

研究ノート

卒業要件の確認スキームと Java アプリの構築

張 本 浩

東京国際大学論叢 人間科学・複合領域研究 第3号 抜刷
2018年（平成30年）3月20日

卒業要件の確認スキームと Java アプリの構築

張 本 浩

Construction of a Scheme & a Java Application for Confirmation of Graduate Requirements

HARIMOTO, Hiroshi

abstract

The aim of the paper is to construct a scheme and a Java application for the senior students of the department of commerce to confirm whether they could meet graduation requirements by checking acquired curriculum subjects in the check sheets. For convenience, the primary target of the paper is limited to the curriculum of the department of commerce. It is, however, possible for you to construct some applications for other departments easily just to renew the titles and the names of curriculum subjects in the check sheets.

key words: Java application, confirmation of graduate requirement, department of commerce, curriculum subject, check sheet.

目 次

1. はじめに
2. カリキュラム体系と卒業要件
3. 情報処理の概要
 - 3.1 情報処理スキーム (information processing scheme)
 - 3.2 フレーム間の情報の流れ
4. Java プログラムと実行結果
 - 4.1 Java プログラム
 - 4.2 実行結果
5. おわりに

1. はじめに

本稿は、本学商学部の卒業年次の学生たちがブラウザ上のチェックシート（check sheet）のなかの単位取得済み科目をチェックすることによって卒業要件を満たしているかどうかを自身で確かめられるようにするアプリを構築することを目的としている。便宜上、商学部のカリキュラムのみを対象とするが、入力画面表示用のタイトルや学科目を入れ替えるだけで他学部用のアプリも容易に構築することができる。

2. カリキュラム体系と卒業要件

現在（2017年9月）、商学部のカリキュラムには、授業科目名で数えると、211の科目（TIUコア科目4、教養コア科目29、言語スキル科目34、自由選択科目25、専門100番台科目7、専門200番台科目38、専門300番台科目73、専門400番台科目1）が置かれ、授業科目区分ごとに卒業に必要な単位数が決められている。商学部ガイドブック（p. 74）に掲載されている卒業要件は次のとおりである。

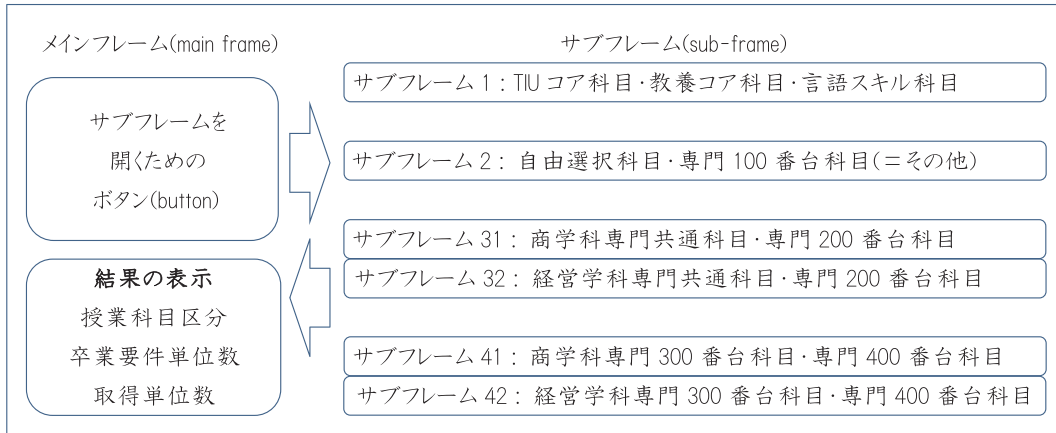
授業科目区分		卒業要件単位数	
必修	TIU コア科目	8	
選択	教養コア科目	20	
	言語スキル科目		
	学科内専門共通科目	4	12
	専門 200 番台科目	8	
	専門 300 番台科目および専門 400 番台科目	32	
その他	52		
合 計		124	

上記の卒業要件表で特記すべき事項は次の3点に絞られる。まず、「学科内専門共通科目」の中から4単位を超えて習得した単位は専門200番台科目の単位として扱われるとしている。次に、「その他」には「自由選択科目」、「専門100番台科目」、および「他学科・他学部履修科目」の中から習得した単位に、「教養コア科目、言語スキル科目、専門200番台科目、専門300番台科目および専門400番台科目」の中から規定以上に習得した単位を加えるとしている。最後に、他大学や他学部、および他学科で履修した授業科目の単位は60単位までを商学部の単位として認め「その他」の単位に算入されることになっている。

3. 情報処理の概要

3.1 情報処理スキーム (information processing scheme)

以上の卒業要件単位と特記事項を組み込み卒業判定を行うための基本フレームは下図のようにする。



- (1) メインフレームの該当ボタンをクリックするとサブフレームが開かれるようにし、各サブフレームに該当する履修学科目を配置する。
- (2) サブフレーム 1 には「TIU コア科目・教養コア科目・言語スキル科目」を配置し、サブフレーム 2 には「自由選択科目・専門 100 番台科目 (=その他)」を配置する。これらの科目は商学科と経営学科の共通科目であるので、それぞれ 1 つのフレームを設ける。
- (3) 商学部内の他学科履修科目の単位は「自由選択科目・専門 100 番台科目 (=その他)」の単位として加算しなければならないので、商学科と経営学科で設置科目の異なる「学科専門共通科目」、「専門 200 番台科目」、および「専門 300 番台科目および専門 400 番台科目」はそれぞれ 2 つのサブフレーム (商学科サブフレームと経営学科フレーム) を設ける必要がある。したがって、サブフレーム 3 とサブフレーム 4 にはそれぞれサブフレーム 31 (商学科)・サブフレーム 32 (経営学科) とサブフレーム 41 (商学科)・サブフレーム 42 (経営学科) を設ける。
- (4) 各サブフレームでそれぞれの分野の習得単位を求めメインフレームに引き渡した後、規定以上に習得した単位 (超過単位) を「自由選択科目・専門 100 番台科目 (=その他)」の単位に加算する。各学科における「その他」としての取得単位は以下の各項目で習得した単位の合計値となる。
 - ① 「自由選択科目・専門 100 番台科目 (=その他)」の中から習得した単位
 - ② 「教養コア科目・言語スキル科目」の中から規定以上に習得した単位
 - ③ 「学科内専門共通科目・専門 200 番台科目」の中から規定以上に習得した単位
 - ④ 「専門 300 番台科目および専門 400 番台科目」の中から規定以上に習得した単位
 - ⑤ 他大学・他学部・他学科で習得した単位

3.2 フレーム間の情報の流れ

メインフレーム (main frame) とサブフレーム (sub-frame) 間の情報の流れは以下のようにまとめられる。

まず、メインフレームからサブフレームを呼び出すと、該当する授業科目区分に属する授業科目がチェックボックス・リスト (checkbox list) の形で表示される。単位取得済みの科目にチェックを入れていくとその授業科目区分の総取得単位と、規定以上に取得した単位 (超過単位) がサブフレームの下の方に表示される。ここで、メインフレームの授業科目区分のボタンをクリックすると、先ほど開いていたサブフレームで求められた総取得単位と、超過単位がメインフレームに引き渡されメインフレームの下の方にそれらの単位が表示される。例えば、メインフレームで「《商学部共通》TIUコア科目・教養コア科目・言語スキル科目」のボタンをクリックすると、サブフレーム1が開かれ「TIUコア科目・教養コア科目・言語スキル科目」がすべて表示される。単位取得済みの科目にチェックを入れていくと、下の方に、①TIUコア科目の取得単位数、②教養コア科目+言語スキル科目の取得単位数、および③その他に回される単位数、が表示される。ここでメインフレームのいずれのボタンをクリックすると、先ほどのサブフレームで求められた単位数がメインフレームに引き渡されメインフレームの下の方の該当するテキストフィールドにそれらの単位数が表示される。

