

<座席配置システム>

2011SE049 久松滉平 2011SE064 市川裕也 2011SE074 今津翔
2011SE261 棚瀬泰宏 2011SE308 吉田将成

1.はじめに

南山大学では指定席クラスや定期試験時に掲示・使用されている座席表は元のデータはあるものの、Excelなどの表ソフトを用いて各学生の番号を手入力で作成されているのが現状である。座席表作成は事務的な仕事であるが事務任せで作成に時間がかかったり、担当講義の教授が作成するということにもなっている。

このように同じような行動を繰り返すことが多い作業ならばコンピュータに配列作業を行わせることができるのではないかと考えた。今回の研究では、一部の教室を例にあげ実際に自動的に座席配置を行うシステムを作る。

2.現状と解決策

現状：

現在存在する「講義登録者リスト」と「座席表」の元データを用い登録者の座席配置を人の目でリストを確認し、人の手で座席表に学生番号を入力するという作業で掲示されている座席表は作られている。人数のみで教室を選択しているためきれいに配置できない事やC103のような大教室の配置を行う際には300人以上という可能性があり人為的ミスにより本来登録しているはずの学生の席がないということにもなりかねない。

解決案：

- ・「座席配置君」のようなGUIを作り、その内部のみで実行可能にする。
- ・システムにより自動で学生の配置を行う。
- ・空席を任意に選べるようにする。
- ・配置結果をPDF化し、保存する。
- ・ネットワーク上にも保存し、教員・事務員・学生が自由に見られるようにする。
- ・各部屋の席自体のファイルを作り、その変更のみで座席配置を可能にする。
- ・自分がどの部屋でどの席であるか検索出来るようにする。

以上の解決案から次の項目を最低限達成することとした。

- ・GUIから利用者の欲しいリストを作る。
- ・システムにより授業の登録リストから自動的に要求されたリストを作る。
- ・空席を任意に選択する。（ただし、要求定義された条件付き）
- ・配置結果をexe化し、保存する。
- ・これらは主に南山大学事務員向けに作られたソフトウェアである。

3.設計

3.1 要求仕様書

- ・並べ方の条件として空席の有無，縦，横，ジグザグを選択できる.
- ・配置は学籍番号の昇順に行う.
- ・教室配置を行う人が登録リストを持っている必要がある.
- ・登録リストはtext形式で管理する.
- ・登録リスト，教室リクエスト，並べ方の条件すべてを選択しなければ配置は実行できない.
- ・教室リクエストはCSVで作成しておく.
- ・実行した結果もCSVで出力する.

3.2 コンテキスト図

座席配置システムについて，外部実体を学生事務とする．学生事務により入力された時，座席配置システムにより座席表を出力するコンテキスト図を作成する．

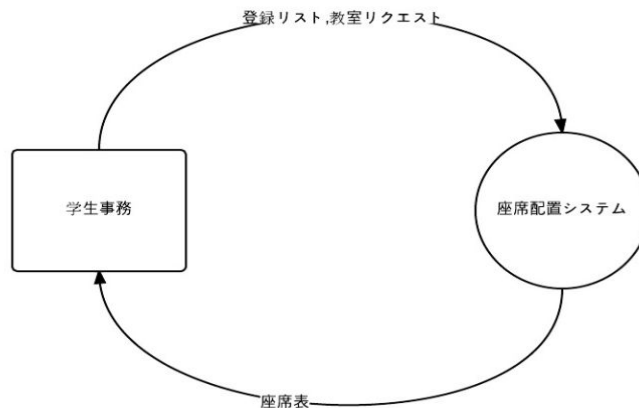


図3.2 コンテキスト図

3.3 DFD

DFD作成にあたり，座席配置システムを[入力処理]，[座席配置]，[出力処理]というシステムに分けた．これらのシステムを構成する際に4章で述べるが，言語はRubyが最適だと判断した．またRubyはCSVが最適と判断したためレイアウト（出力）はCSVで出力することにした．これらのことを考慮してDFDを作成させる．

以下の手順でシステム実行させたい

1. 学生事務が学生の登録リストと教室リクエストを入力する.
2. 入力処理が学生の登録リストと教室リクエストを受け取り，学生ストアに保存する．その後教室の条件を指定し，格納処理にそれを送る.

