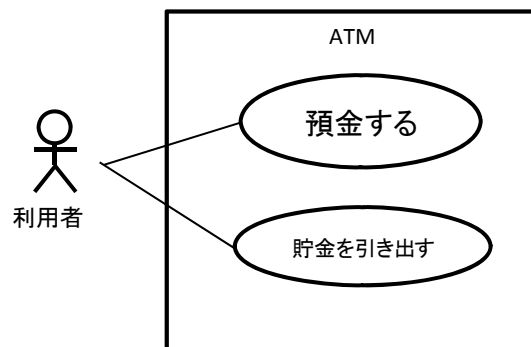
対象となるシステムの内部と外部を区別します。



○ ユースケース図の例

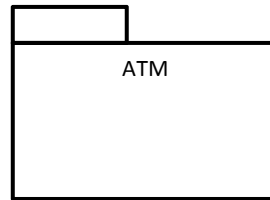
- ・ 関連

ユーザーと機能を関連付けます。



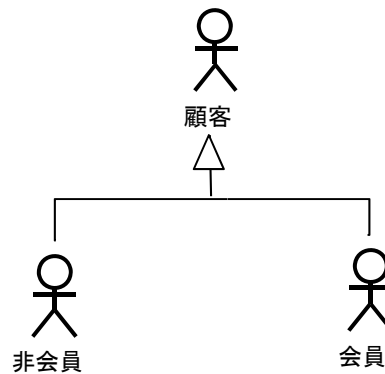
- ・パッケージ

システム境界をパッケージとしてまとめた表現です。システム単位でパッケージ化することで、図を簡素化して分かりやすくします。



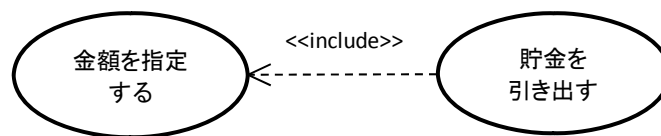
- ・汎化

ユーザーやユースケースを抽象化することで、図を分かりやすくします。



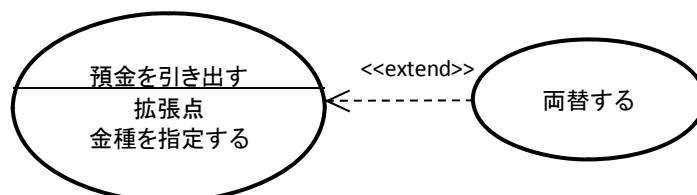
- ・包含

「貯金を引き出す」というユースケースは、「金額を指定する」というユースケースを含んでいることを示します。



- ・拡張

「預金を引き出す」というユースケースに、「金種を指定する」というユースケースを追加したことを示しています。



練習問題

(1) 次のシナリオからユーザーが複合印刷機を使用するユースケース図を描いてください。

(シナリオとは、システムの一連の動作や機能を文章で表現したものです。)

シナリオ

- ① コピーをする。
- ② コンピュータから印刷する。
- ③ FAXを送受信する。
- ④ 画像をスキャンする。

(2) デジタルカメラの機能についてシナリオを考え、ユースケース図を描いてください。

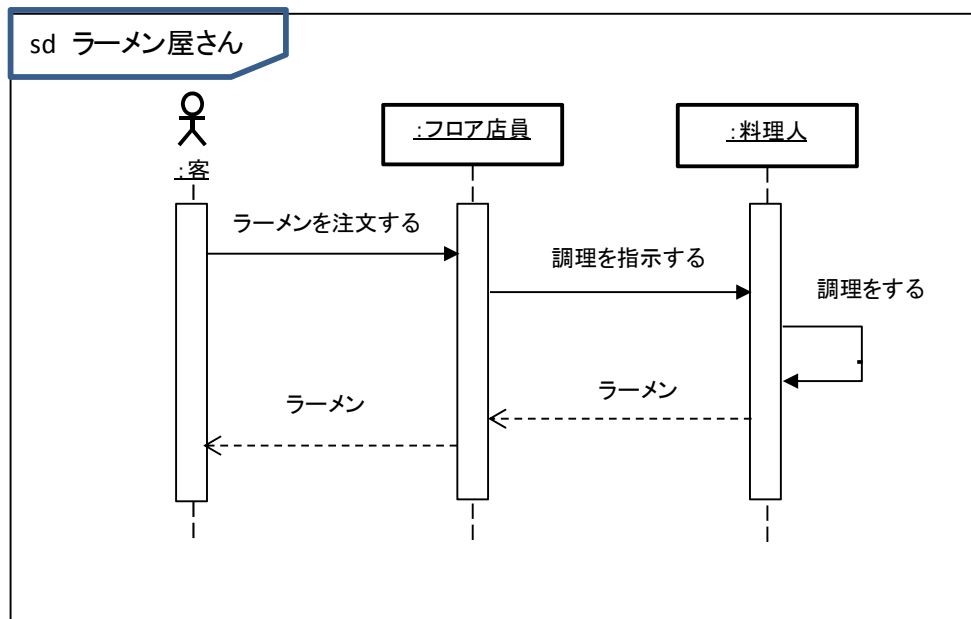
(3) 図書館の貸出返却システムについてシナリオを考え、ユースケース図を描いてください。