4. Javaプログラムと実行結果

4.1 Javaプログラム

```
//Graduate Requirements Confirmation Program
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

<applet code=grache2.class width=850 height=800></applet>

public class grache2 extends Applet implements ActionListener{
    subFrame1 sf1;
    subFrame2 sf2;
    subFrame31 sf31;
    subFrame32 sf32;
    subFrame41 sf41;
    subFrame42 sf42;
    Font font, afont;
    Label step1=new Label("【Step 1】所属学科を選択せよ(初期値=商学科)");
    Button com=new Button("商学科");
    Button man=new Button("経営学科");
    Label step2=new Label("【Step 2】以下のボタンをクリックして各ページを開き単位習得済みの科目をチェックせよ");
    Button bt1=new Button("《商学部共通》TIUコア科目・教養コア科目・言語スキル科目");
```

```

Button bt2=new Button("《商学部共通》自由選択科目・専門100番台科目(=その他)");
Button bt31=new Button("《商学科》専門共通科目・専門200番台科目");
Button bt32=new Button("《経営学科》専門共通科目・専門200番台科目");
Button bt41=new Button("《商学科》専門300番台科目・専門400番台科目");
Button bt42=new Button("《経営学科》専門300番台科目・専門400番台科目");
String[] str1={"【授業科目区分・卒業要件単位数・取得単位数】",
              "■TIUコア科目",
              "■教養コア科目・言語スキル科目",
              "■自由選択科目・専門100番台科目(=その他)",
              "■学科内専門共通科目・専門200番台科目",
              "■専門300番台科目・専門400番台科目",
              "■合計取得単位数"};
String[] str2={" ", "(8単位以上)", "(20単位以上)", "(52単位以上)",
              "(12単位以上)", "(32単位以上)", "(124単位以上)"};
Label[] subj=new Label[str1.length];
Label[] unit=new Label[str2.length];
TextField[] tf=new TextField[str2.length];
Label step3=new Label("【Step 3】他大学・他学部で習得した総単位数を入力しエンターキー
                      を押下せよ⇒");
TextField other=new TextField(50);
Button bt5=new Button("【授業科目区分・卒業要件単位数・取得単位数】");
int[] result=new int[str2.length];
int[] limit={0, 8, 20, 52, 12, 32, 124};
int i, j, k, xPos, yPos;
int tiuScore, genLanScore, sonota, otherUnit;
int f1Score, s21Score, f1s21, f2Score, s22Score, f2s22;
int s31Score, s31ScoreA, s32Score, s32ScoreA;
int over1, over21, over22, over31, over32;

public void init(){
    //サブフレーム・オブジェクトの生成
    sf1=new subFrame1("■TIUコア科目・教養コア科目・言語スキル科目");
    sf1.setSize(1800, 1000);
    sf1.setLocation(800, 20);
    sf2=new subFrame2("■自由選択科目・専門100番台科目");
    sf2.setSize(1800, 1000);
    sf2.setLocation(800, 20);
    sf31=new subFrame31("■商学科専門共通科目・専門200番台科目");
    sf31.setSize(1800, 1000);
    sf31.setLocation(800, 20);
    sf32=new subFrame32("■経営学科専門共通科目・専門200番台科目");

```

```
sf32.setSize(1800, 1000);
sf32.setLocation(800, 20);
sf41=new subFrame41("■商学科専門300番台科目・専門400番台科目");
sf41.setSize(1800, 1000);
sf41.setLocation(800, 20);
sf42=new subFrame42("■経営学科専門300番台科目・専門400番台科目");
sf42.setSize(1800, 1000);
sf42.setLocation(800, 20);
font=new Font("serif", Font.PLAIN, 18);
setFont(font);
setLayout(null);
step1.setBounds(20, 20, 400, 40);
step1.setBackground(Color.black);
step1.setForeground(Color.white);
com.setFont(afont);
com.setBounds(30, 70, 380, 60);
com.setBackground(Color.yellow);
com.setForeground(Color.black);
man.setFont(afont);
man.setBounds(420, 70, 390, 60);
man.setBackground(Color.blue);
man.setForeground(Color.white);
step2.setBounds(20, 150, 750, 40);
step2.setBackground(Color.black);
step2.setForeground(Color.white);
bt1.setBounds(20, 200, 800, 40);
bt1.setBackground(Color.orange);
bt1.setForeground(Color.black);
bt2.setBounds(20, 250, 800, 40);
bt2.setBackground(Color.orange);
bt2.setForeground(Color.black);
bt31.setBounds(20, 300, 395, 40);
bt31.setBackground(Color.yellow);
bt31.setForeground(Color.black);
bt32.setBounds(420, 300, 400, 40);
bt32.setBackground(Color.blue);
bt32.setForeground(Color.white);
bt41.setBounds(20, 350, 395, 40);
bt41.setBackground(Color.yellow);
bt41.setForeground(Color.black);
bt42.setBounds(420, 350, 400, 40);
```



```
bt42.setBackground(Color.blue);
bt42.setForeground(Color.white);
step3.setBounds(20, 410, 670, 40);
step3.setBackground(Color.black);
step3.setForeground(Color.white);
other.setBounds(700, 410, 100, 40);
add(step1);
add(com); add(man);
add(step2);
add(bt1);
add(bt2);
add(bt31); add(bt32);
add(bt41); add(bt42);
add(step3); add(other);
bt5.setBounds(100, 470, 590, 60);
bt5.setBackground(Color.green);
bt5.setForeground(Color.black);
add(bt5);
xPos=100;
yPos=540;
for(i=1; i<str1.length; i++){
    subj[i]=new Label(str1[i]);
    subj[i].setBounds(xPos, yPos, 370, 30);
    subj[i].setForeground(Color.black);
    subj[i].setBackground(Color.white);
    subj[i].setAlignment(Label.LEFT);
    add(subj[i]);
    unit[i]=new Label(str2[i]);
    if(i == 6) unit[i].setBounds(xPos+370, yPos, 120, 30);
    else unit[i].setBounds(xPos+380, yPos, 120, 30);
    unit[i].setForeground(Color.red);
    unit[i].setBackground(Color.white);
    unit[i].setAlignment(Label.LEFT);
    add(unit[i]);
    tf[i]=new TextField(100);
    tf[i].setBounds(xPos+510, yPos, 80, 30);
    tf[i].setForeground(Color.black);
    tf[i].setBackground(Color.white);
    add(tf[i]);
```

```
        yPos+=35;
    }
    com.addActionListener(this);
    man.addActionListener(this);
    bt1.addActionListener(this);
    bt2.addActionListener(this);
    bt31.addActionListener(this);
    bt32.addActionListener(this);
    bt41.addActionListener(this);
    bt42.addActionListener(this);
    other.addActionListener(this);
} //The end of init()

public void actionPerformed(ActionEvent e){
    if(e.getSource()==com){
        sf1.setVisible(false);
        sf2.setVisible(false);
        sf31.setVisible(false);
        sf32.setVisible(false);
        sf41.setVisible(false);
        sf42.setVisible(false);
        k=0;
    }
    else if(e.getSource()==man){
        sf1.setVisible(false);
        sf2.setVisible(false);
        sf31.setVisible(false);
        sf32.setVisible(false);
        sf41.setVisible(false);
        sf42.setVisible(false);
        k=1;
    }
    else if(e.getSource()==bt1){
        sf1.setVisible(true);
        sf2.setVisible(false);
        sf31.setVisible(false);
        sf32.setVisible(false);
        sf41.setVisible(false);
        sf42.setVisible(false);
    }
}
```

```
else if(e.getSource()==bt2){
    sf1.setVisible(false);
    sf2.setVisible(true);
    sf31.setVisible(false);
    sf32.setVisible(false);
    sf41.setVisible(false);
    sf42.setVisible(false);
}
else if(e.getSource()==bt31){
    sf1.setVisible(false);
    sf2.setVisible(false);
    sf31.setVisible(true);
    sf32.setVisible(false);
    sf41.setVisible(false);
    sf42.setVisible(false);
}
else if(e.getSource()==bt32){
    sf1.setVisible(false);
    sf2.setVisible(false);
    sf31.setVisible(false);
    sf32.setVisible(true);
    sf41.setVisible(false);
    sf42.setVisible(false);
}
else if(e.getSource()==bt41){
    sf1.setVisible(false);
    sf2.setVisible(false);
    sf31.setVisible(false);
    sf32.setVisible(false);
    sf41.setVisible(true);
    sf42.setVisible(false);
}
else if(e.getSource()==bt42){
    sf1.setVisible(false);
    sf2.setVisible(false);
    sf31.setVisible(false);
    sf32.setVisible(false);
    sf41.setVisible(false);
    sf42.setVisible(true);
}
else if(e.getSource()==other){
```

```
sf1.setVisible(false);
sf2.setVisible(false);
sf31.setVisible(false);
sf32.setVisible(false);
sf41.setVisible(false);
sf42.setVisible(false);
otherUnit=Integer.parseInt(other.getText());
}
else{
    //brake;
}
over1=over21=over22=over31=over32=0;
//TIU コア科目の習得単位
result[1]=sf1.getTiuScore();
genLanScore=sf1.getGenLanScore();
if(genLanScore > 20){
    over1=genLanScore-20;
    genLanScore=20;
}
else over1=0;
result[2]=genLanScore;
sonota=sf2.getSonotaScore()+otherUnit;
//商学科専門共通科目・専門200番台科目の習得単位
f1Score=sf31.getF1Score();
s21Score=sf31.getS21Score();
f1s21=f1Score+s21Score;
//経営学科専門共通科目・専門200番台科目の習得単位
f2Score=sf32.getF2Score();
s22Score=sf32.getS22Score();
f2s22=f2Score+s22Score;
//商学科専門300番台科目・専門400番台科目の習得単位
s31Score=sf41.getS31Score();
//経営学科専門300番台科目・専門400番台科目の習得単位
s32Score=sf42.getS32Score();
//規定以上の習得単位
if(k == 0){//商学科の場合
    if(f1s21 > 12){
        over21=f1s21-12;
        f1s21=12;
    }
    if(s31Score > 32){
```

```

        over31=s31Score-32;
        s31Score=32;
    }
    result[3]=sonota+over1+over21+over31+f2s22+s32Score;
    result[4]=f1s21;
    result[5]=s31Score;
    result[6]=result[1]+result[2]+result[3]+result[4]+result[5];
}
else{// 経営学科の場合
    if(f2s22 > 12){
        over22=f2s22-12;
        f2s22=12;
    }
    if(s32Score > 32){
        over32=s32Score-32;
        s32Score=32;
    }
    result[3]=sonota+over1+over22+over32+f1s21+s31Score;
    result[4]=f2s22;
    result[5]=s32Score;
    result[6]=result[1]+result[2]+result[3]+result[4]+result[5];
}
repaint();
} //The end of actionPerformed()

public void paint(Graphics g){
    xPos=20;
    yPos=595;
    for(i=1; i<str1.length; i++){
        if(i == 4){
            if((f1Score >= 4 || f2Score >= 4) && (result[i] >= limit[i]))
                tf[i].setBackground(Color.green);
            else tf[i].setBackground(Color.white);
        }
        else{
            if(result[i] >= limit[i]) tf[i].setBackground(Color.green);
            else tf[i].setBackground(Color.white);
        }
        if(result[i] >=10) tf[i].setText(" "+result[i]);
        else tf[i].setText(" "+result[i]);
    }
} //The end of for()