3. 格納処理は、学生ストアから登録リストと教室リクエストと教室条件を受け取り、学籍番号を配列に格納する。
 4. あらかじめ作っておいた教室のレイアウトを示したCSVを読み込み、3の教室条件と格納された学籍番号を用いて座席配置を行い、配列に格納する。それを用いてCSVで出力する。
 5. 学生事務に結果を返す。
- 以上よりDFDを作成する。

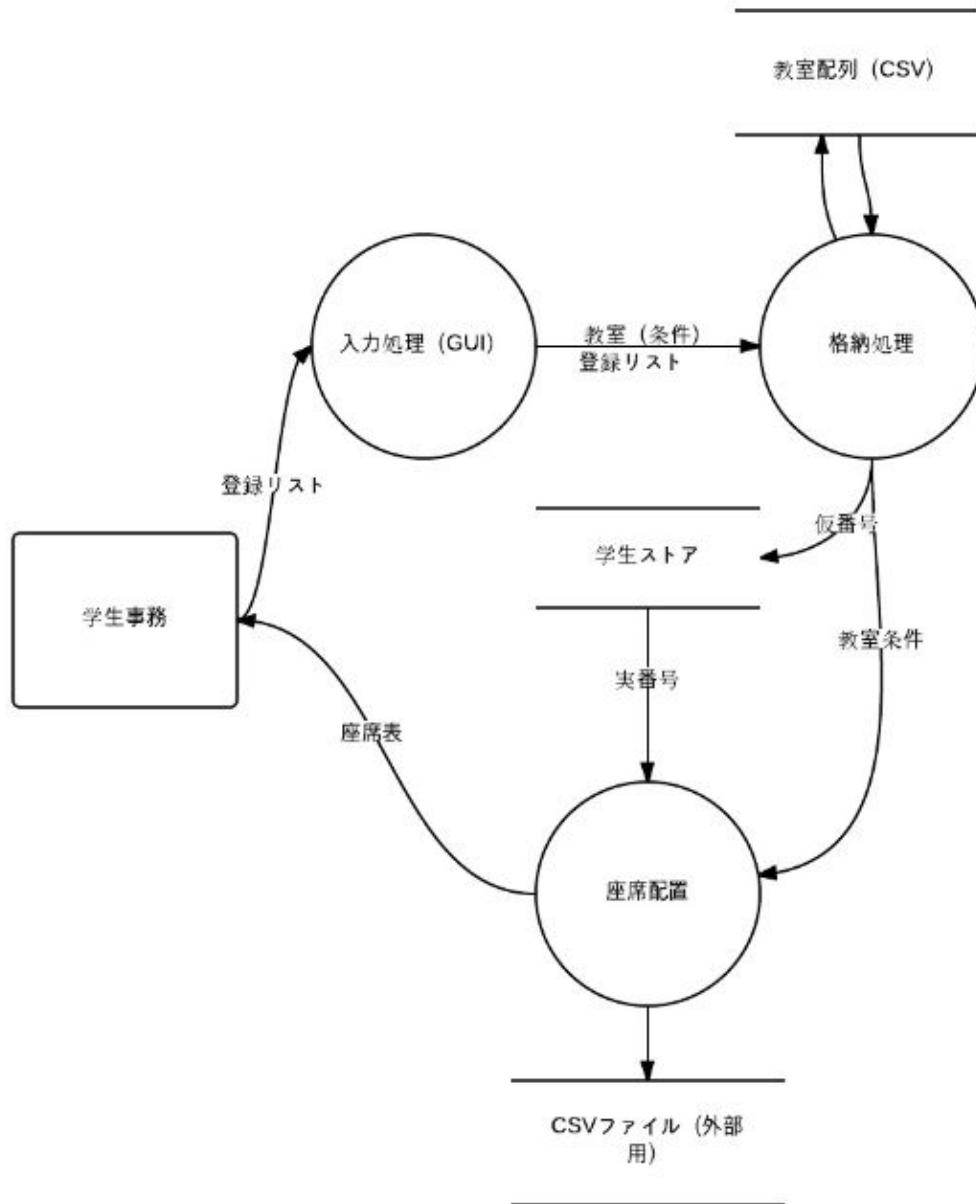


図3.3 DFD

3.4 クラス図

前節3.1~3.2を基にメインメソッドからの条件やリストを受け取り格納する、格納処理クラス(Fileクラス)と格納した配列などをもとに実際に座席配置を行う座席配置クラス(Setupクラス)の2つが必要だと考え、図3.4のクラス図を作成した。



