

PCS3 Exam

Date: Wednesday, April 8, 2015 .

Time: 8.45 – 11.45 h (3 hours).

Rules:

Admitted resources:

- You are allowed to use everything on paper (books, notes, etc.) and on your laptop, but only what you bring in: you are not allowed to borrow something from someone else.
- During the exam it is not allowed to use the network. You should make the exam yourself: so no communication with msdn or google for help and no communication with other students, like communication by Facebook, e-mail, Skype, gsm or whatever.

Way of working during the exam and handing in your exam: see document "Unimpeded testing (Rustig toetsen) Step by Step".

Remark: the displayed remaining time, as shown at "the finish flag of Unimpeded testing", might be wrong.

Grading: see assignments.

The last assignment can be skipped in case you have been allocated as an 'active' student for PCS3 this block. You will get the corresponding amount of 10 points for this assignment as a bonus.

Preliminary remarks:

- Never duplicate code if there is a better way to achieve the same result.
- Whenever this exam paper suggests to use a certain name for a method, variable or anything else, you are required to indeed use that name.
- If you think you need more class-members, you are allowed to add more members to the classes than specified in this exam.
- In this exam we use the word "app", which is an abbreviation for "application" or "program".

Introduction: Controlling my news-items for my website.

The main purpose of this app is to control news-items. With this app it must be easy to add, remove or update news-items. In this app we consider only two types of news-items, but in future there could be more kind of news-items.

In this assignment we will deal with two kind of items: single-line-items and multi-line-items. The body-text of a single-line-item is exactly one string, while the body-text of a multi-line-item can have any number of strings.

This app must have the functionality to generate an html-file. This generated html-file is a text-file, but the layout is in such a way, that you can easily integrate it in your website.

The layout of the generated file must be according to the html-standard. Don't be afraid if you are not familiar with html, what you need to know about it for this assignment, will be explained later.

Below you see an example of such a generated file. First you see the file opened in a browser, like Chrome or Firefox. Secondly you see the contents of the same file, opened by a tool like Notepad.

An example: when you open the file “example.html” with a browser, you see (the black text):

Pinkpop 2015, latest news

Will Miley Cyrus be on Pinkpop?

Most probably she will perform at Pinkpop.

Posted by: Carla, 24-2-2015

About the line-up: Pharrell: yes ; Katy: still doubting.

- I'm happy: Pharrell Williams will be the top act on Friday. Come and see him. It's all about the hat.
- Miley Cyrus canceled since she did not pay attention when the wrecking ball chain broke.
- We are still negotiating with Katy Perry. We still believe she will be one of the top acts this year.

Posted by: Vince, 3-3-2015

Katy Perry, still no decision.

Sorry folks, not 100% sure Katy will roar for you at Pinkpop!

Posted by: Arthur, 13-3-2015

Finally Katy sais YES, she will come to Pinkpop.

Good news: today we contracted Katy Perry. Are you ready? She promised to enter the stage riding a dark horse!

Posted by: Carla, 28-3-2015

Annotations on the right:

- Blue line: Title of the webpart
- Red line: Single line item
- Green line: Multi line item
- Red line: Single line item
- Red line: Single line item

This webpart consists of a title (in this example: "Pinkpop 2015, latest news"), then a horizontal line and then 4 items: first a single-line item, then a multi-line item, and then 2 single-line items.

For every single-line-item you see

- the title of the item, displayed with a bigger font (in html: as `<h2>`),
- the body text of the item displayed with the normal font (in html: as `<p>`)
- the footer “Posted by: <name> , <posting date> .” displayed with a smaller font (in html: as `<h5>`),
- horizontal line (in html: `<hr>`).

For a multi-line-item you see almost the same, except for the body text. Since the body text consists of several lines, every line is displayed bulleted (in html: as ``).

When you open the same file “example.html” with Notepad, you see (for readability some empty lines are added, but you can skip them if you like):

```

<h1>Pinkpop 2015, latest news</h1>
<hr>

<h2>Will Miley Cyrus be on Pinkpop?</h2>
<p>Most probably she will perform at Pinkpop.</p>
<h5>Posted by: Carla, 24-2-2015</h5>
<hr>

<h2>About the line-up: Pharrell: yes ; Katy: still doubting.</h2>
<p><ul>
<li>I'm happy: Pharrell Williams will be the top act on Friday. Come and see him. It's all about the hat.</li>
<li>Miley Cyrus canceled since she did not pay attention when the wrecking ball chain broke.</li>
<li>We are still negotiating with Katy Perry. We still believe she will be one of the top acts this year.</li>
</ul></p>
<h5>Posted by: Vince, 3-3-2015</h5>
<hr>

<h2>Katy Perry, still no decision.</h2>
<p>Sorry folks, not 100% sure Katy will roar for you at Pinkpop!</p>
<h5>Posted by: Arthur, 13-3-2015</h5>
<hr>

<h2>Finally Katy says YES, she will come to Pinkpop.</h2>
<p>Good news: today we contracted Katy Perry. Are you ready? She promised to enter the stage riding a dark horse!</p>
<h5>Posted by: Carla, 28-3-2015</h5>
<hr>