```



```
        tiuBox[i]=new Checkbox(tiuCore[i]);
        tiuBox[i].setBounds(xPos, yPos, boxW, boxH);
        tiuBox[i].setForeground(Color.black);
        tiuBox[i].setBackground(Color.yellow);
        tiuBox[i].addItemListener(this);
        add(tiuBox[i]);
        xPos+=(boxW+5);
    }
    xPos=50;
    yPos+=35;
    //教養コア科目の表示
    head1.setBounds(xPos, yPos, boxW, 30);
    head1.setForeground(Color.white);
    head1.setBackground(Color.black);
    head1.setAlignment(Label.LEFT);
    head1.setVisible(true);
    add(head1);
    for(i=0; i<genNum; i++){
        if(i%5 == 0){
            xPos=50;
            yPos+=35;
        }
        genBox[i]=new Checkbox(genCore[i]);
        genBox[i].setBounds(xPos, yPos, boxW, boxH);
        genBox[i].setForeground(Color.black);
        genBox[i].setBackground(Color.yellow);
        genBox[i].addItemListener(this);
        add(genBox[i]);
        xPos+=(boxW+5);
    }
    xPos=50;
    yPos+=35;
    //言語スキル科目の表示
    head2.setBounds(xPos, yPos, boxW, 30);
    head2.setForeground(Color.white);
    head2.setBackground(Color.black);
    head2.setAlignment(Label.LEFT);
    head2.setVisible(true);
    add(head2);
    for(i=0; i<lanNum; i++){
        if(i%5 == 0){
```



```

        xPos=50;
        yPos+=35;
    }
    lanBox[i]=new Checkbox(lanSkill[i]);
    lanBox[i].setBounds(xPos, yPos, boxW, boxH);
    lanBox[i].setForeground(Color.black);
    lanBox[i].setBackground(Color.yellow);
    lanBox[i].addItemListener(this);
    add(lanBox[i]);
    xPos+=(boxW+5);
}
xPos=50;
yPos+=50;
// 選択結果表示用ラベルとテキストフィールドの表示
for(i=0; i<str1.length; i++){
    show[i]=new Label(str1[i]);
    tf[i]=new TextField(50);
    show[i].setBounds(xPos, yPos, 350, 30);
    tf[i].setBounds(xPos+355, yPos, 100, 30);
    add(show[i]);
    if(i != 0) add(tf[i]);
    yPos+=40;
}
} //The end of showSub1()

public void itemStateChanged(ItemEvent ie){
    int xPos=50, yPos=200, hei=30;
    tiuScore=genScore=lanScore=over=0;
    //TIU コア科目の習得単位 (tiuScore)
    for(j=0; j<tiuNum; j++){
        if(tiuBox[j].getState()){
            tiuScore+=tiuUnit[j];
            tiuBox[j].setBackground(Color.green);
        }
        else tiuBox[j].setBackground(Color.yellow);
    }
    //教養コア科目の習得単位 (genScore)
    for(j=0; j<genNum; j++){
        if(genBox[j].getState()){
            genScore+=genUnit[j];
            genBox[j].setBackground(Color.green);
        }
    }
}

