発表

2023年4月21日

Unity/C#ゼミ

立教大学大学院
人工知能科学研究科
M2 星野 貴彦

1日目 はじめの一歩

- プログラミングとは何か
- プログラミングの基本的な考え方
- プログラムの実行

ビットとバイト

- ビット（bit）

→0か1のどちらかしか表現できない情報の単位

→コンピュータは0と1で情報を表現

- バイト（byte）

→ビットが8つ集まった単位

→8bit = 1byte

→2^8(=256)種類の情報を表現可能

論理演算

- 論理回路

→この回路によってコンピュータを構成

→論理値の「真」と「偽」、2進法の「0」と「1」を
　電圧の強弱で表現

AND演算

- AND演算

→英語のANDと同様、「かつ」という意味

A（入力）	B（入力）	X（出力）
0	0	0
1	0	0
0	1	0
1	1	1

→AとB両方1のときのみ、Xは1

OR演算

- OR演算

→英語のORと同様、「～または」という意味

A（入力）	B（入力）	X（出力）
0	0	0
1	0	1
0	1	1
1	1	1

→AとBの少なくとも一方が1なら、Xは1

NOT演算

- NOT演算

→0と1をひっくり返す

A（入力）	X（出力）
0	1
1	0

→入力が0なら、出力は1
　入力が1なら、出力は0

論理演算の例

- 宝箱の例

0（=False:閉じた状態）

1（=True:開いた状態）

例1：3つすべて開いているか

- 3つの宝箱がすべて開いているか判定

A=1

AND

B=1

AND

C=1

A AND B AND C
= (A AND B) AND C
=1 AND C
=1

例2：AかBの一方とCが開いているか

- A, Bの少なくとも一方の宝箱が開き、Cが開いているか判定

OR

AND

A=1　　B=0　　C=1

A OR B AND C
= (1 OR 0) AND C
=1 AND C
=1
→C=0、またはA=0に変更すると演算は偽（=0）

追加：簡単なビット演算

- 複数の宝箱の開閉状況を1変数で保持

→Box=011(2進数で表記) とし、
3bitの情報を保持する例

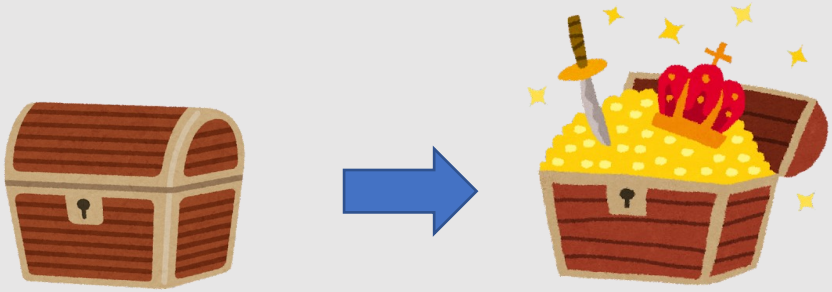
追加：Aの宝箱を開く

- Aを開きたい

→Aの情報を保持するbitに1をOR

→開けたい宝箱の情報を保持するbitを
　Open=100とする

→Box OR Open
　=011 OR 100
　=111



追加：Cの宝箱を閉じる

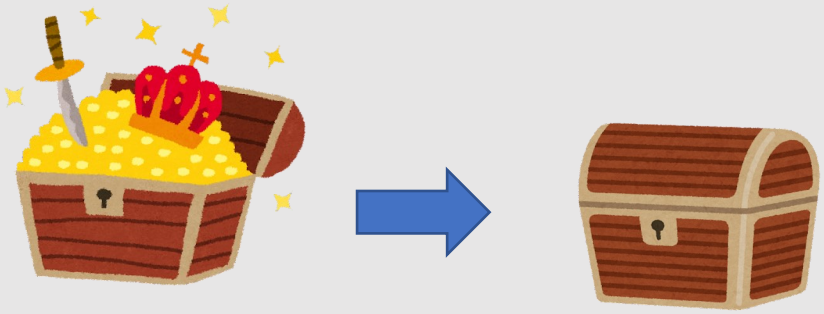
- Cを閉じたい

→閉じたい宝箱の情報を保持するbitを
　Close=001とする

→閉じたいので、CloseをNOTで否定

→NOT Close = 110
　これをAND演算

→Box AND (NOT Close)
　=011 AND (NOT 001)
　=011 AND 110
　=010



→Close=011とすれば、Bも同時に閉じることが可能

コンパイラとインタープリタ

- 高級言語

→人間が理解できる言語

→コンピュータには理解不能

→コンピュータに理解させるには、
　これをマシン言語に変換することが必要

コンパイラとインタープリタ

- コンパイラ
→プログラム全体を機械語に変換して実行
→コンパイル（変換）に時間がかかる
→実行速度は速い

- インタープリタ
→プログラムを機械語に変換しながら実行
→コンパイルは不要だが、実行速度は遅い

C#の特徴

- C#の特徴

→コンパイラとインタープリタの2つの側面を持つ
→コンパイラは共通中間言語（CIL）を生成
→CILを.NET Frameworkという実行環境で動作
→.NET Frameworkは、CILをインタープリタで生成

→ハードウェアやOSが異なる場合も、
同様にアプリケーションが実行可能
→ハードウェアを直接制御するには不向き

ネームスペース

- ネームスペース（名前空間）

→プログラムを分類するための名前を決定

例）namespace Sample101

→Sample101と呼ばれるネームスペースに属す

クラス

- クラス

→C#では、プログラムのまとまった処理の単位

→C#はオブジェクト指向のプログラミング言語

例）class クラス名

エントリーポイント

- エントリーポイント

→プログラムを実行し、最初に実行する箇所

→static void Main(string[] args)に続く{}内の処理

→1つのプロジェクトに、1つのみ

usingとネームスペース

- using ディレクティブ

→プログラムの中で指定したネームスペースを
　利用するという宣言

例）using ネームスペース;

→ネームスペースで定義されている
　クラス等が利用可能になる

次回

- 次回

→「2日目 変数と条件分岐」から