(4) シーケンス図 (Sequence Diagram)

シーケンス図は、ユーザーが要求する機能（ユースケース）を実現するためのオブジェクト間の相互作用（メッセージのやり取り）を時系列に表現します。システム内のオブジェクト同士がどのように振る舞うのかが明確になります。

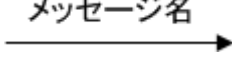
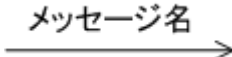
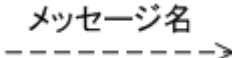
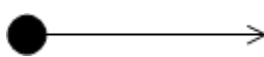
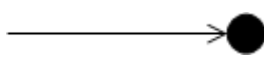
○ シーケンス図の例

ラーメン屋さんで、客が注文してから配膳されるまでの図です。



○ シーケンス図の要素

要素	表示形式	意味
相互作用 (sd)		処理単位の範囲を示し、他の図で利用する
ライフライン (Lifeline)		相互作用に参加する要素
実行仕様 (Execution Specification)		実行状態のライフラインを示す
停止 (Stop)		ライフライン自体の消滅

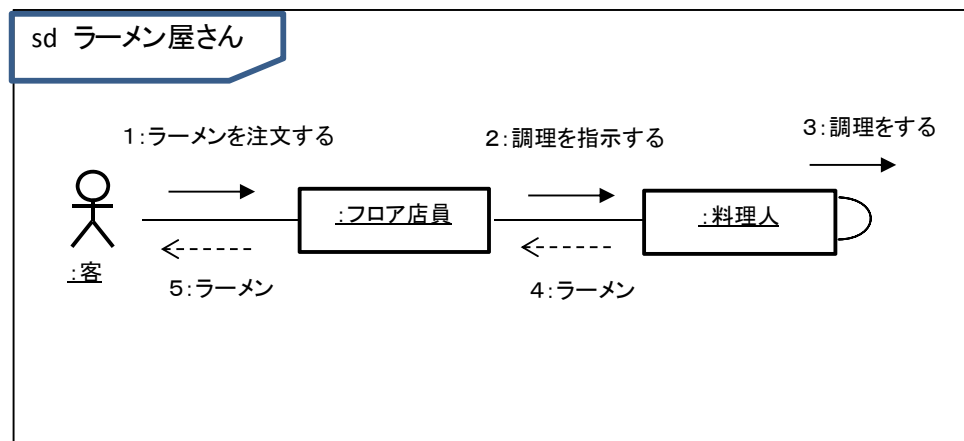
メッセージ (Message)	同期 (Synchronous) メッセージ		メッセージの処理が終わるまで次に進まない
	非同期 (Asynchronous) メッセージ		メッセージの処理が終わらなくても次に進む
	応答 (Reply) メッセージ		送り先からの戻り値
	出現 (Found) メッセージ		範囲外からのメッセージ
	消失 (Lost) メッセージ		範囲外へのメッセージ

(5) コミュニケーション図 (Communication Diagram)

コミュニケーション図は、オブジェクト間のメッセージのやり取りを示します。シーケンス図が時系列を重視しているのに対し、コミュニケーション図はオブジェクトなどの要素間の関連を重視します。なお、メッセージには必ず、実行の順番を示すシーケンス番号を記述します。

○ コミュニケーション図の例

ラーメン屋さんで、客が注文してから配膳されるまでの図です。



○ コミュニケーション図の要素

要素		表示形式	意味
相互作用 sd (範囲)			処理単位の範囲を示し、他の図で利用する
ライフライン (Lifeline)			相互作用に参加する要素
メッセージ (Message)	同期 (Synchronous) メッセージ	シーケンス番号:メッセージ名 —————→	メッセージの処理が終わるまで次に進まない
	非同期 (Asynchronous) メッセージ	シーケンス番号:メッセージ名 —————>	メッセージの処理が終わらなくても次に進む
	応答 (Reply) メッセージ	シーケンス番号:メッセージ名 ----->	送り先からの戻り値