```

```
    }
    else genBox[j].setBackground(Color.yellow);
}
//言語スキル科目の習得単位 (lanScore)
for(j=0; j<lanNum; j++){
    if(lanBox[j].getState()){
        lanScore+=lanUnit[j];
        lanBox[j].setBackground(Color.green);
    }
    else lanBox[j].setBackground(Color.yellow);
}
//TIU コア科目の習得単位の表示
if(tiuScore >= 8) tf[1].setBackground(Color.green);
else tf[1].setBackground(Color.white);
tf[1].setText(" "+tiuScore);
//教養コア科目・言語スキル科目の習得単位 (genLanScore) と
//規定以上習得した単位 (over) の表示
genLanScore=genScore+lanScore;
if(genLanScore > 20) over=genLanScore-20;
else over=0;
if(genLanScore >= 20) tf[2].setBackground(Color.green);
else tf[2].setBackground(Color.white);
tf[2].setText(" "+genLanScore);
tf[3].setText(" "+over);
repaint();
} //The end of itemStateChanged()
public void paint(Graphics g){
    //g.drawString("subFrame1", 20, 750);
}
public int getTiuScore(){
    return tiuScore;
}
public int getGenLanScore(){
    return genLanScore;
}
} //The end of subFrame1

class subFrame2 extends Frame implements ItemListener{
    int xPos=50, yPos=50;
    int i, j, k, n;
    int freScore, slScore, sonota;
```

```

Font font;
//自由選択科目
String[] freSelect={"インターンシップ入門","インターンシップ(体験型1単位)","インターンシップ(体験型2単位)","インターンシップ(実践学習型3単位)","インターンシップ(実践学習型4単位)","ボランティア活動(1回目)","ボランティア活動(2回目)","アカデミック・ライティング","地域の安全と警察","観光まちおこしワークショップ入門","観光まちおこしワークショップ実践A","観光まちおこしワークショップ実践B","観光まちおこしワークショップ実践C","留学準備(ASP Prep)","海外ゼミナールA","海外ゼミナールB","特別授業A","特別授業B","特別実習 I A","特別実習 I B","特別実習 II A","特別実習 II B","特別実習 III A","特別実習 III B","特別実習 IV A","特別実習 IV B","基礎科目特殊講義A","基礎科目特殊講義B"};

int[] freUnit={2,1,2,3,4,1,1,2,2,2,2,2,1,4,2,2,4,2,2,2,2,2,2,2,2,4};
int freNum=freUnit.length;
Checkbox[] freBox=new Checkbox[freNum];
//専門100番台科目
String[] s1={"簿記","簿記演習","公務員対策講座(教養)A","公務員対策講座(教養)B","公務員対策講座(教養)C","公務員対策講座(教養)D","公務員対策講座(教養)E","公務員対策講座(教養)F","公務員対策講座(教養)G","公務員対策講座(教養)H","公務員対策講座(教養)I","公務員対策講座(教養)J","公務員対策講座(教養)K","公務員対策講座(教養)L","税理士対策講座A","税理士対策講座B","数学入門","社会史A","社会史B"};

int[] s1Unit={4,2,2,2,2,2,2,2,2,2,2,2,2,2,4,4,4};
int s1Num=s1Unit.length;
Checkbox[] s1Box=new Checkbox[s1Num];
//チェックボックスのタイトル表示用ラベルの生成
Label head0=new Label("自由選択科目");
Label head1=new Label("専門100番台科目");
//結果表示用ラベルとテキストフィールドの生成
String[] str1={"[取得単位数]","■自由選択科目・専門100番台科目(=その他)"};
Label[] show=new Label[str1.length];
TextField[] tf=new TextField[str1.length];
//subFrame2のコンストラクター
subFrame2(String title){
    super(title);
    font=new Font("TimeNewRoman", Font.PLAIN, 18);
    setFont(font);
    setLayout(null);
    showSub2();
    addWindowListener(new WindowAdapter()){

```

```
        public void windowClosing(WindowEvent e){
            setVisible(false);
        }
    });
} //The end of subFrame2( )
```

```
public void showSub2( ){
    int xPos=50, yPos=80, boxW=330, boxH=30;
    //自由選択科目の表示
    head0.setBounds(xPos, yPos, boxW, 30);
    head0.setForeground(Color.white);
    head0.setBackground(Color.black);
    head0.setAlignment(Label.LEFT);
    head0.setVisible(true);
    add(head0);
    for(i=0; i<freNum; i++){
        if(i%5 == 0){
            xPos=50;
            yPos+=35;
        }
        freBox[i]=new Checkbox(freSelect[i]);
        freBox[i].setBounds(xPos, yPos, boxW, boxH);
        freBox[i].setForeground(Color.black);
        freBox[i].setBackground(Color.yellow);
        freBox[i].addItemListener(this);
        add(freBox[i]);
        xPos+=(boxW+5);
    }
    xPos=50;
    yPos+=35;
    //専門100番台科目の表示
    head1.setBounds(xPos, yPos, boxW, 30);
    head1.setForeground(Color.white);
    head1.setBackground(Color.black);
    head1.setAlignment(Label.LEFT);
    head1.setVisible(true);
    add(head1);
    for(i=0; i<s1Num; i++){
        if(i%5 == 0){
            xPos=50;
            yPos+=35;
        }
    }
}
```

```

    }
    s1Box[i]=new Checkbox(s1[i]);
    s1Box[i].setBounds(xPos, yPos, boxW, boxH);
    s1Box[i].setForeground(Color.black);
    s1Box[i].setBackground(Color.yellow);
    s1Box[i].addItemListener(this);
    add(s1Box[i]);
    xPos+=(boxW+5);
}
xPos=50;
yPos+=50;
for(i=0; i<str1.length; i++){
    show[i]=new Label(str1[i]);
    tf[i]=new TextField(50);
    show[i].setBounds(xPos, yPos, 370, 30);
    tf[i].setBounds(xPos+375, yPos, 100, 30);
    add(show[i]);
    if(i != 0) add(tf[i]);
    yPos+=40;
}
} //The end of showSub2()

public void itemStateChanged(ItemEvent ie){
    int xPos=50, yPos=200, hei=30;
    freScore=s1Score=sonota=0;
    //自由選択科目の習得単位 (freScore)
    for(j=0; j<freNum; j++){
        if(freBox[j].getState()){
            freScore+=freUnit[j];
            freBox[j].setBackground(Color.green);
        }
        else freBox[j].setBackground(Color.yellow);
    }
    //専門100番台科目の習得単位 (s1Score)
    for(j=0; j<s1Num; j++){
        if(s1Box[j].getState()){
            s1Score+=s1Unit[j];
            s1Box[j].setBackground(Color.green);
        }
        else s1Box[j].setBackground(Color.yellow);
    }
}

```

```
sonota=freScore+s1Score;
if(sonota >= 52) tf[1].setBackground(Color.green);
else tf[1].setBackground(Color.white);
tf[1].setText(""+sonota);
} //The end of itemStateChanged()
public void paint(Graphics g){
    //g.drawString("subFrame2", 20, 750);
}
public int getSonotaScore(){
    return sonota;
}
} //The end of subFrame2

class subFrame31 extends Frame implements ItemListener{
    int xPos=50, yPos=50;
    int i, j, k, n;
    int f1Score, s21Score, over;
    Font font;
    //商学科専門共通科目
    String[] f1Comm={"商学概論", "ファイナンス概論"};
    int[] f1Unit={4,4};
    int f1Num=f1Unit.length;
    Checkbox[] f1Box=new Checkbox[f1Num];
    //商学科専門200番台科目
    String[] s2={"基礎演習(2年春)", "基礎演習(2年秋)", "商業史", "貿易論", "経営学概論", "金融
        論", "国際経済学", "財政学", "公共経済学", "民法", "労働法", "国際法", "スポーツ
        産業論基礎", "生涯スポーツ論", "スポーツ英語", "スポーツ心理学基礎", "トレー
        ニング論", "会計学概論", "情報社会と情報論理", "ウェブアプリ論", "ソーシャル
        メディア論", "データベース論", "公務員対策講座(専門) A", "公務員対策講座(専
        門) B", "公務員対策講座(専門) C", "公務員対策講座(専門) D", "税理士対策講座
        C", "税理士対策講座 D", "ビジネス英語", "ビジネス中国語", "ビジネススペイン語"};
    int[] s2Unit={2,2,4,4,4,4,4,4,4,4,4,2,2,2,2,4,4,4,4,2,2,2,2,2,2,4,4,4};
    int s2Num=s2Unit.length;
    Checkbox[] s2Box=new Checkbox[s2Num];
    //チェックボックスのタイトル表示用ラベルの生成
    Label head0=new Label("商学科専門共通科目");
    Label head1=new Label("商学科専門200番台科目");
    //結果表示用ラベルとテキストフィールドの生成
    String[] str1={"取得単位数",
        "■商学科専門共通科目・専門200番台科目",
        "■その他に回される単位数"};
```

```

Label[ ] show=new Label[str1.length];
TextField[ ] tf=new TextField[str1.length];
//subFrame31 のコンストラクター
subFrame31(String title){
    super(title);
    font=new Font("TimesNewRoman", Font.PLAIN, 18);
    setFont(font);
    setLayout(null);
    showSub31( );
    addWindowListener(new WindowAdapter(){
        public void windowClosing(WindowEvent e){
            setVisible(false);
        }
    });
}
} //The end of subFrame31( )

