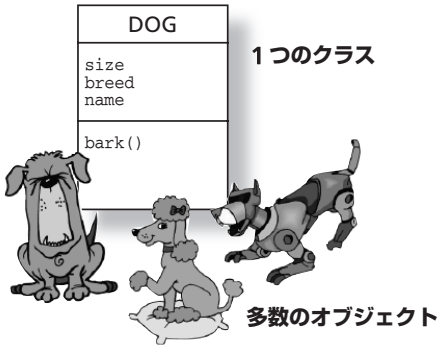
目次

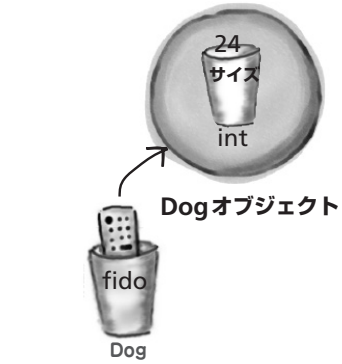
2章　クラスとオブジェクト ——オブジェクト指向の国へ 25

2章では、オブジェクト（クラス）について解説します。1章で紹介するJavaプログラムの例は、すべてmain()メソッドのみから構成されるものですが、これは正確には「オブジェクト指向」のプログラムとは言えません。オブジェクト指向でない言語のことを「手続き指向言語」と呼びますが、2章では、どのようにすれば手続き指向的ではない、オブジェクト指向的なプログラムが書けるのか、どうすればオブジェクトを作れるのかを具体的に学ぶことになります。Javaでのオブジェクト指向プログラミングの楽しさを知ってもらえればと思います。その他、この章では「クラス」、「オブジェクト」という言葉の違い、オブジェクト指向のメリットなどについても触れます。



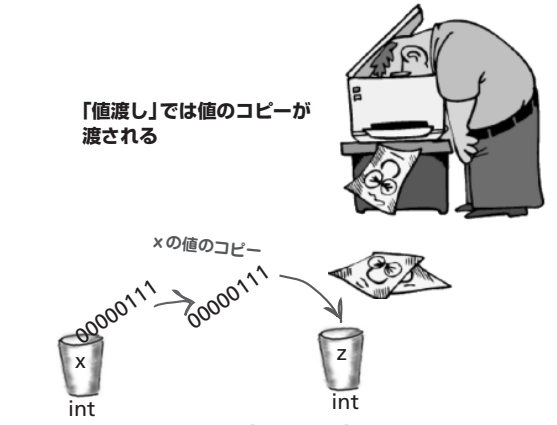
3章　プリミティブと参照 ——変数の基礎知識 47

変数にはプリミティブ型、参照型という2つの種類があります。また、参照型の変数には、配列型、文字列（String）型、クラス型などの種類があります。こうした変数を使用すれば、単なる数値だけを使用するよりも複雑なプログラムが作れます。この章では、Javaにおける変数がどのようなもので、どのように使うのか、といったことを解説します。また、「ガーベージコレクション」についても詳しく説明します。



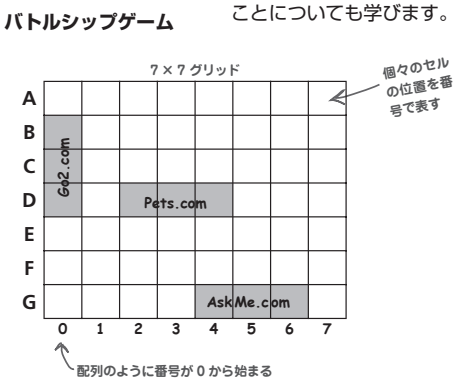
4章　メソッドとインスタンス変数 ──オブジェクトの「ふるまい」.....69

ステートは、メソッドに影響を与え、メソッドはステートに影響を与えます。オブジェクトにはインスタンス変数（ステート）とメソッド（機能）があります。4章では、インスタンス変数とメソッドの関係がどのようになっているかということを学びます。インスタンス変数の値は、同じクラスのオブジェクトでも個々に変わります。そして、インスタンス変数の値は、メソッドの機能に影響を与えるのです。メソッドがインスタンス変数の値を利用して機能するとも言えます。たとえば、weight変数の値が「5」増えるごとに、makeNoise()メソッドが出す音が少しずつ大きくなる、というようなことが起きるのです。



5章　本格的なプログラムの作成 ──本格的なプログラム（メソッド）を書く 93

この章では、実用に耐えうる本格的なプログラムを作成するのに必要ないくつかのことがらについて解説します。実用的なプログラムを書くには、演算子、ループなどを使いこなせるようになる必要があります。ループはたとえば、乱数の生成や、Stringからintへのデータ変換などに使えます。この章では、ゲーム（バトルシップゲーム）のプログラムを例にとり、本格的なプログラムをゼロから作り、テストするという作業がどのようなものになるのか、ということについても学びます。

