

第4章【学習19】演習解答

●演習1

(例)	情報システム	集める情報	提供する利便性
	緊急地震速報システム	P波の波を地震計で計測	震源位置や地震規模の推定 S波到達前の早期避難

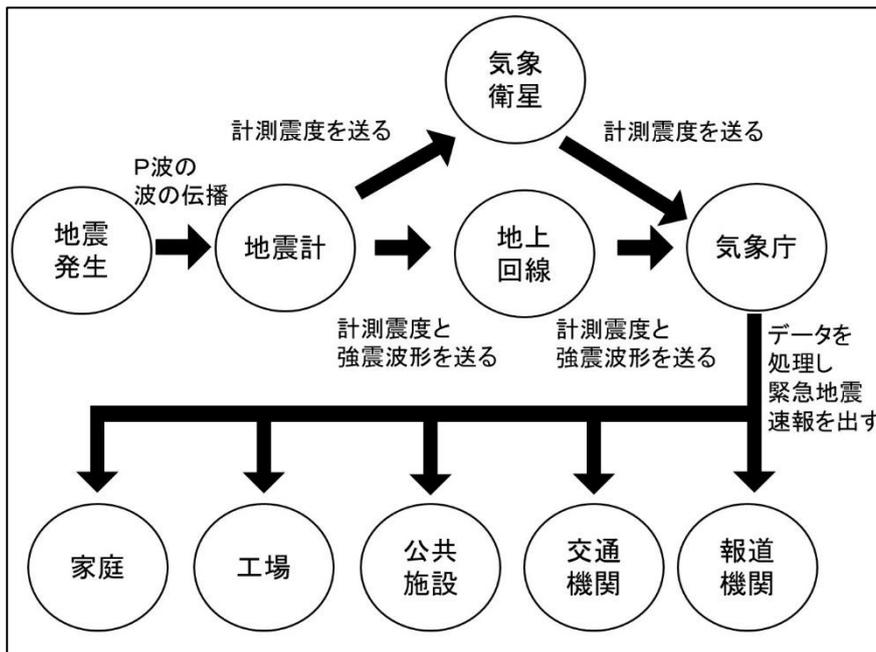
参考:国土交通省 気象庁 緊急地震速報とは

<https://www.data.jma.go.jp/svd/eew/data/nc/shikumi/whats-eew.html>

(例)	共通ポイントシステム	集める情報	提供する利便性
	ポイントカード	登録時の個人情報等 購買履歴等	誕生日クーポンなどの送付 ポイントの付与 購買履歴に応じた自店及び他店のクーポンなどの発行