```

```

public void showSub31( ){
    int xPos=50, yPos=80, boxW=330, boxH=30;
    //商学科専門共通科目
    head0.setBounds(xPos, yPos, boxW, 30);
    head0.setForeground(Color.white);
    head0.setBackground(Color.black);
    head0.setAlignment(Label.LEFT);
    head0.setVisible(true);
    add(head0);
    for(i=0; i<flNum; i++){
        if(i%5 == 0){
            xPos=50;
            yPos+=35;
        }
        flBox[i]=new Checkbox(flComm[i]);
        flBox[i].setBounds(xPos, yPos, boxW, boxH);
        flBox[i].setForeground(Color.black);
        flBox[i].setBackground(Color.yellow);
        flBox[i].addItemListener(this);
        add(flBox[i]);
        xPos+=(boxW+5);
    }
    xPos=50;
    yPos+=35;
    //商学科専門200番台科目

```

```
head1.setBounds(xPos, yPos, boxW, 30);
head1.setForeground(Color.white);
head1.setBackground(Color.black);
head1.setAlignment(Label.LEFT);
head1.setVisible(true);
add(head1);
for(i=0; i<s2Num; i++){
    if(i%5 == 0){
        xPos=50;
        yPos+=35;
    }
    s2Box[i]=new Checkbox(s2[i]);
    s2Box[i].setBounds(xPos, yPos, boxW, boxH);
    s2Box[i].setForeground(Color.black);
    s2Box[i].setBackground(Color.yellow);
    s2Box[i].addItemListener(this);
    add(s2Box[i]);
    xPos+=(boxW+5);
}
xPos=50;
yPos+=50;
for(i=0; i<str1.length; i++){
    show[i]=new Label(str1[i]);
    tf[i]=new TextField(50);
    show[i].setBounds(xPos, yPos, 350, 30);
    tf[i].setBounds(xPos+355, yPos, 100, 30);
    add(show[i]);
    if(i != 0) add(tf[i]);
    yPos+=40;
}
} //The end of showSub31( )

public void itemStateChanged(ItemEvent ie){
    int xPos=50, yPos=200, hei=30, total;
    flScore=s21Score=over=0;
    //商学科専門共通科目習得単位 (flScore)
    for(j=0; j<fl1Num; j++){
        if(fl1Box[j].getState( )){
            flScore+=fl1Unit[j];
            fl1Box[j].setBackground(Color.green);
        }
    }
}
```



```

        else f1Box[j].setBackground(Color.yellow);
    }
    //商学科専門200番台科目習得単位 (s21Score)
    for(j=0; j<s2Num; j++){
        if(s2Box[j].getState()){
            s21Score+=s2Unit[j];
            s2Box[j].setBackground(Color.green);
        }
        else s2Box[j].setBackground(Color.yellow);
    }
    //商学科専門共通科目・専門200番台科目 (total)
    total=f1Score+s21Score;
    if((f1Score >= 4) & (total >= 12)) tf[1].setBackground(Color.green);
    else tf[1].setBackground(Color.white);
    if(total > 12) over=total-12;
    else over=0;
    tf[1].setText(""+total);
    tf[2].setText(""+over);
} //The end of itemStateChanged()
public void paint(Graphics g){
    //g.drawString("subFrame31", 20, 750);
}
public int getF1Score(){
    return f1Score;
}
public int getS21Score(){
    return s21Score;
}
} //The end of subFrame31

class subFrame32 extends Frame implements ItemListener{
    int xPos=50, yPos=50;
    int i, j, k, n;
    int f2Score, s22Score, over;
    Font font;
    //経営学科専門共通科目
    String[] f2Comm={"経営学概論", "会计学概論", "情報処理と情報倫理", "情報処理論"};
    int[] f2Unit={4,4,2,2};
    int f2Num=f2Unit.length;
    Checkbox[] f2Box=new Checkbox[f2Num];
    //経営学科専門200番台科目