練習問題

(1) 次のシナリオは Web ページで画像を検索して閲覧するまでの仕組みです。このシナリオからシーケンス図を描いてください。

シナリオ

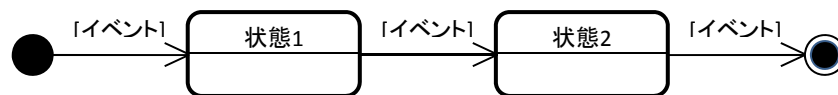
- ① 登場するオブジェクトは、ユーザー、ブラウザ、Web サーバー、画像サーバー、です。
- ② ユーザーはブラウザ画面から、閲覧したい画像のリンクボタンを押しました。
- ③ ブラウザはその画像を探すように画像のファイル名を Web サーバーに伝えます。
- ④ Web サーバーは画像サーバーから該当する画像ファイルを探します。
- ⑤ 見つかった画像ファイルは Web サーバーで閲覧可能な状態にします。
- ⑥ ブラウザは Web サーバーからの画像ファイルを再生します。

(2) (1) のシナリオからコミュニケーション図を描いてください。

(6) ステートマシン図 (State Machine Diagram)





ステートマシン図では、あるオブジェクトに着目して、状態の遷移とその振る舞いについて表現します。状態名はどのような状態であるか理解しやすいように、英語ならば「～ing」日本語ならば「～中」「～済」のようにします。

○ ステートマシン図の表記



○ ステートマシン図の要素

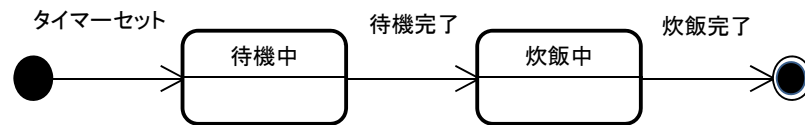
要素	表示形式	意味
開始状態(begin state)		状態遷移の開始
終了状態(sink state)		状態遷移の完了
状態(state)		実体の状態(機能は省略可能)
遷移(transaction)		状態の遷移を示す矢印。契機、条件、効果はそれぞれ省略可能。
選択点(conditional pseudo state)		前の状態が、遷移条件を示すガードに合致する方へ状態遷移する。複数の条件に合致した場合は、その遷移先は任意となり、合致するものが無い場合は記述の誤りとみなす。
連結点(junction)		二つ以上の遷移が集約される状態、または、二つ以上の遷移が離散する状態、もしくはその両方が起こる状態を表す。
コンポジット状態(composite state)		合成状態の略式表記
入場点(entry state)		合成状態の入り口を表す。

退場点 (exit state)		合成状態の出口を表す。
停止状態 (terminate node)		停止状態は、状態マシン図で定義されている事象そのものが完了することを表す。
浅い履歴状態 (history, shallow pseudo state)		以前状態に戻ることを表す。浅い履歴状態では、形式的に戻ることを意味しており、実際に遷移前と遷移後とで以前の状態に変化があっても考慮しない。
深い履歴状態 (history, deep pseudo state)		以前状態にもどることを表す。深い履歴状態では、実際に遷移前と同じ状態に戻る。