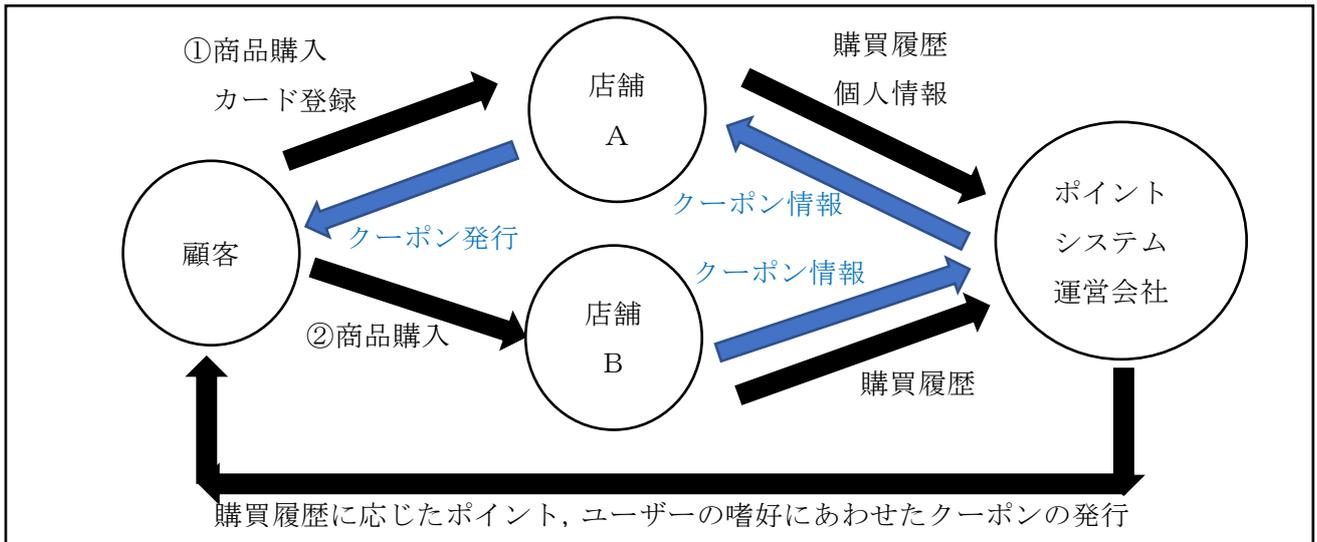
●演習2

緊急地震速報システム



緊急地震速報は、地震発生直後に強い揺れの到達時間や震度を予測し、できるだけ早く伝える情報である。地震計で観測した情報が気象衛星と地上回線の2つの経路を経て気象庁に届けられる。地上の回線が被害を受けたときを考え、気象衛星を通じて気象庁に計測震度のデータが送られている。地上回線のみ、地盤の違いなどによる揺れ方の違いなどの調査のため、強震波形を送っている。P波の後に来るS波が来る前に、地震速報を家庭などに伝達することが、この情報システムのねらいである。

共通ポイントシステム



ユーザーはポイントカード登録時に個人情報、商品を買うごとに購買履歴を店舗経由でポイントシステム運営会社に提供する。ポイントシステム運営会社は、購買履歴に応じたポイント、ユーザーの嗜好にあわせたクーポンの提供が行われる。商品を購入した店舗とは別の店舗のクーポンが商品購入時に提供される場合もある。

ポイントシステムの加盟店が多くなると、よりユーザーの利便性が増すとともに店舗の売り上げにも貢献し、ポイントシステム運営会社も手数料、匿名化された顧客データの活用などで潤う。付与されたポイントの使用を通じて顧客の囲い込みもはかれるため、ユーザー獲得のためのキャンペーンが絶えず行われている。

●演習3

緊急地震速報は、地震発生直後のP波(毎秒 6~7km)の震度を計測し、S波(毎秒 3.5~4.5km)が到達する前に早期避難を促さないといけないため、極めて厳しい即時性が要求される。そのために、震度計で震度を常に監視するリアルタイム制御処理を行っている。

ポイントカードシステムは、緊急地震速報ほどの即時性は求められない。オンラインランザクション処理またはバッチ処理で運営されていると思われる。

●演習4

GPS による位置情報システムと、SNSによるユーザーの投稿を組み合わせることで、現在お互いのいる位置に近いユーザー間の交流を促進する新しいサービスが考えられる。

第4章【学習20】演習解答

●演習1

二次元バーコード決済システムにて不正アクセス被害が発生し、利用者の機密情報の漏洩が起こった。原因としては、ID を乗っ取る簡易手段を情報システムが提供していたこと、そしてクレジットカード情報などの重要な情報を扱っているのに2段階認証に対応していなかったことなどが挙げられる。その結果、身に覚えのない不正な取引が大量に発生し、システムを停止することになった。対策としては外部の機関にこの情報システムの評価・テストをサービス開始前に依頼し、実施しておけばトラブルは起こらなかったのではないかと考えている。

●演習2

リース会社との契約が切れ、運用していた情報システムのデータを保存していた記録媒体をフォーマットした上で専門業者に渡し、データの消去や処分を依頼したがそのデータが外部に流出してしまった。扱っていたデータには膨大な数の重要な個人情報が残っていたため、個人情報を預けていた方からの信頼を損なった。また契約書に沿って外部組織に依頼したとしても、その通りに処理されずに機密情報が流出する事例となった。対策としては、外部組織に記録媒体を渡す場合は、データを暗号化するとともに、データにアクセスできない状態にした上で、記録媒体の消去や処分を依頼するようにすることが挙げられる。

●演習3

教育システムを利用していた学習者が、授業に必要なデジタル教材を取り込む際に、教育システムの管理者用のログイン情報を学習者に与える仕組みになっていた。また、学習者用ネットワークと管理者用ネットワークが同じネットワークに存在していたため、その学習者が管理者としてログインし、学習データなどの機密情報を入手していた。このようなこともあり、機密情報に学習者がアクセスしてしまうリスクを一層回避する手立てが重要とされた。対策として、重要な個人情報を扱う教育システムについて、学習者が利用するネットワークと管理者側が利用するネットワークを分離することや、管理者用のパスワードを厳重に管理するとともに、学習者用と管理者用のネットワークを別のセグメントに分け、管理者側のネットワークからではないと管理者側の機密情報にアクセスできないようにするなどが考えられる。

