Personal Information Manager (with GUI)

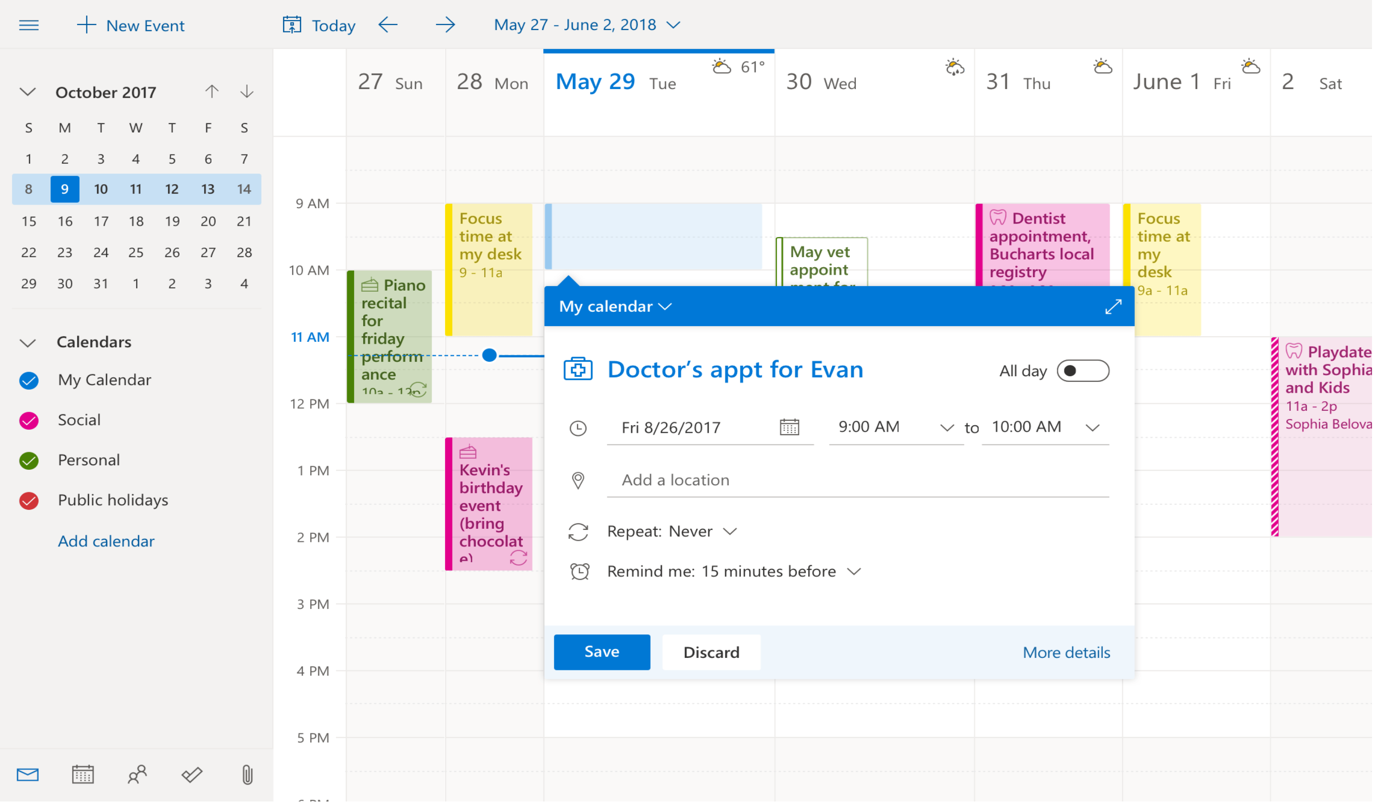
Assignment

Part 1: Java GUI Calendar with PIMDateables

This assignment involves creating a GUI Calendar that can display one month at a time. The month and year are specified on the command line or the current year and month as default.

Your calendar should show all the dates in one month, and must also display any date-able PIMEntitys (Todos and Appointments) that are retrieved from a local file or PIMCollection server (database or network server, wherever the items were stored, all of them will be encouraged selected by config options). In general you need to call getItemsForDate() for each day in the month displayed, and add the item text (just the text of the todo or appointment is fine) to the cell used to display that day.

