

プログラム設計 後期期末試験

提出する全ての asta ファイルの図内に、自分の学科の出席番号と氏名を「ノート」を使って書くこと

PBL 実験で、「授業で出されている課題を管理する」システムを開発することにした。この開発における設計と実装に関して、以下の **1** ～ **5** の間に答えなさい。

1 開発するシステムのシナリオを考えるために、システムの機能を以下のように整理した。

- 学生は、課題を閲覧することができる
- 学生は、期限の近い課題の通知を受けることができる
- 学生は、提出した課題を完了状態にすることができる
- 教員は、課題を閲覧することができる
- 教員は、課題を登録することができる
- 教員は、課題を取り消すことができる
- 教員は、課題を登録・取消をする際に、必ずパスワードを入力する必要がある

この時、以下の要素をアクタ、ユースケースとしたユースケース図を astah で作成しなさい。

アクタ 学生, 教員

ユースケース 課題を登録する, 課題を取り消す, 課題を閲覧する, 期限の近い課題の通知を受ける, 提出した課題を完了状態にする, パスワードを入力する

2 「照度センサと赤外線センサによって人の通過を検知した際に、課題の期限をディスプレイに出力し、メールで学生に通知する」部分の機能を以下のような処理でまとめた。

出力判断 照度センサと赤外線センサから値を受け取ると、出力指示を「期限確認」へ渡す。

期限確認 「出力判断」から出力指示を受け取り、課題リストから課題内容を読み込み、「期限出力」と「期限通知」に課題内容を渡す。

期限出力 「期限確認」から課題内容を受け取り、ディスプレイに更新を渡す。

期限通知 「期限確認」から課題内容を受け取り、学生に通知を渡す。

この時、以下のような要素を使ったデータフロー図を astah で作成しなさい。

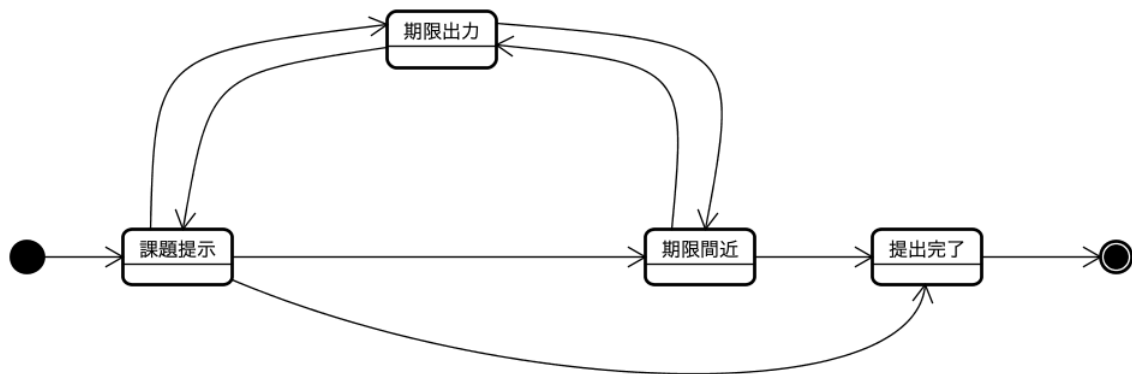
外部エンティティ 照度センサ, 赤外線センサ, ディスプレイ, 学生

プロセス 出力判断, 期限確認, 期限出力, 期限通知

データストア 課題リスト

データフロー シナリオ中の下線部

3 「課題状況の変化」を、ステートマシン図で設計することを考える。次のような図のところまで作成した。



この時、上記の図に以下の要素を追加したステートマシン図を astah で完成させなさい。

アクティビティ

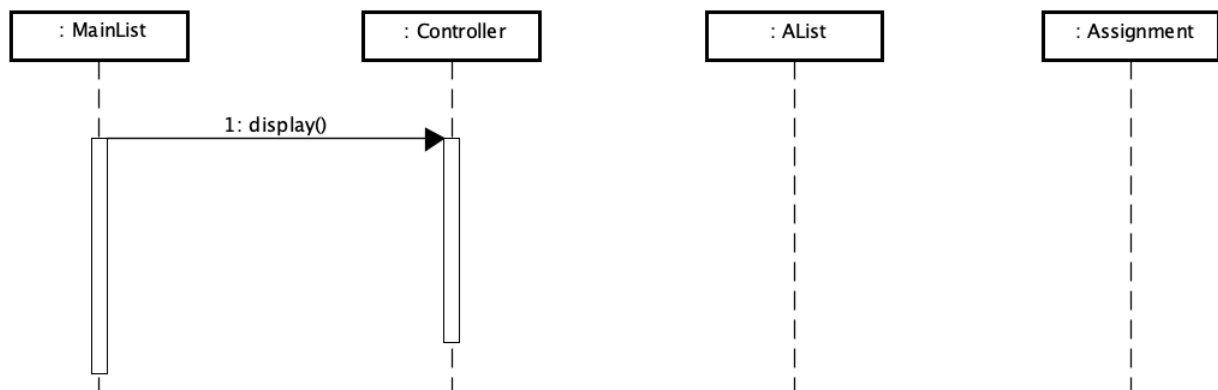
- 状態「課題提示」の entry アクティビティとして、「課題リストへ追加」を追加
- 状態「期限出力」の内部アクティビティとして、トリガーが「出力」、ガードが「期限間近」、アクションが「先頭行に出力する」を追加
- 状態「課題完了」の exit アクティビティとして、「課題リストから削除」を追加