●演習4

情報システムのトラブルが発生すると、社会的に大きな影響を与えると同時に、情報システムを運用している組織に大きな損失を与えることになる。情報システムのセキュリティを考えるときに、個人の情報セキュリティと組織の情報セキュリティでは扱う情報量や重要度がまったく異なる。そのため情報システムは適宜、外部機関の適切なテスト・評価を行いながら運用を考えていくことが重要である。また、長く運用されている交通機関の電子決済システムなどの身近な情報システムは、情報セキュリティがしっかりと守られている。情報セキュリティが守られた上で、初めて利用者に情報システムの恩恵が受けられるのではないだろうか。

※長期に渡りトラブルのないシステムであっても、情報セキュリティが万全とはいえない。不断の見直しが大切である。

第4章【学習21】演習解答

●演習1

業務要件定義

- ・煩雑な蔵書管理を情報システムに置き換えることで管理コストをカットする。
- ・利用者はシステムを通して、常に正しい蔵書数を確認することができる。

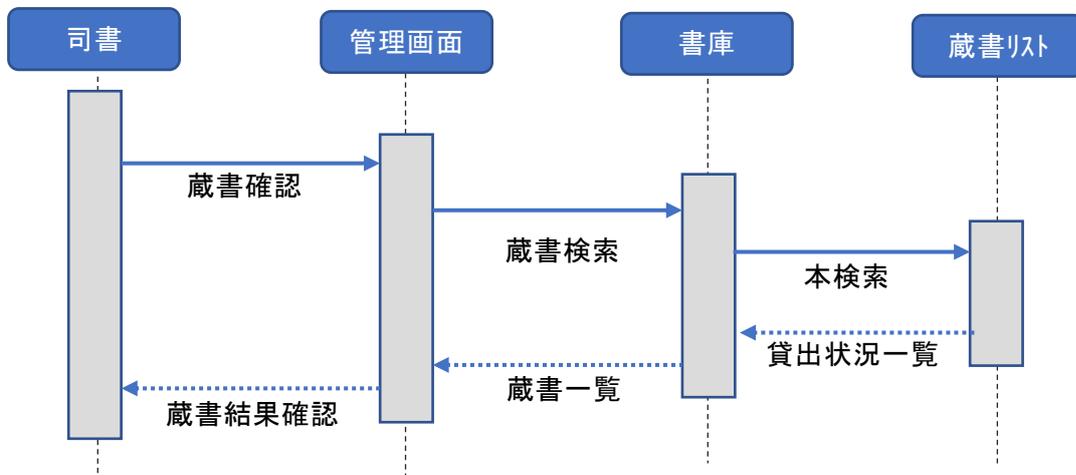
機能要件定義

- ・利用者はシステムを使って、「蔵書確認」「貸出処理」「返却処理」「貸出履歴印刷」ができる。
- ・システムは蔵書のリストをデータとして保持し、貸出処理後はデータを更新し、蔵書確認時は蔵書数を正しく表示する。