You do not need to make sure that every item (Todo, Appointment) is completely visible in the calendar, it's OK if only part of the text appears on the right date. You do not have to provide any *clickable* items, the calendar of this Homework needs only show the days of the month and the PIMDateable items.



My calendar

Doctor's appt for Evan

All day

Fri 8/26/2017 9:00 AM to 10:00 AM

Add a location

Repeat: Never

Remind me: 15 minutes before

You are required to use the interface defined below (edit as you need) to access a storage :

```
interface RemotePIMCollection {
  public PIMCollection getNotes() throws CustomizedException;
  public PIMCollection getNotes(String owner) throws CustomizedException;
  public PIMCollection getTodos() throws CustomizedException;
  public PIMCollection getTodos(String owner) throws CustomizedException;
  public PIMCollection getAppointments() throws CustomizedException;
  public PIMCollection getAppointments(String owner) throws CustomizedException;
  public PIMCollection getContacts() throws CustomizedException;
  public PIMCollection getContacts(String owner) throws CustomizedException;
  public PIMCollection getItemsForDate(Date d) throws CustomizedException;
  public PIMCollection getItemsForDate(Date d, String owner) throws CustomizedException;
  public PIMCollection getAll() throws CustomizedException;
  public PIMCollection getAllByOwner(String owner) throws CustomizedException;
  public boolean add(PIMEntity) throws CustomizedException;
}
```

NOTE:

- Generics should be used.
- CustomizedException can be an exception you use to get your Entities. For example, if you get entities from local file, you can just use IOException.
- RemotePIMCollection is just an interface name for getting PIMCollection from a storage. It doesn't mean to use a specific technology.

PIM classes

The PIM related class definitions provided are to be used for the GUI project (a full, GUI based Personal Information Manager). The followings should be changed to the various interfaces we have used in previous assignments:

- PIMEntity should be changed so that it includes support for an Owner (a string) and a boolean that indicates whether the item is private or public. In general the simple scheme used will be that if you know the owner name, you can access a private item (This is enforced by the methods in PIMCollection, not the individual PIMEntity derived classes).
- The constructors for each of the 4 types of PIMEntitys should be changed to support the owner and shared flag.
- The fromString methods is no longer part of the interface PIMEntity.
- The toString method generates a multi-line string that is suitable for printing stuff out, but is not meant to be parsed. The general idea is that we will object serialization instead of parsing from now on...
- The PIMCollection class supports more methods, specifically there are methods to get public items (when no owner is specified), and to get public and private items corresponding to a specific owner.

Assignment

Part 2: PIM - Personal Information Manager (GUI)

The GUI project is a full-blown Personal Information Manager with same general idea (todos, appointments, etc.).

Your Personal Information Manager must support all 4 PIMEntity types, and provide (at a minimum) the following via a graphical user interface:

- A calendar that displays one month at a time, and allows users to change months (back and forward buttons are fine). All items that have dates associated with them must be displayed in the calendar. The calendar needs to meet the preview assignment's requirements and be capable of moving to another month...
- The user must be able to create new PIM items via a GUI. You decide what makes sense for this, there is no required specific type of GUI we have in mind.
- The user must be able to edit existing PIM items via a GUI.
- The user must be able to view all PIM items by item type, so there should be some way to view all Todo items, or Appointments, etc. There is no required format for the display of various kinds of items (do whatever looks nice to you).
- There must be some way for the user to save/load items, either a local file, or a shared database, or a network server shared with others.
- There must be a menubar that allows users to select from the supported functions (view calendar, view todos, view contacts, etc). The menubar should also include menu items for creating new PIM items ("new TODO", "new Appointment", etc.).
- The user should be able to specify an account name (used as the owner for PIMEntities) once, not each time an item is created. The username can be prompted each time the program starts, it can also be saved in some kind of configuration file (or specified on the command line). The important issue is that the user is not required to specify the owner every time a new entity is created.
- For PIMDateable objects (PIMTodo and PIMAppointment), you don't need to worry about times, a single date (month, day and year) is all that you are required to deal with.
- All source code written by you must be commented with Javadoc comments (and whatever other comments you think are appropriate).

Java Technology Homework Submission

For All Assignments

How To Submit

Electronic Submission Only!

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.