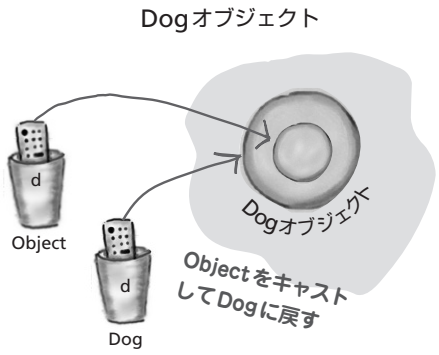


6章　Java APIの基礎 ——Javaライブラリを使用する 121

Javaの開発環境にはあらかじめ数多くのクラスが用意されています。そうしたクラスの集まりをJavaライブラリ（Java API）と呼びます。作りたいプログラムに合うクラスをJavaライブラリで見つけることができれば、自分で同じようなクラスを改めて作る必要はなくなります。その分の時間や労力を別のことに振り向けることができるのです。Java APIを使用すれば、それぞれのプログラムの、他のプログラムにない特徴的な部分だけを書けばよくなります。通常は、プログラムのかなりの部分が、Java APIの中でも特に重要な「コアライブラリ」に用意された大量のクラスを「ブロック」のように使うだけでできあがります。

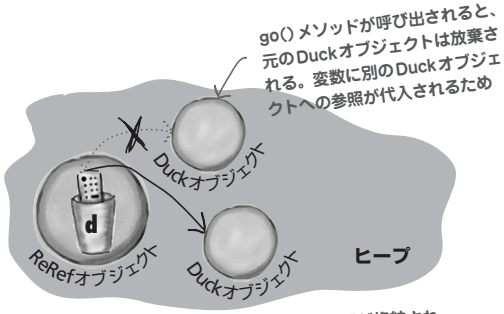
「java.utilパッケージにArrayListというクラスがあることがわかったのはよかったと思います。でも、自分の力でこのようなクラスを見つけるにはどうしたらいいのですか？」

ー Julia（31 才）、職業ハンドモデル



9章　コンストラクタとガーベージコレクション——オブジェクトの生と死 231

オブジェクトには、「生まれて、やがて死ぬ」というライフサイクルがあります。そのライフサイクルをどのようなものにするか決めるのはプログラマの仕事です。あるオブジェクトがいつ作られ、いつ削除されるか、ということをプログラマが決めるということです。ただし、オブジェクトを削除するためのコードをプログラマが直接書くわけではありません。オブジェクトを削除するのは、非情なガーベージコレクタです。この章では、オブジェクトをどのように作るか、作ったオブジェクトはどこに置かれるか、どうすればオブジェクトを効率的に消滅させることができるか、といったことについて解説します。具体的には、ヒープ、スタック、スコープ、コンストラクタ、スーパーコンストラクタなどについて詳しく触れることになります。

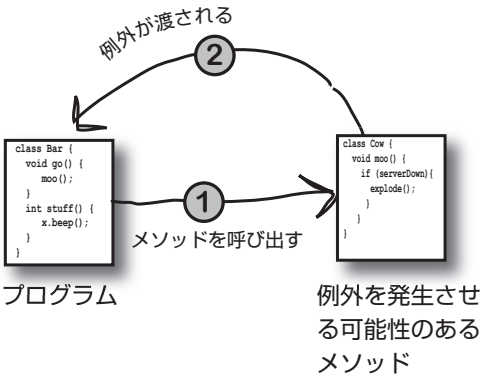


この章ではJavaにおける数値の扱いについて解説します。「数値の扱い」には、絶対値を求める、丸め（概算）を行う、2つの数値の大小を比較する、などさまざまなことがらが含まれますが、Java APIにはそうしたことを容易に行えるメソッドが用意されています。金額や日付といった数値を扱う際には、そのための独自の処理が必要です。通貨の単位を付けることや、時には小数点以下の桁数を定めることなども必要になるでしょう。日付には、国によって表示の仕方が異なるという問題があります。その他、String（文字列）から数値、あるいは数値からStringへの変換などについても知っておく必要があるでしょう。注意すべきなのは、そうしたことに使うメソッドの多くが「スタティック」なものであるということです。この章ではJavaにおいて変数（定数も含む）やメソッドがスタティックであるとはどういうことであるか、ということについても解説します。