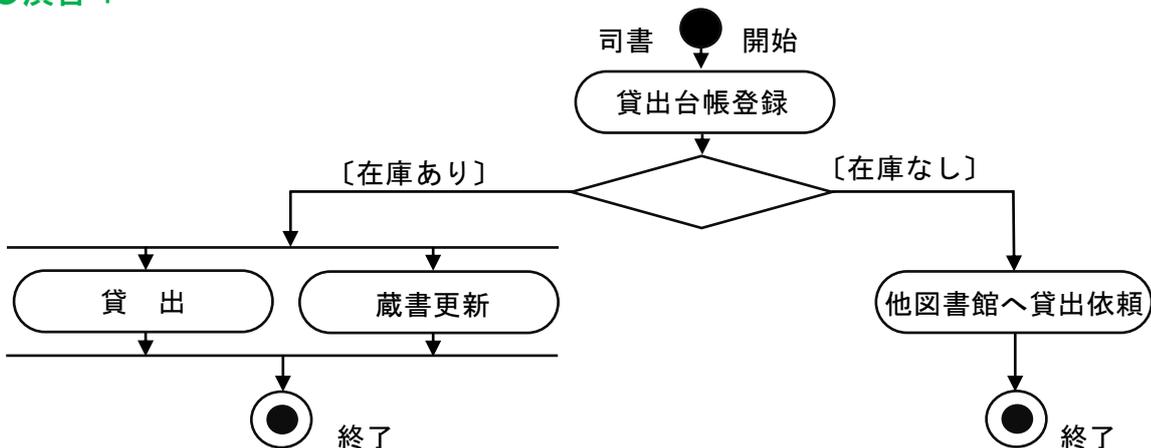
●演習2



●演習3



●演習4



第4章【学習22】演習解答

●演習1

貸し出す本があるか検索するプログラムのモジュール分割

- ・操作画面のプログラム
- ・蔵書のデータを参照するプログラム
- ・貸し出した本を記録するプログラム
- ・指定した本の状態を確認するプログラム

●演習2

蔵書のデータのカプセル化

データ1: 貸出中か否か

データ2: 返却予定日

メソッド1: 貸出中に切り替える

メソッド2: 返却したため蔵書ありに切り替える

第4章【学習23】演習解答

●演習1

プログラミング言語は、作成しようとするものに合わせて適切なものを選択する。そのためには、図表1にあるようなプログラミング言語について、ある程度はその特性を知っておいた方が良い。たとえば、図書館管理などのシステムを作成するのであれば、データベース機能が必要であり、機械の制御をするのであれば、そのような命令が充実しているものがよいなどの判断ができるようにすることが望ましい。

●演習2

```
import urllib.request, urllib.parse, json
def get(url,params):
    p=urllib.parse.urlencode(params)
    url=url+"?" +p
    with urllib.request.urlopen(url) as res:
        return json.loads(res.read().decode("utf-8"))

params = {
    "token": "test",
    "table": "book",
    "column": "author",
    "value": "クジラ飛行机",
    "format": "json"
}
url="https://api.eplang.jp/mext2/search"
data = get(url,params)
print(data)
```

●演習3

同値分割で考える。

同値クラス	テストする値の候補
延滞本がある	利用証を表示(警告表示)
すでに5冊貸出中	本を1冊貸出処理(警告表示)
すでに4冊貸出中	本を2冊貸出処理(1冊目まで貸出処理が行われ、2冊目で警告表示)
すでに3冊貸出中	本を3冊貸出処理(2冊目まで貸出処理が行われ、3冊目で警告表示)

すでに 2 冊貸出中	本を 4 冊貸出処理(3 冊目まで貸出処理が行われ, 4 冊目で警告表示)
すでに 1 冊貸出中	本を 5 冊貸出処理(4 冊目まで貸出処理が行われ, 5 冊目で警告表示)
それ以外	本を 6 冊貸出処理(5 冊目まで貸出処理が行われ, 6 冊目で警告表示)

●演習 4

演習4のリストでは, 同じ配列について, 定義した関数と Python で準備された関数で計算した値を比較している。値が一致すれば True, そうでない場合は False と表示されるので, False と表示された関数の定義を見直すことになる。

ここでは, 3つとも False となっているので, 最初の関数から順に

訂正→実行

True が表示されたら次の関数の訂正

という操作を繰り返していく。

(def_mean)

逆戻りで考える。「return sum/n」なので, sum と n にどのような値を持たなければならないかを考え, 実際の値を確認する。sum と n を出力すると, sum が正しくない値になっていることがわかる。

(def_median)

「return (ary[index1] + ary[index2]) / 2」なので, ary[index1]と ary[index2]がどのような値を持たなければならないかを考え, 実際の値を確認する。出力してみると, 本来の中間ではなく, 1つだけ大きい方へずれていることがわかる。そこから index1 と index2 が想定より 1 つずつ右にずれていることと, リストが 0 番目からカウントしていることから, どこを修正すればよいかを特定する。

(def_max)

def_max(ary)を print してみると, 2 が出てくる。最大値がほしいのに, リストの最小値が出ていることがわかる。つまり, 最大値の判定に誤りがあると考えて, 各回の判定で何が起きているかを, 実際の値と合わせて検証すると, 「これまでより小さい値が見つければ x に保持する」という逆の判定をしていることに気づくことができる。

