

## Command Oriented Personal Information Manager

### Assignment

#### Command Oriented Personal Information Manager

This assignment involves the creation of simple Personal Information Management system that can deal with 4 kinds of items: todo items, notes, appointments and contacts. Each of these kinds of items is described in more detail below. The assignment requires that you create a class for each item type, and that each class extends an abstract base class provided for you. In addition to creating the four classes, you need to create a manager class that supports some simple text-based commands for creating and managing items.

Each of your 4 item type classes will be derived from the following abstract class:

```
public abstract class PIMEntity {
    String Priority; // every kind of item has a priority

    // default constructor sets priority to "normal"
    PIMEntity() {
        Priority = "normal";
    }

    // priority can be established via this constructor.
    PIMEntity(String priority) {
        Priority = priority;
    }

    // accessor method for getting the priority string
    public String getPriority() {
        return Priority;
    }

    // method that changes the priority string
    public void setPriority(String p) {
        Priority = p;
    }

    // Each PIMEntity needs to be able to set all state information
    // (fields) from a single text string.
    abstract public void fromString(String s);

    // This is actually already defined by the super class
    // Object, but redefined here as abstract to make sure
    // that derived classes actually implement it
    abstract public String toString();
}
```

#### PIMTodo

Todo items must be PIMEntites defined in a class named **PIMTodo**. Each todo item must have a **priority** (a string), a **date** and a string that contains the actual **text of the todo item**.

#### PIMNote

Note items must be PIMEntites defined in a class named **PIMNote**. Each note item must have a **priority** (a string), and a string that contains the actual **text of the note**.

## PIMAppointment

Appointment items must be PIMEntites defined in a class named `PIMAppointment`. Each appointment must have a `priority` (a string), a `date` and a `description` (a string).

## PIMContact

Contact items must be PIMEntites defined in a class named `PIMContact`. Each contact item must have a `priority` (a string), and strings for each of the following: `first name`, `last name`, `email address`.

There is one additional requirement on the implementation of the 4 item classes listed above, the 2 classes that involve a date must share an *interface* that you define. You must formally create this interface and have both PIMAppointment and PIMTodo implement this interface.

## PIMManager

You must also create a class named `PIMManager` that includes a `main` and provides some way of creating and managing items (from the terminal). You must support the following commands (functionality):

- **List:** *print a list of all PIM items*
- **Create:** *add a new item*
- **Save:** *save the entire list of items (HW3: simple version, just print out; complex version, to a file, should be finished after I/O topic, to database, can be finished after JDBC topic, to a remote server, can (optional) be after Networking topic. )*
- **Load:** *read a list of items from a file*

When creating a new item it is expected that the user must response to a sequence of prompts to enter the appropriate information (and even to indicate what kind of item is being created). Do this any way you want, just make sure that your system provides enough information (instructions) so that we can use your systems!

There is no required format for the user interface, anything that allows users to create, list, save and load is fine. Here is what it might look like (user input shown in red):

```
java PIMManager

Welcome to PIM.
---Enter a command (suported commands are List Create Save Load Quit)---
List
There are 0 items.
---Enter a command (suported commands are List Create Save Load Quit)---
Create
Enter an item type ( todo, note, contact or appointment )
todo
Enter date for todo item:
04/20/2018
Enter todo text:
Submit java homework.
Enter todo priority:
urgent
---Enter a command (suported commands are List Create Save Load Quit)---
List
There are 1 items.
Item 1: TODO urgent 04/20/2018 Submit Java homework.
---Enter a command (suported commands are List Create Save Load Quit)---
Save
Items have been saved.
---Enter a command (suported commands are List Create Save Load Quit)---
Quit
```

Note that there is not a required Delete command. Feel free to use any data structure you want to hold a list of items, you are allowed to use a simple array with size 100 (you are not required to support lists of more than 100 items). Handling array bound exception is required.

We will talk about the various kinds of collection objects supported by java.util (but you don't need to use them for this assignment).

## Alternative PIMEntity

Reimplement `PIMEntity` as `interface` , add change the rest of all code to use it.

## How to submit

Submission of your homework is via huaweicloud Classroom and the general idea is to upload your files as attachments. For this Homework you should submit your .java files (remain your src directories) with a readme description inside very beginning of block comment.

The name of attachment (zip or rar file) of your message should include your **student#\_HW#\_fullname**. Anything else you want to tell us should be included in a **readme.txt** file. Make sure your submission includes your full name, student No and Email address in every single source file, we can't record your grade unless we know your name or student No.

**Don't submit compiled code (class files)!**

**Multiple Submissions:** You can resubmit for each project following the huawei Classroom instructions, we will always grade the last submission received.