図3.4

4. 言語

言語を決めるにあたって、上記に書かれたことを適切に実行するためオブジェクト指向プログラミングを調べた。そこで最初に浮上したのが様々なAPIをそろえているJavaである。しかし大学の講義で習得した中にAPIを使った演習は一度もなく、一から覚えるのが困難だと判断したためJavaは諦めた。

次に「テキスト処理」に着目したときに浮上したのがRubyである。Rubyの強みは5章で述べるがGUIを作成することができ、さらにテキスト処理では特定のファイルの読み書きが行えることである。またスクリプト言語なのでJavaなどとは違い簡単に記述できるので言語の仕様に割く時間が大幅に削減することができた。

後にexe化をする時にうまく行かなかった時期があったため、JavaとRubyを兼ね備えたJRubyを取り入れようとも考えたが6. 3章で述べるがRubyのexe化に成功したためこのままとなった。

5. 内部設計

5.1 Tk(GUI)

Tkを記述するにあたり参考書である[1]にTkのことが載っていなかったためインターネットで検索をした。検索したら様々なページで逆引きが載っていたため参考にし([2], [3])、上記で述べた工程を再現した。

ソースコードを記述する以前にユーザが直接触れる場所がGUIなので図5. 1のようにモデルを設計し図5. 1のように実装を開始した。しかし、実装していくにつれてRuby Tk/tclの資料([2], [3])が少なかったため図5. 2のようになってしまったがメリットも存在するため図5. 3にメリット・デメリットをまとめる。

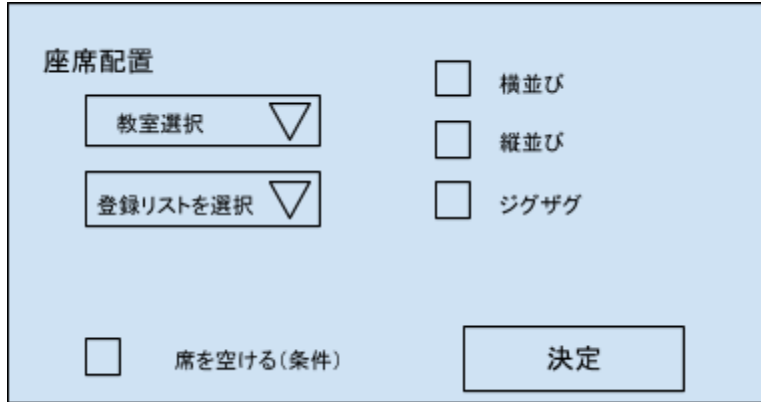


図5.1 GUIのモデル



図5.2 GUI完成系

メリット	デメリット
<p>・教室選択では選択が多いためチェックボックスにすることによりスペースをとらなくてすんだ</p>	<p>・パターンを選択のときに文字が長くなってしまって拡大・縮小してしまう</p>

図5.3 GUIの表

5.2 学生番号の格納処理

出力結果は学籍番号を並び変えたものを表示するため、学籍番号を名簿もしくはリストから受け取らなければならない（一時的に保存する）。今回はリスト(text形式)から受講者の学籍番号を受け取ると仮定し、受け取った学籍番号を配列に格納する事にした。ここでリストをtextファイル形式にした理由として、他の拡張子のファイルより汎用性があり、一般的なPCで容易に扱う事が出来るからである。

textファイルの制約は、

- ①学籍番号を先頭に書かなければならない
- ②一人ごとに学生情報(学籍番号, 名前)を改行しなければならぬ
- ③学籍番号は9文字以内でなければならない

以上の3つである。

リストから学籍番号を読み込む大まかな手順は

- ①格納するための配列を宣言 例gakuseki[]
- ②リストから読み込んだ学籍番号と名前を分けてにgakuseki保存する
- ③gakusekiを学籍番号で昇順に並び変える（その際にgakusekiとnameは対応付けしてある）