第 4 章【学習 24】演習解答

●演習 1

表示された中から 1 冊の本を選ぶ関数(仮に `select_book()` とする)と, 選んだ本の番号(`isbn`)を指定して本を予約する関数(仮に `reserve()` とする)が用意されているとすると, 図表 1 の末尾でこれらの関数を呼び出すことで, 検索した本の一覧から 1 冊の本を選び, その本を予約することができる。

```
# 全体の実行
title=title_input()
result=title_search(title)
disp(result)
isbn=select_book()
reserve(isbn)
```

●演習 2

図書館のシステムでは, 夜中や早朝にシステム全体のバックアップを保存する。図書館の開館中は, 図書の登録や貸出返却のデータをシステムのデータベースに書き込むと同時に, 処理内容のデータをログファイルとしてデータベースとは別のディスクに記録する。システムの障害等でデータベースのデータが壊れた場合には, バックアップデータから昨夜の内容のデータベースを復元し, ログファイルから当日の処理内容をデータベースに反映する処理を行うことで, システムを障害発生直前の状態に戻すことが可能になる。

●演習 3

ブルートフォース攻撃は, 一つのアカウントに対する攻撃なので, 個別のアカウントのログイン失敗回数で制限をかければ対策できるが, リバースブルートフォース攻撃は, 複数のアカウントに対する攻撃なので, 個別のアカウントのログイン失敗回数では防御できない。

この場合, ログイン認証システムのログを常に監視し, 規定回数以上連続して同一のパスワードが使われた場合, 該当のパスワードを使用できないようにするなどの対策が必要。

この際, 該当のパスワードを使用しているユーザーがいれば事情を説明してパスワードを変更してもらうなどの対応がある。大規模システムにおいては, 同一時間内に多数のログインが想定されるため, 連続ではなくても短い一定時間内に同一のパスワードが規定回数以上使われた場合は同様の対策を取るなどの配慮も考えられる。

●演習 4

学校や教育センターなどで管理しているパソコン等の場合, `netstat` コマンドが使えない場合がある。その場合は, 後述する方法で対応。`netstat` コマンドの使い方についてはインターネット等を検索。

`netstat` コマンドをオプションなしで実行すると, 大量のポートが表示される。この中には外部からのアクセスが不要なものも含まれる。そのようなポートは外部からアクセスできないようにしておくことで安全性が高まる。

どれが外部からのアクセスが不要か, ポートを閉じる時はどのような手順を用いるかについては, インターネット等を検索して調べることができる。

よくあるサイバー攻撃等で使用される既知のポートについては、株式会社ラックが無料の自己診断サービスを提供している。これは、Windows のコマンドではないので、どのパソコンでも実施可能である。

<https://jisin.lac.co.jp/>

第4章【学習25】演習解答

●演習 1

1. 登録実行前にデータを確認する
2. 図書データを登録する
3. 登録実行後のデータを取得する
4. データが追加されていることを確認する

●演習 2

以下の URL を開き, 学習25の2を実行するとよい。

<https://colab.research.google.com/drive/1r26wcPi3jcVfe0BhzbyTj0-oGhi8AgBC#scrollTo=jk2BmGwA53N7>

※上記が表示されない, あるいは実行できない場合は, それができる所で内容を取り込む, 校内の共有フォルダにおいて生徒に実行させるなどの対応が考えられる。

●演習 3

基本的には, 作成した関数の最初に

```
import unittest
```

をつけ, テキストにあるように関数をテストするコードを書く。

●演習 4

「関数名の変更」の例

検索語入力関数の名称は妥当なのでそのまま

`title_input()` → `inputBookTitle()`

図書検索関数の名称も妥当なのでそのまま

`title_search()` → `searchBookInfo()`

貸出中判定関数の名称は中身が分からないので変更

`disp()` → `checkLending()`

「変数のインライン化」の例

(解答例 1)

```
title = input()
```

```
return title
```

↓

```
return input()
```

(解答例 2)

```
data = get(url,params)
```

```
return data
```

↓

```
return get(url,params)
```

(解答例 3)

```
result = title_search(title)
```

```
disp(result)
```

↓

```
disp(title_search(title))
```

(解答例 4)

```
title=title_input()
```

```
result=title_search(title)
```

```
disp(result)
```

↓

```
disp(title_search(title_input()))
```

(解答例 5)

```
p=urllib.parse.urlencode(params)
```

```
url=url+"?" +p
```

↓

```
url = url+"?" +urllib.parse.urlencode(params)
```

● 演習 5

テキストにある参考 URL などで確認する。