```

```
String[] s2={"基礎演習(2年春)","基礎演習(2年秋)","商業概論","経営史","国際ビジネス論","ファイナンス概論","金融論","国際経済学","財政学","民法","スポーツ産業論基礎","中級簿記","工業簿記","中級簿記演習","工業簿記演習","ウェブアプリ論","ソーシャルメディア論","データベース論","経営科学","プログラミング基礎","公務員対策講座(専門)A","公務員対策講座(専門)B","公務員対策講座(専門)C","公務員対策講座(専門)D","税理士対策講座C","税理士対策講座D","ビジネス英語","ビジネス中国語","ビジネススペイン語"};

int[] s2Unit={2,2,4,4,4,4,4,4,4,4,2,4,4,4,4,2,2,4,4,4,4,2,2,2,2,2,4,4,4};
int s2Num=s2Unit.length;
Checkbox[] s2Box=new Checkbox[s2Num];
//チェックボックスのタイトル表示用ラベルの生成
Label head0=new Label("経営学科専門共通科目");
Label head1=new Label("経営学科専門200番台科目");
//結果表示用ラベルとテキストフィールドの生成
String[] str1={"取得単位数",
              "■経営学科専門共通科目・専門200番台科目",
              "■その他に回される単位数"};
Label[] show=new Label[str1.length];
TextField[] tf=new TextField[str1.length];
//subFrame31のコンストラクター
subFrame32(String title){
    super(title);
    font=new Font("TimesNewRoman", Font.PLAIN, 18);
    setFont(font);
    setLayout(null);
    showSub32();
    addWindowListener(new WindowAdapter(){
        public void windowClosing(WindowEvent e){
            setVisible(false);
        }
    });
}
//The end of subFrame32()
public void showSub32(){
    int xPos=50, yPos=80, boxW=330, boxH=30;
    //経営学科専門共通科目
    head0.setBounds(xPos, yPos, boxW, 30);
    head0.setForeground(Color.white);
    head0.setBackground(Color.black);
    head0.setAlignment(Label.LEFT);
    head0.setVisible(true);
    add(head0);
}
```

```
for(i=0; i<f2Num; i++){
    if(i%5 == 0){
        xPos=50;
        yPos+=35;
    }
    f2Box[i]=new Checkbox(f2Comm[i]);
    f2Box[i].setBounds(xPos, yPos, boxW, boxH);
    f2Box[i].setForeground(Color.black);
    f2Box[i].setBackground(Color.yellow);
    f2Box[i].addItemListener(this);
    add(f2Box[i]);
    xPos+=(boxW+5);
}
xPos=50;
yPos+=35;
//経営学科専門200番台科目
head1.setBounds(xPos, yPos, boxW, 30);
head1.setForeground(Color.white);
head1.setBackground(Color.black);
head1.setAlignment(Label.LEFT);
head1.setVisible(true);
add(head1);
for(i=0; i<s2Num; i++){
    if(i%5 == 0){
        xPos=50;
        yPos+=35;
    }
    s2Box[i]=new Checkbox(s2[i]);
    s2Box[i].setBounds(xPos, yPos, boxW, boxH);
    s2Box[i].setForeground(Color.black);
    s2Box[i].setBackground(Color.yellow);
    s2Box[i].addItemListener(this);
    add(s2Box[i]);
    xPos+=(boxW+5);
}
xPos=50;
yPos+=50;
for(i=0; i<str1.length; i++){
    show[i]=new Label(str1[i]);
    tf[i]=new TextField(50);
    show[i].setBounds(xPos, yPos, 350, 30);
```

```
tf[i].setBounds(xPos+355, yPos, 100, 30);
add(show[i]);
if(i != 0) add(tf[i]);
yPos+=40;
}
} //The end of showSub32( )

public void itemStateChanged(ItemEvent ie){
    int xPos=50, yPos=200, hei=30, total;
    f2Score=s22Score=over=0;
    //経営学科専門共通科目習得単位 (f2Score)
    for(j=0; j<f2Num; j++){
        if(f2Box[j].getState()){
            f2Score+=f2Unit[j];
            f2Box[j].setBackground(Color.green);
        }
        else f2Box[j].setBackground(Color.yellow);
    }
    //経営学科専門200番台科目習得単位 (s22Score)
    for(j=0; j<s2Num; j++){
        if(s2Box[j].getState()){
            s22Score+=s2Unit[j];
            s2Box[j].setBackground(Color.green);
        }
        else s2Box[j].setBackground(Color.yellow);
    }
    //経営学科専門共通科目・専門200番台科目習得単位 (total)
    total=f2Score+s22Score;
    if((f2Score >= 4) & (total >= 12)) tf[1].setBackground(Color.green);
    else tf[1].setBackground(Color.white);
    if(total > 12) over=total-12;
    else over=0;
    tf[1].setText(""+total);
    tf[2].setText(""+over);
} //The end of itemStateChanged( )

public void paint(Graphics g){
    //g.drawString("subFrame32", 20, 750);
}

public int getF2Score(){
    return f2Score;
}
}
```

```

public int getS2Score(){
    return s2Score;
}
}
//The end of subFrame32

class subFrame41 extends Frame implements ItemListener{
    int xPos=50, yPos=50;
    int i, j, k, n;
    int s31Score, over;
    Font font;
    //専門科目300番台
    String[] s3={"専門演習(3年春)","専門演習(3年秋)","専門演習(4年春)","専門演習(4年秋)",
        "卒業研究","マーケティング論","流通論","マーケティングコミュニケーション論",
        "マーケティングリサーチ","マーケティング戦略論","消費者行動論","商業・流通政策",
        "製品ブランド論","貿易実務","グローバルマーケティング論",
        "リレーションシップマーケティング","経営管理論","経営戦略論","ファイナンス論",
        "金融システム論","国際金融論","ベンチャーファイナンス論","ファイナンシャルプランニング論",
        "ファイナンシャルマーケット論","地域金融口座","日本経済論","産業経済論",
        "社会経済システム論","福祉の経済学","都市経済論","地域経済社会事情(中東)",
        "地域経済社会事情(東アジア)","地域経済社会事情(人の移動と地域)","会社法","行政法",
        "手形・小切手法","商取引法","スポーツ産業論","スポーツ管理論","スポーツ組織論",
        "財務会計論","財務分析論","管理会計論","企業情報システム論","FP3級資格講座",
        "FP2級資格講座","中小企業診断士資格講座","税理士対策講座E",
        "税理士対策講座F","税理士対策講座G","税理士対策講座H",
        "税理士対策講座I","税理士対策講座J","税理士対策講座K",
        "税理士対策講座L","税理士対策講座M","税理士対策講座N","事業継承",
        "職業指導","環境問題とビジネス","交通論","立地論","専門外国書研究(英語)",
        "商学部特殊講義A","商学部特殊講義B"};
    int[] s3Unit={2,2,2,2,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,2,4,4,4,4,4,4,4,4,2,4,4,4,2,2,2,4,4,4,2,2,4,2,2,2,2,2,2,2,2,2,2,2,2,4,4,2,4,4,2,2,4};
    int s3Num=s3Unit.length;
    Checkbox[] s3Box=new Checkbox[s3Num];
    //チェックボックス上のタイトル表示用ラベルの生成
    Label head0=new Label("専門300番台科目");
    String[] str1={"取得単位数",
        "■専門300番台科目",
        "■その他に回される単位数"};
    Label[] show=new Label[str1.length];
    TextField[] tf=new TextField[str1.length];
    //subFrame41のコンストラクター
    subFrame41(String title){

```

```
super(title);
font=new Font("TimeNewRoman", Font.PLAIN, 18);
setFont(font);
setLayout(null);
showSub41();
addWindowListener(new WindowAdapter(){
    public void windowClosing(WindowEvent e){
        setVisible(false);
    }
});
} //The end of subFrame41()

public void showSub41(){
    int xPos=50, yPos=80, boxW=330, boxH=30;
    //学科内専門共通科目
    head0.setBounds(xPos, yPos, boxW, 30);
    head0.setForeground(Color.white);
    head0.setBackground(Color.black);
    head0.setAlignment(Label.LEFT);
    head0.setVisible(true);
    add(head0);
    for(i=0; i<s3Num; i++){
        if(i%5 == 0){
            xPos=50;
            yPos+=35;
        }
        s3Box[i]=new Checkbox(s3[i]);
        s3Box[i].setBounds(xPos, yPos, boxW, boxH);
        s3Box[i].setForeground(Color.black);
        s3Box[i].setBackground(Color.yellow);
        s3Box[i].addItemListener(this);
        add(s3Box[i]);
        xPos+=(boxW+5);
    }
    xPos=50;
    yPos+=50;
    for(i=0; i<str1.length; i++){
        show[i]=new Label(str1[i]);
        tf[i]=new TextField(50);
        show[i].setBounds(xPos, yPos, 350, 30);
        tf[i].setBounds(xPos+355, yPos, 100, 30);
```

```

        add(show[i]);
        if(i != 0) add(tf[i]);
        yPos+=40;
    }
} //The end of dispSub41()

public void itemStateChanged(ItemEvent ie){
    int xPos=50, yPos=200, hei=30;
    s31Score=over=0;
    // 専門300番台科目 (s3Score)
    for(j=0; j<s3Num; j++){
        if(s3Box[j].getState()){
            s31Score+=s3Unit[j];
            s3Box[j].setBackground(Color.green);
        }
        else s3Box[j].setBackground(Color.yellow);
    }
    if(s31Score >= 32) tf[1].setBackground(Color.green);
    else tf[1].setBackground(Color.white);
    if(s31Score > 32) over=s31Score-32;
    else over=0;
    tf[1].setText(""+s31Score);
    tf[2].setText(""+over);
} //The end of itemStateChanged()
public void paint(Graphics g){
    //g.drawString("subFrame41", 20, 950);
}
public int getS31Score(){
    return s31Score;
}
} //The end of subFrame41

class subFrame42 extends Frame implements ItemListener{
    int xPos=50, yPos=50;
    int i, j, k, n;
    int s32Score, over;
    Font font;
    // 経営学科専門300番台科目
    String[] s3={" 専門演習 (3年春)", " 専門演習 (3年秋)", " 専門演習 (4年春)", " 専門演習 (4年秋)",
        " 卒業研究", " マーケティング論", " 流通論", " 経営管理論", " 経営戦略論", " 産業・
        組織心理学", " 制度と組織の経済学", " 中小企業論", " 経営組織論", " 生産管理論",