○ ステートマシン図の例

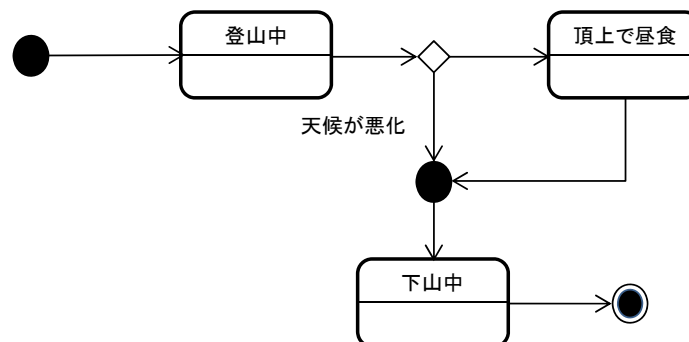
・基本構成

希望の時間にご飯を炊く、炊飯器です。



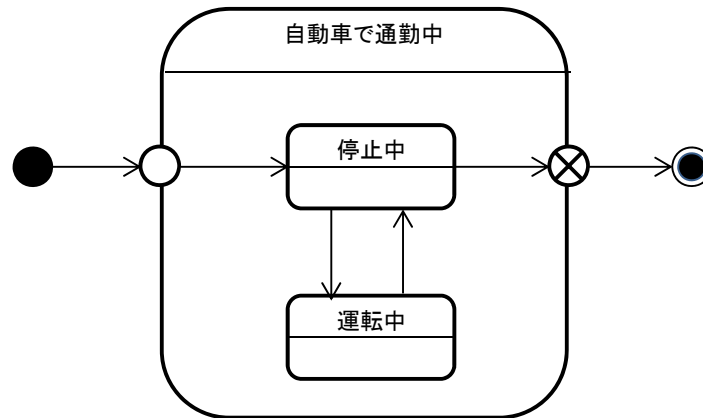
・選択点と結合点

登山中に天候が悪化したら、登頂をあきらめて下山します。



- ・入場点と退場点

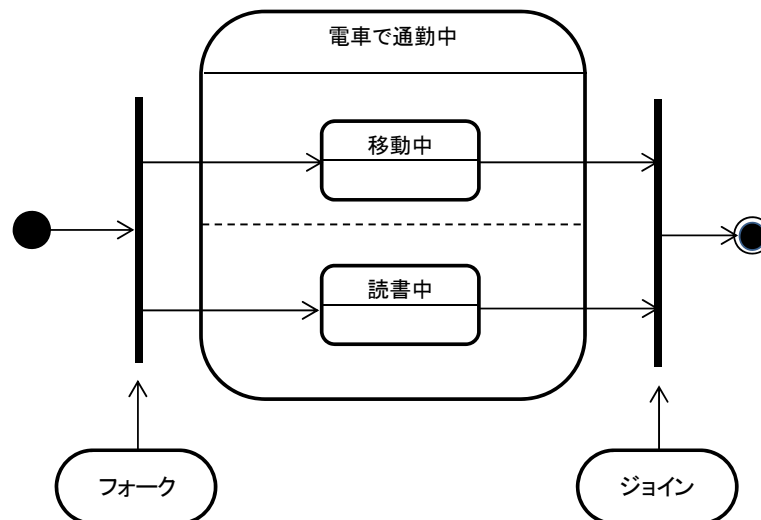
自動車で通勤中の状態を入場点と退場点で表します。この自動車で通勤中の状態の中には、停止中と運転中の状態があります。なお、自動車に乗るときと降りるときは停止中です。



- ・直交状態、フォークとジョイン

電車で通勤中は、移動する状態と読書する状態は互いに影響を及ぼさないので直交状態と言います。

二つの状態は波線で区切って表示します。また、この直交状態の始まりと終わりはフォークとジョインで表すことができます。



練習問題

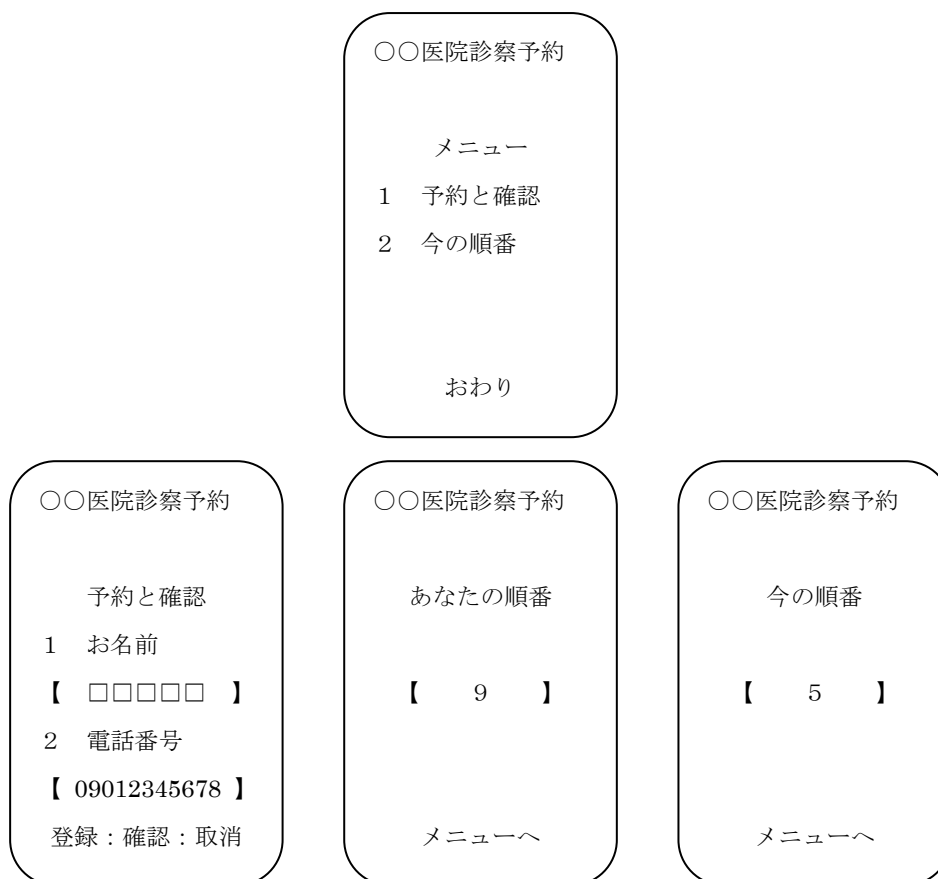
- (1) あるプリンターには「待機中」「印刷中」「用紙無し」の各状態があります。これらの状態の遷移をステートマシン図で描いてください。