5.3 テキスト処理の提案と解決

テキスト処理をするにどのように格教室のレイアウトをデータ化するかを提案した。その提案と解決案を記す。また、提案と解決案の番号は対になっている。

提案

(1)二重配列にして座れる箇所に"1", 座れない箇所を"nil"をあらかじめ代入し、行と列にして格納する

(2)RubyではCSVの読み書きができるため、CSVのシートで座れる箇所に"1", 座れない箇所に" "を入れておきシステムで配列に"1"または" "格納させる

解決案

(1)Rubyでは多次元配列の使用が複雑

(2)Rubyで読み込ませて配列に格納するとき、二重配列になる

上記より(2)を採用しCSVを読み込ませた。その過程を下記で述べる。

5.3.1 CSV

CSVで教室の座席を作成させる。座席がある時は1。座席が無い、または通路を空白で表した。

これらを元に作られたCSVの座席表を、Rubyコードで読み込ませて配列に格納した。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	1	1	1			1	1	1	1	1			1	1	1	
2	1	1	1			1	1	1	1	1			1	1	1	
3	1	1	1			1	1	1	1	1			1	1	1	
4	1	1	1			1	1	1	1	1			1	1	1	
5	1	1	1			1	1	1	1	1			1	1	1	
6	1	1	1			1	1	1	1	1			1	1	1	
7	1	1	1			1	1	1	1	1			1	1	1	
8	1	1	1			1	1	1	1	1			1	1	1	
9	1	1	1			1	1	1	1	1			1	1	1	
10	1	1	1			1	1	1	1	1			1	1	1	
11	1	1	1			1	1	1	1	1			1	1	1	
12																
13																
14																

図5.3.1 CSV

5.3.2 教室配置

上記の5.3.1のCSVと学生番号を格納させた配列を用いて、各教室ごとの条件に従いソートを行う。その結果をCSVで出力させる。

その際、ソートを行った学生番号を格納した配列と指定された教室の座席を格納させた配列を用いて、教室条件に合わせてソートする。その結果をCSVで出力させた。この結果を以下に示す。

B203.csv - Microsoft Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	1	1	1				1	1	1	1				1	1	1
2	1	1	1				1	1	1	1				1	1	1
3	1	1	1				1	1	1	1				1	1	1
4	1	1	1				1	1	1	1				1	1	1
5	1	1	1				1	1	1	1				1	1	1
6	1	1	1				1	1	1	1				1	1	1
7	1	1	1				1	1	1	1				1	1	1
8	1	1	1				1	1	1	1				1	1	1
9	1	1	1				1	1	1	1				1	1	1
10	1	1	1				1	1	1	1				1	1	1
11	1	1	1				1	1	1	1				1	1	1
12																
13																
14																

<入力>



B203(1).csv - Microsoft Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1									教卓									
2																		
3	1	10101		10111			10121		10132		10143			10209		10220		
4	2	10102		10112			10122		10133		10144			10210		10221		
5	3	10103		10112			10123		10134		10145			10211		10222		
6	4	10104		10113			10124		10135		10201			10212		10223		
7	5	10105		10114			10125		10136		10202			10213		10224		
8	6	10106		10115			10126		10137		10203			10214		10225		
9	7	10107		10116			10127		10138		10204			10215		10226		
10	8	10108		10117			10128		10139		10205			10216		10227		
11	9	10108		10118			10129		10140		10206			10217		10228		
12	10	10110		10119			10130		10141		10207			10218		10229		
13	11	10111		10120			10131		10142		10208			10219		10230		
14	12																	
15	13																	
16	14																	

<出力>

図5.3.2 座席配置

6. テスト

6.1 単体テスト

6.1.1 Tk

5.1で述べたTkの実装で作成されたGUIを基にテストを行い、誤作動や機能の追加項目などを以下に記す.