```

"異文化リーダーシップ論", "現代企業論", "人的資源管理論", "多国籍企業論", "起業論", "企業評価論", "企業研究", "ファイナンス論", "ベンチャーファイナンス論", "地域金融口座", "日本経済論", "社会経済システム論", "福祉の経済学", "都市経済論", "地域経済社会事情(中東)", "地域経済社会事情(東アジア)", "地域経済社会事情(人の移動と地域)", "会社法", "商取引法", "スポーツ産業論", "スポーツ管理論", "スポーツ組織論", "財務会計論", "財務分析論", "上級簿記", "国際会計論", "原価計算論", "会計監査論", "税務会計論", "管理会計論", "パソコン財務会計", "企業情報システム論", "ビジネスゲーム", "表計算プログラミング", "問題解決技法", "FP 3級資格講座", "FP2級資格講座", "中小企業診断士資格講座", "税理士対策講座 E", "税理士対策講座 F", "税理士対策講座 G", "税理士対策講座 H", "税理士対策講座 I", "税理士対策講座 J", "税理士対策講座 K", "税理士対策講座 L", "税理士対策講座 M", "税理士対策講座 N", "事業継承", "職業指導", "環境問題とビジネス", "交通論", "立地論", "専門外国書研究(英語)", "商学部特殊講義 A", "商学部特殊講義 B");

```
int[] s3Unit={2,2,2,2,4,4,4,4,4,4,4,4,4,4,4,4,4,4,2,4,4,2,4,4,4,4,4,4,4,2,2,2,2,2,2,4,4,4,4,4,4,2,2,2,4,4,2,4,2,2,2,2,2,2,2,2,2,2,2,4,4,2,4,2,2,4};
```

```
int s3Num=s3Unit.length;
```

```
Checkbox[] s3Box=new Checkbox[s3Num];
```

```
//チェックボックスのタイトル表示用ラベルの生成
```

```
Label head0=new Label("専門300番台科目");
```

```
String[] str1={"取得単位数",
```

```
    "■専門300番台科目",
```

```
    "■その他に回される単位数"};
```

```
Label[] show=new Label[str1.length];
```

```
TextField[] tf=new TextField[str1.length];
```

```
//subFrame41のコンストラクター
```

```
subFrame42(String title){
```

```
    super(title);
```

```
    font=new Font("TimeNewRoman", Font.PLAIN, 18);
```

```
    setFont(font);
```

```
    setLayout(null);
```

```
    showSub42();
```

```
    addWindowListener(new WindowAdapter(){
```

```
        public void windowClosing(WindowEvent e){
```

```
            setVisible(false);
```

```
        }
```

```
    });
```

```
}//The end of subFrame41()
```

```
public void showSub42(){
```

```
    int xPos=50, yPos=80, boxW=330, boxH=30;
```



```

//経営学科専門300番台科目の表示
head0.setBounds(xPos, yPos, boxW, 30);
head0.setForeground(Color.white);
head0.setBackground(Color.black);
head0.setAlignment(Label.LEFT);
head0.setVisible(true);
add(head0);
for(i=0; i<s3Num; i++){
    if(i%5 == 0){
        xPos=50;
        yPos+=35;
    }
    s3Box[i]=new Checkbox(s3[i]);
    s3Box[i].setBounds(xPos, yPos, boxW, boxH);
    s3Box[i].setForeground(Color.black);
    s3Box[i].setBackground(Color.yellow);
    s3Box[i].addItemListener(this);
    add(s3Box[i]);
    xPos+=(boxW+5);
}
xPos=50;
yPos+=50;
for(i=0; i<str1.length; i++){
    show[i]=new Label(str1[i]);
    tf[i]=new TextField(50);
    show[i].setBounds(xPos, yPos, 350, 30);
    tf[i].setBounds(xPos+355, yPos, 100, 30);
    add(show[i]);
    if(i != 0) add(tf[i]);
    yPos+=40;
}
} //The end of showSub42( )

public void itemStateChanged(ItemEvent ie){
    int xPos=50, yPos=200, hei=30;
    s32Score=over=0;
    //専門300番台科目 (s3Score)
    for(j=0; j<s3Num; j++){
        if(s3Box[j].getState()){
            s32Score+=s3Unit[j];
            s3Box[j].setBackground(Color.green);
        }
    }
}

```

```

    }
    else s3Box[j].setBackground(Color.yellow);
}
if(s32Score >= 32) tf[1].setBackground(Color.green);
else tf[1].setBackground(Color.white);
if(s32Score > 32) over=s32Score-32;
else over=0;
tf[1].setText(""+s32Score);
tf[2].setText(""+over);
} //The end of itemStateChanged()
public void paint(Graphics g){
    //g.drawString("subFrame42", 20, 750);
}
public int getS32Score(){
    return s32Score;
}
} //The end of subFrame42

```

4.2 実行結果

[メインフレーム]

【授業科目区分・卒業要件単位数・取得単位数】		
■TIUコア科目	(8単位以上)	0
■教養コア科目・言語スキル科目	(20単位以上)	0
■自由選択科目・専門100番台科目(=その他)	(52単位以上)	0
■学科内専門共通科目・専門200番台科目	(12単位以上)	0
■専門300番台科目・専門400番台科目	(32単位以上)	0
■合計取得単位数	(124単位以上)	0

このメインフレームで、(a) 商学科をクリックした後、(b) 「《商学部共通》TIUコア科目・教養コア科目・言語スキル科目」のボタンをクリックすると以下のサブフレームが開き、単位取得済み科目にチェックを入れるとフレームの下に該当科目の取得単位数が表示される。

[サブフレーム 1]

TIUコア科目・教養コア科目・言語スキル科目

TIUコア科目	初年次演習(春)	初年次演習(秋)	ICT基礎	大学生生活デザイン演習	キャリア・Reスタート
教養コア科目	哲学	倫理学	芸術論	文学	こは学入門
	アジア・アラブ史	アメリカ・ヨーロッパ史	日本史	地理学概論	人文地理学
	日本の文化	異文化概論	文化人類学	Introduction to American Society	Life & Tradition in America
	法学	憲法	政治学	経済学	社会学
	現代の社会	心理学概論	自然科学概論	環境と自然	科学思想史
	統計学入門	情報処理の基礎	健康スポーツ科学	健康スポーツ実技(1科目)	健康スポーツ実技(2科目)
	健康スポーツ実技(3科目)	健康スポーツ実技(4科目)	健康スポーツ実技(5科目)	健康スポーツ実技(6科目)	健康スポーツ実技(7科目)
	健康スポーツ実技(8科目)				
言語スキル科目	Oral Communication	Reading & Writing	English Workshop	Media English	TOEIC
	英語特論	Communication Basic I	Communication Basic II	English Comprehension I	English Comprehension II
	Basic Speaking	Basic Writing	Advanced Speaking & Listening A	Advanced Speaking & Listening B	Advanced Speaking & Listening C
	Advanced Reading & Writing A	Advanced Reading & Writing B	Advanced Reading & Writing C	ドイツ語入門	実践ドイツ語
	フランス語入門	実践フランス語	スペイン語入門	実践スペイン語	中国語入門
	実践中国語	ハンガール入門	実践ハンガール	ロシア語入門	アラビア語入門
	日本語文章理解・表現 I	日本語文章理解・表現 II	日本語文章理解・表現 III	聴解・口頭表現 I	聴解・口頭表現 II
	総合日本語 I	総合日本語 II	基礎日本語文法 A	基礎日本語文法 B	ビジネス日本語

取得単位数

- TIUコア科目: 6
- 教養コア科目+言語スキル科目: 16
- その他に回される単位数: 0

ここで、メインフレームの「《商学部》専門共通科目・専門200番台科目」のボタンをクリックすると、メインフレームの下の「授業科目区分・卒業要件単位数・取得単位数」のところに先ほどの「《商学部共通》TIUコア科目・教養コア科目・言語スキル科目」のサブフレームの下に表示された取得単位数がメインフレームの下の「授業科目区分・卒業要件単位数・取得単位数」のところに集計表示されると同時に、「《商学部》専門共通科目・専門200番台科目」のサブフレームが開示されるので、こでも単位取得済み科目にチェックを入れるとフレームの下に該当科目の取得単位数が表示される。

[メインフレーム]

アプレット・ビューア: grache2.class

アプレット

【Step 1】 所属学科を選択せよ(初期値=商学科)

商学科 | 経営学科

【Step 2】 以下のボタンをクリックして各ページを開き単位取得済みの科目をチェックせよ

《商学部共通》TIUコア科目・教養コア科目・言語スキル科目

《商学部共通》自由選択科目・専門100番台科目(=その他)