遷移

- 状態「課題提示」から状態「期限出力」へは、トリガー「センサ反応あり」で遷移する
- 状態「期限出力」から状態「課題提示」へは、トリガー「センサ反応なし」で遷移する
- 状態「期限間近」から状態「期限出力」へは、トリガー「センサ反応あり」で遷移する
- 状態「期限出力」から状態「期限間近」へは、トリガー「センサ反応なし」で遷移する
- 状態「課題提示」から状態「期限間近」へは、トリガー「期限 3 日前になる」、ガード「未提出」で遷移する
- 状態「課題提示」から状態「提出完了」へは、トリガー「提出する」で遷移する
- 状態「期限間近」から状態「提出完了」へは、トリガー「提出する」で遷移する
- 上記以外の遷移は、トリガー、カードなしで遷移する

4 「期限間近な課題を先頭に出力し、課題期限を出力する」部分のみを、Java のクラス Assignment, AList, Controller, MainList で作成した。(プログラムは問題の後に掲載している。)

この時の振る舞いを表したシーケンス図の一部が以下のようになった。



astah を使って、上記のメソッド display に関するシーケンス図を完成させなさい。以下の点に注意して作成すること。

- ライフラインは、クラス Assignment, AList, Controller, MainList のみとする
- ライフラインのオブジェクト名は必要ない（クラス名のみ）
- メッセージの引数は描く必要はない
- ライフラインの消滅の「×」印は描く必要はない
- 複合フラグメントを使う際は名前は付けなくてもよい（その処理内容がわかる名前を付けてもよい）
- 複合フラグメントのガードは省略せずに描くこと
- 複合フラグメントに含まれるメッセージは、含まれることが分かるように枠内に配置すること

5 astah を使って、クラス AList のメソッド show の処理を表すアクティビティ図を作成しなさい。

問題はここまで

1～**5** 各 20 点

定期試験の実施について

試験中に使用できるもの

- 筆記用具（メモ用紙は必要な人に配布）
- 演習室のコンピューター一台（一つの机に一人の配置で、座る場所はどこでもよい）

試験中に参照できるもの

- 自分のホームディレクトリ（ホームフォルダ）以下に保存されているファイル
- * 上記以外の情報を参照することは不正行為とする
（例：USB で接続された機器に保存されているファイルの参照など）
- * 試験中は、演習室外へのネットワークアクセスは遮断される

答案の提出

- 提出する全ての asta ファイルの図内に、自分の学科の出席番号と氏名を「ノート」を使って書く
- 保存したファイルは次のように「report」コマンドで提出する
（ちゃんと提出できた場合は、「Succeed.」と画面に表示される）

```
$ ~kogai/report pd2term 「プログラムファイル」
```
- 各問で作成したファイルを以下のように、report コマンドを分けて提出する
例えば、test1.asta と test2.asta のファイルを提出したい場合は、次のように 2 回に分けて提出する

```
$ ~kogai/report pd2term test1.asta  
$ ~kogai/report pd2term test2.asta
```
- 同じ問題に対して、複数の提出ファイルが存在した場合は、更新日時が新しい方を提出ファイルとする

期限間近な課題を先頭に出だし、課題期限を出力するプログラム

```

class Assignment {
    private String title;
    private int deadline;
    private String status;
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getStatus() {
        return status;
    }
    public void setStatus(String status) {
        this.status = status;
    }
    public int getDeadline() {
        return deadline;
    }
    public void setDeadline(int deadline) {
        this.deadline = deadline;
    }
    public void show() {
        System.out.printf("title: %s, ",
                           title);
        System.out.printf("deadline: %d, ",
                           deadline);
        System.out.printf("status: %s\n",
                           status);
    }
}