①テスト項目

- | | |
|-----------------|-----------------------|
| (1)何も指定しなくて実行 | → 選択し直すテキストボックスが出力される |
| (2)登録リストを選択して実行 | → 選択し直すテキストボックスが出力される |
| (3)教室を指定して実行 | → 完了メッセージが出力される |
| (4)終了ボタンを押す | → 正常に終了する |

②誤作動項目の修正と出力

(1)①の(2)の完了メッセージが出力されたので条件に項目を足して同じ項目を実行したら選択し直す出力が表示された

(2)何が指定されたのか分からないため、端末上の出力に各々の変数の値を出力させたら選択したのと等しい値が返ってきた

6.1.2 CSV

空席の有無，縦，横，ジグザグの条件通りに正しく並び替えがされているか。

空席無しの縦並び

	A	B	C	D	E	F	G	H	I	J	K	L	M
1											教卓		
2													
3	1	09se001	09se016	09se031	09se046	10se011			10se026	10se041	11se006	11se021	11se036
4	2	09se002	09se017	09se032	09se047	10se012			10se027	10se042	11se007	11se022	11se037
5	3	09se003	09se018	09se033	09se048	10se013			10se028	10se043	11se008	11se023	11se038
6	4	09se004	09se019	09se034	09se049	10se014			10se029	10se044	11se009	11se024	11se039
7	5	09se005	09se020	09se035	09se050	10se015			10se030	10se045	11se010	11se025	11se040
8	6	09se006	09se021	09se036	10se001	10se016			10se031	10se046	11se011	11se026	11se041
9	7	09se007	09se022	09se037	10se002	10se017			10se032	10se047	11se012	11se027	11se042
10	8	09se008	09se023	09se038	10se003	10se018			10se033	10se048	11se013	11se028	11se043
11	9	09se009	09se024	09se039	10se004	10se019			10se034	10se049	11se014	11se029	11se044
12	10	09se010	09se025	09se040	10se005	10se020			10se035	10se050	11se015	11se030	11se045
13													
14													
15	11	09se011	09se026	09se041	10se006	10se021			10se036	11se001	11se016	11se031	11se046
16	12	09se012	09se027	09se042	10se007	10se022			10se037	11se002	11se017	11se032	11se047
17	13	09se013	09se028	09se043	10se008	10se023			10se038	11se003	11se018	11se033	11se048
18	14	09se014	09se028	09se044	10se009	10se024			10se039	11se004	11se019	11se034	11se049
19	15	09se015	09se030	09se045	10se010	10se025			10se040	11se005	11se020	11se035	11se050
20													



空席有りのジグザグ並び

	A	B	C	D	E	F	G	H	I	J	K	L	M
1											教卓		
2													
3	1	09se001		09se002		09se003			09se046		09se047		09se048
4	2	09se004		09se005		09se006			09se049		09se050		10se001
5	3	09se007		09se008		09se009			10se002		10se003		10se004
6	4	09se010		09se011		09se012			10se005		10se006		10se007
7	5	09se013		09se014		09se015			10se008		10se009		10se010
8	6	09se016		09se017		09se018			10se011		10se012		10se013
9	7	09se019		09se020		09se021			10se014		10se015		10se016
10	8	09se022		09se023		09se024			10se017		10se018		10se019
11	9	09se025		09se026		09se027			10se020		10se021		10se022
12	10	09se028		09se028		09se030			10se023		10se024		10se025
13													
14													
15	11	09se031		09se032		09se033			10se026		10se027		10se028
16	12	09se034		09se035		09se036			10se029		10se030		10se031
17	13	09se037		09se038		09se039			10se032		10se033		10se034
18	14	09se040		09se041		09se042			10se035		10se036		10se037
19	15	09se043		09se044		09se045			10se038		10se039		10se040
20													

上図より各処理は正しく処理され，正しくCSVに反映されている。

6.2 結合

6.1で述べた単体テストを基にクラス図の属性と照らし合わせて記述した。その際に単体で記述したソースコードにはコンストラクタがなかったため、属性やコンストラクタに注意して記述した。

はじめにGUIであるTkを記述し、学生番号と氏名を分ける格納処理クラスとのメッセージパッシングをできるようにした。その際、数値化された情報を送るため意図した処理がされているか分からないため、端末上に出力されるようにしてテストを行った。