《商学科》専門共通科目・専門200番台科目 | 《経営学科》専門共通科目・専門200番台科目

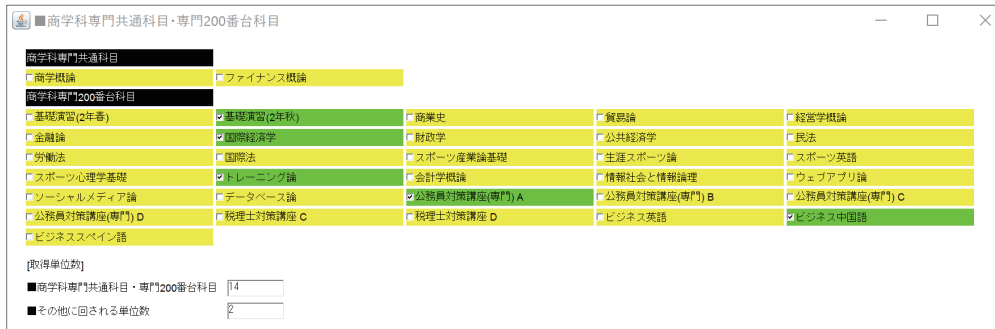
《商学科》専門300番台科目・専門400番台科目 | 《経営学科》専門300番台科目・専門400番台科目

【Step 3】 他大学・他学部で習得した総単位数を入力しエンターキーを押下せよ⇒

【授業科目区分・卒業要件単位数・取得単位数】

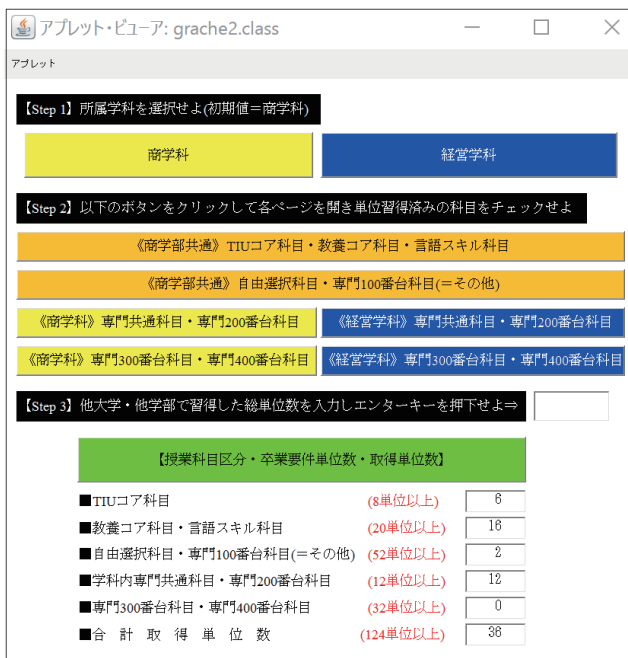
TIUコア科目	(8単位以上)	6
教養コア科目・言語スキル科目	(20単位以上)	16
自由選択科目・専門100番台科目(=その他)	(52単位以上)	0
学科内専門共通科目・専門200番台科目	(12単位以上)	0
専門300番台科目・専門400番台科目	(32単位以上)	0
合計取得単位数	(124単位以上)	22

[サブフレーム31]



ここで、メインフレームの「《商学部》専門300番台科目・専門400番台科目」のボタンをクリックすると、いままでチェックしたすべての単位取得済み科目の取得単位数がメインフレームの下の「授業科目区分・卒業要件単位数・取得単位数」のところに集計表示されると同時に、「《商学部》専門300番台科目・専門400番台科目」のフレームが開示される。

[メインフレーム]



[サブフレーム 41]

■商学科専門300番台科目・専門400番台科目				
■専門300番台科目				
■専門演習(3年春)	■専門演習(3年秋)	■専門演習(4年春)	■専門演習(4年秋)	■卒業研究
■マーケティング論	■流通論	■マーケティングコミュニケーション論	■マーケティングリサーチ	■マーケティング戦略論
■消費者行動論	■商業・流通政策	■製品ブランド論	■貿易実務	■グローバルマーケティング論
■リレーションシップマーケティング	■経営管理論	■経営戦略論	■ファイナンス論	■金融システム論
■国際金融論	■ベンチャーファイナンス論	■ファイナシャルプランニング論	■ファイナシャルマーケット論	■地域金融口座
■日本経済論	■産業経済論	■社会経済システム論	■福祉の経済学	■都市経済論
■地域経済社会事情(中東)	■地域経済社会事情(東アジア)	■地域経済社会事情(人の移動と地域)	■会社法	■行政法
■手形・小切手法	■商取引法	■スポーツ産業論	■スポーツ管理論	■スポーツ組織論
■財務会計論	■財務分析論	■管理会計論	■企業情報システム論	■FP3級資格講座
■FP2級資格講座	■中小企業診断士資格講座	■税理士対策講座E	■税理士対策講座F	■税理士対策講座G
■税理士対策講座H	■税理士対策講座I	■税理士対策講座J	■税理士対策講座K	■税理士対策講座L
■税理士対策講座M	■税理士対策講座N	■事業継承	■職業指導	■環境問題とビジネス
■交通論	■立地論	■専門外国語研究(英語)	■商学部特殊講義 A	■商学部特殊講義 B

取得単位数

■専門300番台科目

■その他に回される単位数

5. おわりに

前節のプログラムは以下の点で拡張可能である。

- プログラムの処理対象は「商学部2016年度以降入学の学生」を対象とするカリキュラムであり、実用のためには「2015年度以前入学の学生」を対象とするカリキュラムの内容を反映する必要がある。
- プログラムでは商学部の2学科のカリキュラムに含まれている諸科目が網羅されているので商学部の他学科の単位取得済み科目の単位計算には問題ないが、他大学および本学商学部以外の他学部の諸科目は含まれておらず、他大学と他学部で履修し単位取得した科目の単位は初期画面で直接数字入力するようになっている。したがって、実用上せめて他学部のカリキュラムに載っている諸科目まで取り扱えるようにすることが望まれる。

前節のプログラムは筆者のWebページ「<http://www2.tiu.ac.jp/~harimoto/>」からダウンロード可能であるので、応用的Javaプログラムの開発に関心のある人は筆者のWebページからプログラムをダウンロードし上記の拡張可能事項を組み込んだプログラムを開発されたい。

参考文献

- 1) 深井裕二『Java人工知能プログラミング』三恵社, 2016年.
- 2) 水野信也・中田 誠・上杉徳彦・永田正樹『Java GUI編』静岡学術出版, 2011年.
- 3) 森崎雅稔『最新Java コーディング作法』日経BP社, 2011年.
- 4) プライアン・ゲーツ, ダグ・リー『Java並行処理プログラミング』Soft Bank Creative, 2010年.
- 5) 伊藤俊秀・草薙信照『コンピュータ・シミュレーション』Ohmsha, 2006年.
- 6) 児玉公信『モデリングの本質』日経BP社, 2004年.
- 7) 峯村吉泰『コンピュータ・グラフィックス』森北出版, 2003年.
- 8) 速水治夫, 宮崎収兄, 山崎清明『データベース』Ohmsha, 2002年.
- 9) J. G. ヒューズ『オブジェクト指向データベース』サイエンス社, 1998年.
- 10) ティモシー A・パッド『オブジェクト指向プログラミング入門』アジソン・ウェスレイ, 1997年.

- 11) Daniel Shiffman, *Nature of Code*, Born Digital Inc. 2014.
- 12) Kirk Knoernschild, *Java Application Architecture*, Prentice Hall, 2012.
- 13) Robert Sedgewick, *Algorithms in Java*, Princeton University Press, 2010.
- 14) Cay S. Horstmann/Gary Cornell, *Core Java 2: Volume II-Advanced Features*, ASCII, 2006.
- 15) John. G. Hughes, *Object Oriented Database*, Science, 1998.