```

```

class AList {
    private Assignment[] alist;
    private int count;
    public AList() {
        alist = new Assignment[7];
        count = 0;
    }
    public void add(Assignment a) {
        alist[count] = a;
        count++;
    }
    public void show() {
        if(count==0) {
            System.out.println("課題なし");
        } else {
            for(int i=0; i<count; i++) {
                alist[i].show();
            }
        }
    }
}

```

```

    public void displayHead(int n) {
        int d;
        for(int i=0; i<count; i++) {
            d = alist[i].getDeadline();
            if(d-n<=3) {
                alist[i].setStatus("期限間近");
                alist[i].show();
            }
        }
    }
}

```

```

class Controller {
    private AList alist;
    public void display(int n) {
        System.out.println("---期限間近---");
        alist.displayHead(n);
        System.out.println("---課題リスト---");
        alist.show();
    }
    public AList getAlist() {
        return alist;
    }
    public void setAlist(AList alist) {
        this.alist = alist;
    }
}

```

```

class MainList {
    public static void main(String[] args) {
        AList ls;
        Assignment a1, a2;
        Controller c;

        a1 = new Assignment();
        a1.setTitle("PBL レポート");
        a1.setDeadline(20190109);
        a1.setStatus("課題提示");
        a2 = new Assignment();
        a2.setTitle("課題研究レポート");
        a2.setDeadline(20190118);
        a2.setStatus("課題提示");

        ls = new AList();
        ls.show();
        ls.add(a1);
        ls.add(a2);
        ls.show();

        c = new Controller();
        c.setAlist(ls);
        c.display(20190106);
    }
}

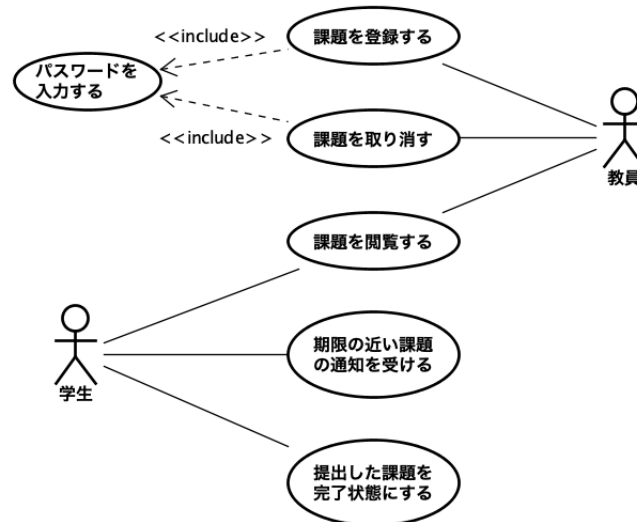
```

プログラム設計 後期期末試験模範解答（試験時間 90 分，平均 78.5 点）

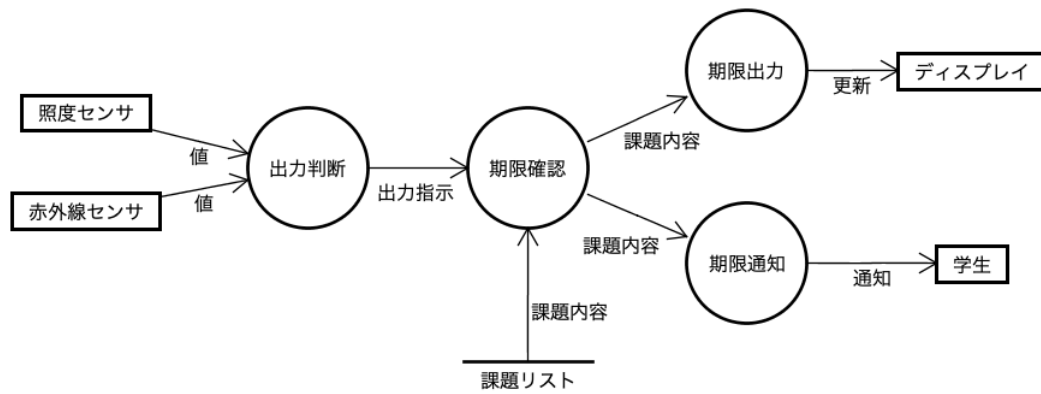
採点について 不十分な箇所につき -2 点を基本とする。（部分点については各問を参照）