次に座席配置クラスとmainメソッドをメッセージパッシングをできるようにした。まず教室名で各教室のメソッドに送らないといけなかったため判定メソッドでCSV処理ができるようにした。次に判定メソッドから各教室の処理（座席などの位置が違うため）のメソッドに移行した。その際も格プロセスが行われたのかを確認するために端末上に出力されるようにした。

最後にシステム全体のテストを行い正常に処理できていたので、端末上の出力部分はコメントとして記述されている。

6.3 exe化

6.3.1 exe化を行う理由

Rubyで実装したのでプログラムを実行するには、Rubyのライブラリや処理系などをインストールし実行環境を整える必要があるため、これではユーザビリティが欠けていると判断したためである。

exe化する事のメリットとデメリットは下記に記す。

メリット

- ・ Rubyの処理系をインストールしなくてよい
- ・ exeから簡単に起動する事が出来る

デメリット

- ・ 起動に時間がかかる
- ・ プログラム全体の容量が大きくなった

6.3.2. 解決案

exe化を行う手段としてexerb, ocra, RBatの3つのツールがあり、そのいずれかを使用することでexe化出来る事が分かった。

exerbとRBatはRubyのversionのアップデートにより対応していないので使用できない事もあり、対応しているocraを使用することにした。

ocraを使いexe化する過程で、教室配置を行うクラスと学籍番号格納クラスはメインクラスとソースファイルを分けて実装していたが、その事が起因となってexe化時にエラーが生じたためソースファイルを一つにまとめる形で統合する結果となった。それにより統合する作業が生じたがエラーは解消された。

7.考察

2章から最低限達成目標から考察する.

- ・システムにより授業の登録リストから自動的に要求されたリストを作る.

利用者は登録者名簿をtexで保存しなければならないが, その処理を除けば手動で作成していた手間が大幅に削減できたと考えられる.

- ・空席を任意に選択する. (ただし, 要求定義された条件付き)

教室のレイアウトCSVを手動で編集することで, 空席を任意に選択するという機能は実装する事が出来た. しかし, 本ソフトウェアのGUI上の機能としては, Ruby/tkの仕様とプログラム設計上, 空席を選択する機能を実装できなかったため, 設計をより厳密に考える必要があったと考える.

南山大学の事務員が使えるようなソフトウェアを作成できた.

このシステムはこの状態で完結ではなく, GUIに機能を追加する事が可能である.

また, 「ネットワーク上で見ることができる」という解決案もあげられていたが, 学内掲示板の載せ方などの配布方法の案が浮かんでこなかったため, この機能を追加することができなかった.

内部設計の面からの反省点で, 座席配置クラスを教室ごとに分けて作成したため, 座席配置のメソッドを教室の数の分だけ複数個記述する形になってしまった. exe化する際のエラーも上記が要因で起こった. これを一つのメソッドで座席配置を実現出来れば, より効率的で汎用的なプログラムが作成できたのではないかと考える.

その他に, 今回のソフトウェアではGUI上で選択されたファイルを確認することができた. そのためユーザーインターフェースの観点から, 選択したファイルを確認できるインターフェースを作ることが必要だと考察した.

8.役割分担

役割	レポート	作業工程
要求仕様書	久松	全員
DFD	棚瀬	棚瀬
クラス図	吉田	全員
言語設定	市川	全員
Tk(GUI)	今津	今津
格納処理	吉田	市川・吉田
CSV	棚瀬,久松	棚瀬, 吉田, 久松, 市川

参照

[1]たのしいRuby; SoftbankCreative株式会社発行 高橋征義, 後藤裕蔵 著

[2]逆引きRuby/Tk;

<http://pub.cozmixng.org/~the-rwiki/?cmd=view;name=%B5%D5%B0%FA%A4%ADRuby%2FTk>

[3]Tk-逆引きRuby;

<https://sites.google.com/site/gyakuhikiruby/home/gui/tk#TOC--12>

逆引きRuby 「ファイルを読み込む」;

<http://www.namarai.com/rubytips/?%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB%E3%82%A2%E3%82%AF%E3%82%BB%E3%82%B9#8>