総合問題Ⅰ

次の病院の診察予約システムについて、各図を作成してください。

シナリオ

携帯電話の画面は次の4画面です。



- ① 予約システムに接続すると、「メニュー」画面が表示され、各機能へリンクする。
- ② 予約をする場合は「予約と確認」画面では、名前と電話番号を入力して、登録ボタンを押す。
- ③ 登録ボタンを押すと、システムには、現在の登録者番号の次に、名前、電話番号、が記録される。そして、「あなたの順番」画面へリンクする。
- ④ 登録した自分の順番を確認する場合は、「予約と確認」画面で、名前か電話番号を入力してから確認ボタンを押すと、「あなたの順番」画面へリンクする。
- ⑤ 登録した自分の順番を取り消す場合は、「予約と確認」画面で、名前か電話番号を入力してから取消ボタンを押すと、「メニュー」画面へリンクする。
- ⑥ 「あなたの順番」画面ではあなたの登録者番号を表示する。
- ⑦ 「今の順番」画面では、現在診察中の患者の順番を表示する。診察が終わると看護師が現在の患者番号を一つ増やす。

問題 1

ユースケース図を作成してください。

ヒント アクターは2人です。

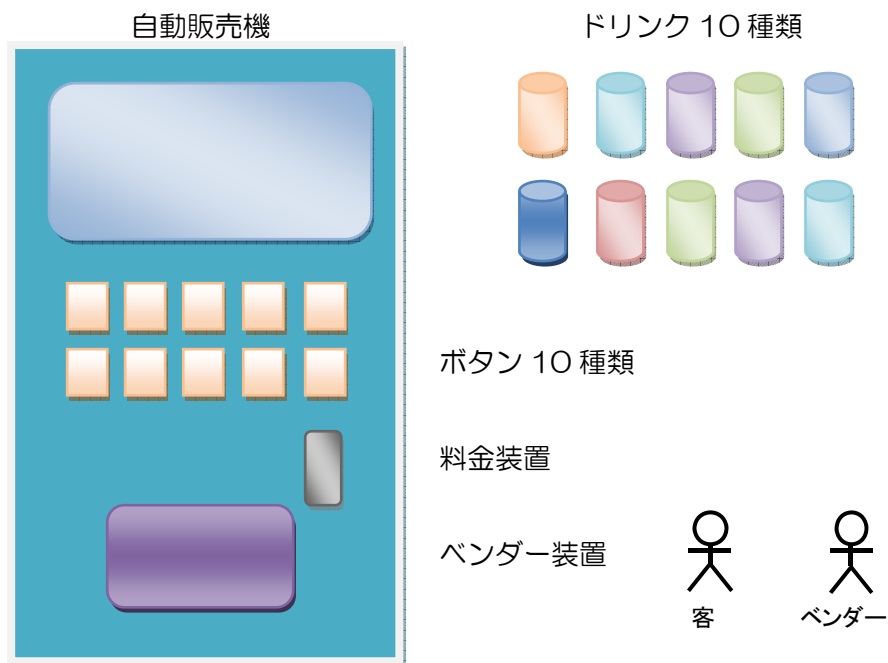
問題 2

携帯の4つの画面をオブジェクトとして、ステートマシン図を作成してください。

総合問題Ⅱ

次のソフトドリンクの自動販売機について、各図を作成してください。

シナリオ



- ① 自動販売機は、10 個のボタン、料金装置、ベンダー装置からできている。
- ② 10 種類のドリンクは、残りが半分以下になるとベンダーが運んでくる。
- ③ ②の時、ベンダーは集まった料金を回収する。
- ④ 客はお金を料金装置に入れる。（おつりは考えなくて良い）
- ⑤ ボタンを押す。
- ⑥ ベンダー装置から、ドリンクが出てくる。