スタティック変数は、あるクラスのすべてのインスタンスによって共有される

11章　例外処理 ——危険をともなうプログラムの作成 297

プログラムにはトラブルはつきものです。原因は、必要とされているファイルが実は存在しない、サーバがダウンしている……などさまざまです。プログラマがいかに優秀でも、あらゆる物事を完全にコントロールすることはできません。トラブルの起きそうな処理を行うプログラムを書く時には、あらかじめそのトラブルに対処するコードを組み込んでおく必要があります。ただ、問題は、トラブルが起きそうかどうかをどのように判断するか、ということです。トラブルに対処するコードをどこに入れるかということも知っておかねばなりません。この章では、JavaSound APIを利用した音楽演奏プログラムを例に、そうしたことを解説していきます。



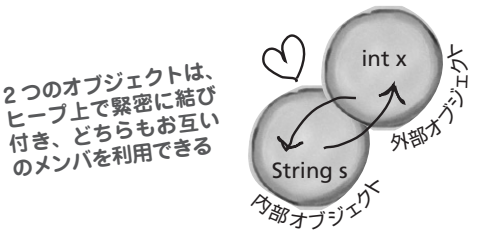
12章　GUIの基礎 ——グラフィカルなインターフェース 335

GUI（Graphical User Interface）は今や、プログラム（アプリケーション）にとって不可欠なものと言えるでしょう。たとえ、作るのがWebブラウザによって利用されるサーバサイドのプログラムであったとしても、その点に変わりはありません。12、13章では、このGUIについて、そしてイベント処理、内部クラス、といったJava言語にとって重要な概念について解説していきます。12章ではまず、画面上にボタンを作る方法を学ぶことになります。その他、画面上に絵を描く方法、JPEG画像を表示する方法、アニメーションを作る方法なども学んでいきます。


```
class MyOuterClass {

    class MyInnerClass {
      void go() {
      }
    }
}
```

外部オブジェクトと内部オブジェクトは緊密に結び付く



13章　Swingの基礎知識 ―Swingを利用する 381

Swingはさほど難しいものではありません。ウィジェットが画面上のどこに表示されるかを気にしなければ、の話ですが……。Swingを利用するコード自体は比較的簡単に書けるのですが、プログラムをコンパイルして実行してみると「あれ、なぜこれがこんなところに……」ということが起きやすいのです。実は、コードが簡単に書けるのも、ウィジェットが思いがけない位置に表示されるのも、その理由は「レイアウトマネージャ」というオブジェクトにあります。レイアウトマネージャは、少し手間をかければ思いどおりにコントロールすることができます。この章では、レイアウトマネージャをはじめ、Swingに関する基本的なことがらについて解説します。

east、west領域のコンポーネントは、横の長さを必要に応じてのばせる

north、south領域のコンポーネントは、縦の長さを必要に応じてのばせる

14章　シリアライゼーションとファイルI/O ―オブジェクトの保存 411

オブジェクトは、ファイルに保存することができます。オブジェクトは大きく「機能」と「状態（ステート）」という要素に分けられるのですが、オブジェクトの機能は、そのオブジェクトが属するクラスのコードによって決まります。一方、「状態」はオブジェクトごとに異なります。この「状態」、つまりインスタンス変数の値さえ保存すれば、いつでも同じオブジェクトを再現できます。とはいえ、インスタンス変数の値を逐一確認し、それをファイルに保存するコードを書くのは大変です。そこで、Javaには、そうしたことを自動的に（オブジェクト指向言語らしく）行う「シリアライゼーション」という機能が用意されています。

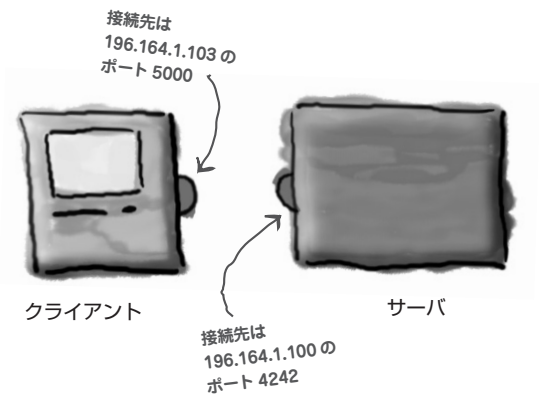
シリアライズした状態

何かご質問は？

デシリアライズ後

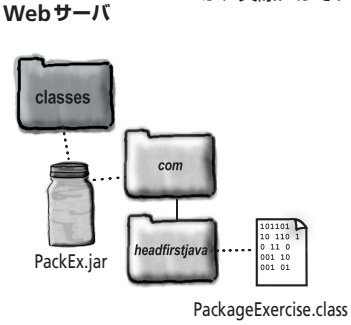
15章　ネットワークとスレッド ——ネットワーク接続のためのプログラムを書く 451