1～5 各 20 点

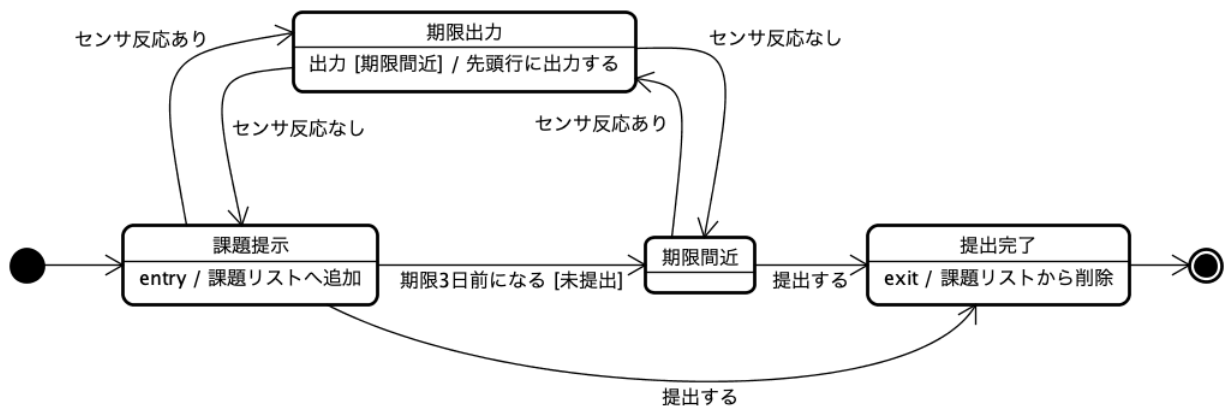
1



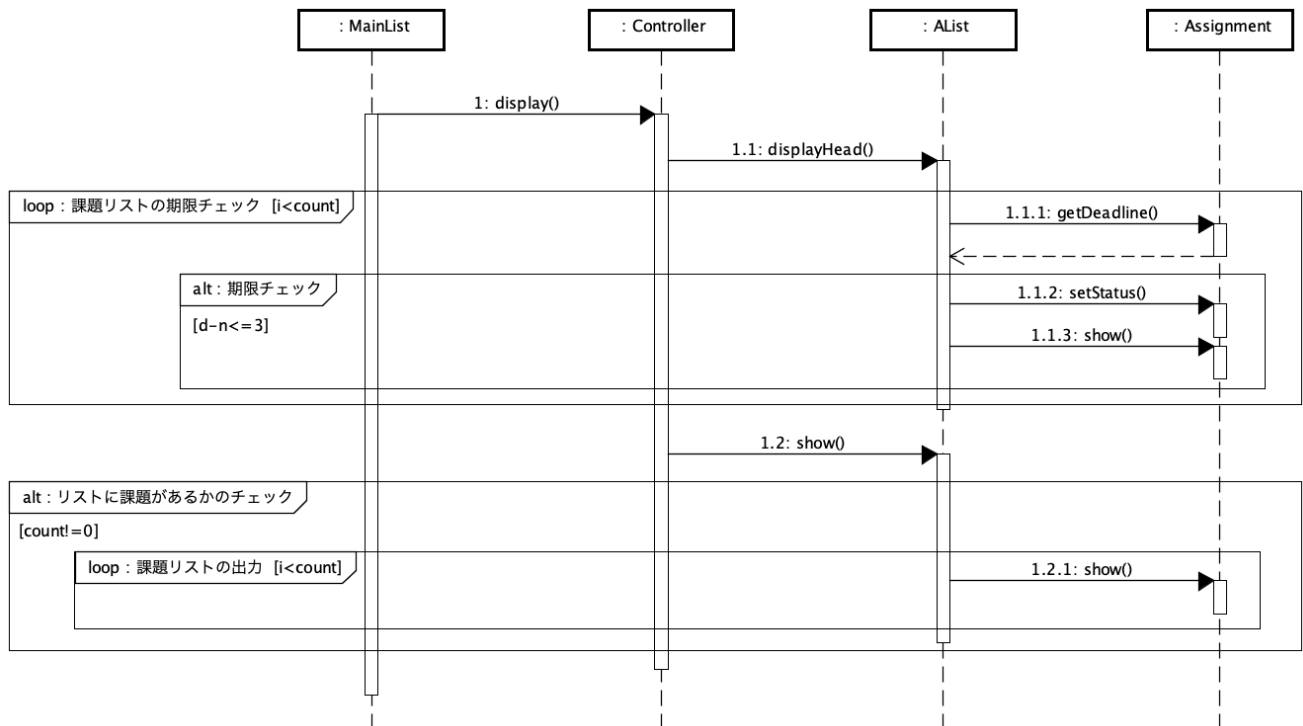
2



3



4 (ライフライン・display: 2 点, メッセージ・実行指定: 6 点, loop: 6 点, alt: 6 点)



5