問題 1

「自動販売機」「ボタン」「料金装置」「ベンダー装置」をクラスとして、クラス図を作成してください。

問題2

客とベンダーから見た自動販売機の機能を、ユースケース図で作成してください。

問題3

客がドリンクを手にするまでの手順をシーケンス図で作成してください。

問題4

客がドリンクを手にするまでの手順をコミュニケーション図で作成してください。

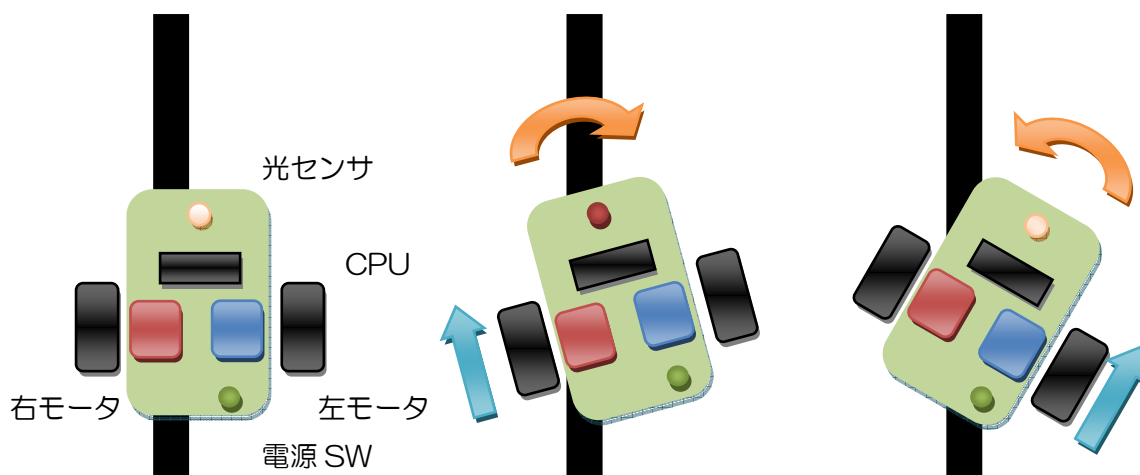
問題5

自動販売機の状態遷移をステートマシン図で作成してください。

総合問題Ⅲ

次のライントレーサーについて、次の各図を作成してください。

シナリオ



- ① このライントレーサーは、光センサー1個、左右モータ、CPU、電源SW（スイッチ）から出来ている。
- ② 電源SWを入れるとシステムが起動する。
- ③ 光センサーが黒いライン上にあると、「暗い」と判断し、右モータOFF、左モータONにする。したがって、ライントレーサーは右旋回を始める。
- ④ 光センサーがラインを外れると、「明るい」と判断し、右モータON、左モータOFFにする。したがって、ライントレーサーは左旋回を始める。
- ⑤ ライントレーサーはラインの右境界線上を進む。
- ⑥ 電源SWを切るとシステムが停止する。

問題1

「ライントレーサー」「光センサー」「右モータ」「左モータ」「CPU」「電源SW」についてクラス図を作成してください。

問題2

ユースケース図を作成してください。

ヒント ユーザーが求める機能は1つです。また、ユーザー以外にもアクターになる部品があります。

問題3

ステートマシン図を作成してください。

ヒント ① ライントレーサーが動作しているときの状態図を描きます。(状態は2つです)

② 電源 SW が ON・OFF の状態図を描きますが、ON の状態は①の図が入れ子になります。

参考文献

- 1 ゼロからわかる UML 超入門 技術評論社 2010/11/25
- 2 ゼロから学ぶソフトウェア設計 日経ソフトウェア 2002/12/30
- 3 独習 UML 第4版 テクノロジックアート 2011/4/10
- 4 UML 速習リファレンス ソフトバンククリエイティブ 2006/11/17
- 5 UML 超入門 <http://objectclub.jp/technicaldoc/uml/umlintro3>
- 6 IT 専科 UML 入門 <http://www.itsenka.com/contents/development/uml/>

初版 2012/8/11

改訂 2012/8/13

作成 高橋 等