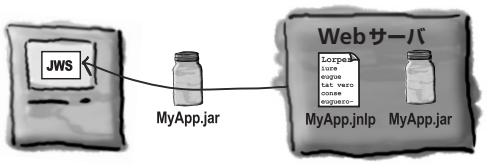
Javaではネットワーク接続のためのプログラムも簡単に書くことができます。複雑な、いわゆる「低水準の」処理は、java.netライブラリのクラスによってすべて自動的に行われるからです。その他、ネットワーク経由でのデータ送受信を単なるI/Oのように扱うことができる、というのもJavaの大きな特徴でしょう。ただ、同一マシン上のI/Oとは使用する接続ストリームが若干異なるだけです。この章では、ネットワーク経由でのデータ送受信、そして、複数の処理を同時に行うために必要な「スレッド」という概念について解説します。サンプルプログラムとして、ドラムパターンをサーバとの間でやりとりできる一種の「マルチスレッドチャットクライアント」も紹介します。



16章　プログラムの提供 ——プログラムが完成したら…… 509

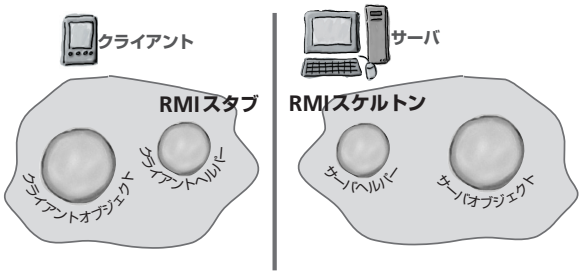
プログラムが完成したら……。何度も修正をし、さまざまな苦労を重ねて（途中、「もう、やめたい」などと言ったりしながら）素晴らしいプログラムを完成させても、人に使ってもらえなければ意味がありません。しかし、どうすれば使ってもらえるのでしょうか。16 章と 17 章では、プログラムの「パッケージング」などについて触れる他、プログラムの提供方法の種類（ローカル、リモート、その中間、など）などにも触れていきます。具体的には、実行可能JARファイル、Java Web Start、RMI、Java Web Startなどについて解説することになります。一見、難しそうですが、実際にはそれほどでもないので安心してください。





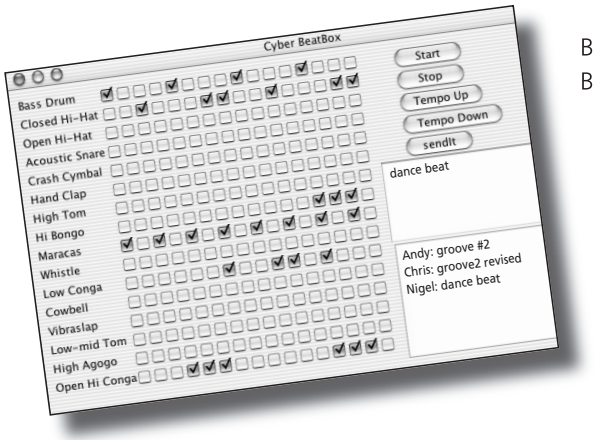
17章　RMIによるプログラムの提供 ―分散コンピューティング 535

プログラムの構成要素は一か所にまとまっていなくてはならないわけではありません。すべてが同じコンピュータ上の同じヒープにあり、使用するJVMも1つで済めば、これほど楽なことはありません。しかし、それがいつでも可能とは限らないのです。また、そうすべきでないこともあります。たとえば、プログラムは大変な処理能力を必要とするものなのに、ユーザーが携帯用の小型デバイスしか持っていないとしたらどうでしょうか。セキュリティ上の理由から、データベースアクセスをサーバ上のコードにのみ許可したい、ということもあるでしょう。この章では、主に、RMI(Remote Method Invocation)という技術について解説します。その他、サーブレットや、EJB(Enterprise Java Beans)、Jiniといった技術にも簡単に触れます。



付録A　BeatBoxプログラム完全版 577

BeatBoxプログラム完成版。付録Aには、本書の後半で紹介したBeatBoxプログラム（ドラムマシンプログラム）の完成版のコードを載せてあります。このプログラムがあればロックスターも夢ではありません。



付録B　本文で触れなかった重要事項ベスト10 587

本文で触れなかった重要事項ベスト10。本文で学んだことだけではまだ一人前のJavaプログラマとは言えません。付録Bでは、本文では触れなかった重要なことがらについて解説します。これで、この本での勉強は本当に終わりです。