```

Startup project

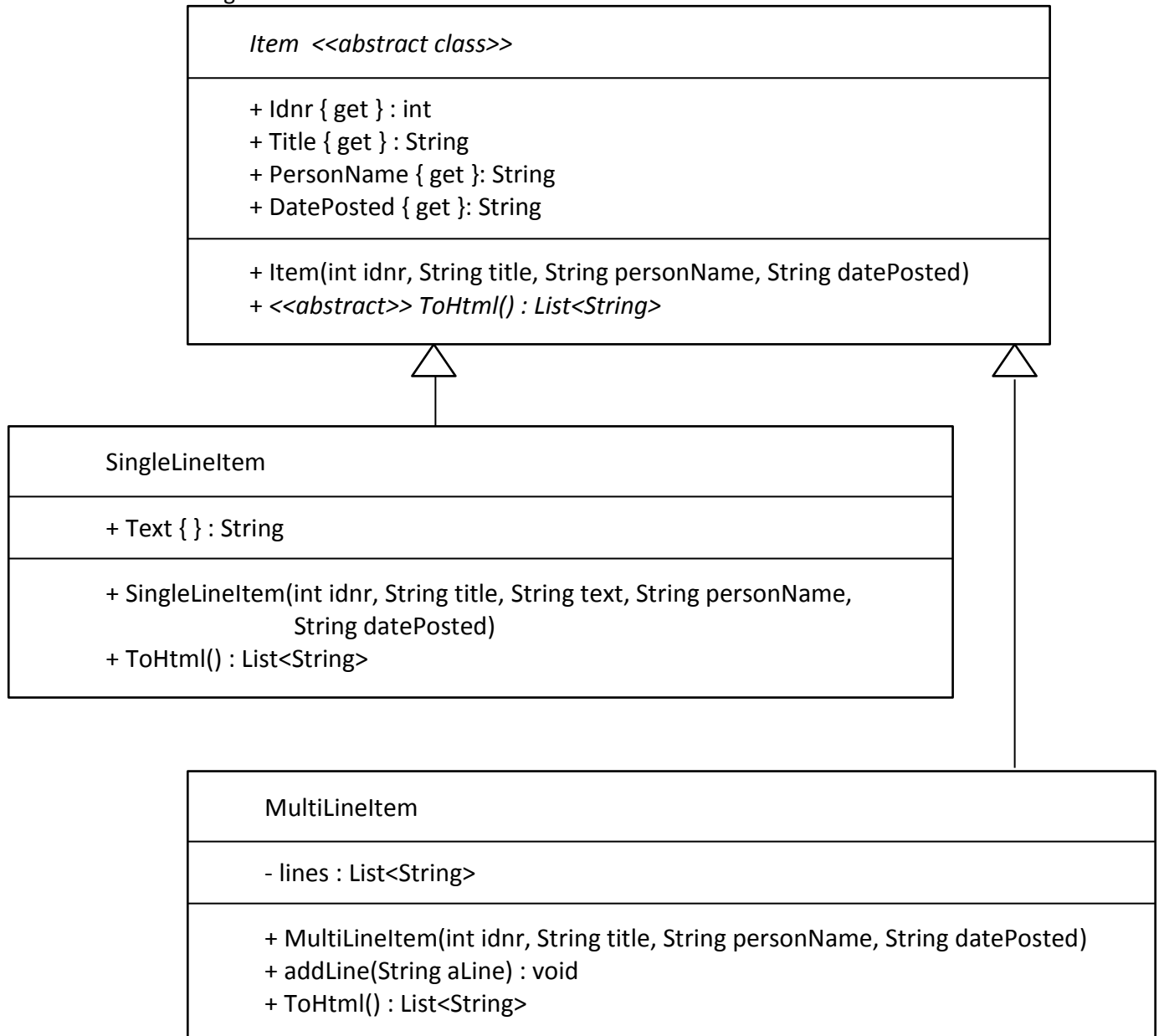
The startup-project is already created and the user interface (Form1) is already filled with controls. The user interface looks like:

The screenshot shows a Windows application window titled "Controlling my news-items, made by <<YOUR NAME>>". The window contains a large empty rectangular area on the left, likely for displaying news items. On the right, there is a form with several input fields and buttons. The form fields are: "id" (25), "title" (Entrance open at. . .), "line of text" (The entrance will be opened at 9.00 h.), "posted by" (Bert), and "date of post" (6-4-2015). Below these fields are two radio buttons: "want to add a singleline-item" (selected) and "want to add a multiline-item". At the bottom of the form are two buttons: "Add an item" and "Add a line to a Multiline-item". At the bottom of the window are three buttons: "Show all", "Show all Singleline-items", and "Export to Html".

Assignments:

Assignment 1: The classes `Item`, `SingleLineItem`, `MultiLineItem` (25 points)

Consider the class diagram:



As you can see in the class diagram above

The class `Item`

- has an `idnr` (a unique identity number), a `title`, a `personName` (the name of the person who posted it) and a `datePosted` (for simplicity: a string. It contains the date it is posted, but you do not need to do any checking if it represents a correct date). In the class diagram these members are depicted as property's with only getters, but you may choose how to implement these members: as fields or as (auto-implemented) property's : it is your choice.
- Has a constructor and an abstract method `ToHtml()`, which returns a list of strings.

The classes SingleLineItem and MultiLineItem inherit from Item.

The class SingleLineItem

- has a field to store the main text of this SingleLineItem-object.
- Has a constructor with 5 parameters, and
- Should have an implementation for the method ToHtml(), which should return a list of strings. These strings should be build according to the html-format. These strings should hold the information:
"<h2>", then the title, then "</h2>",
"<p>", then the text, then "</p>",
"<h5>Posted by: ", then the personName and the datePosted, then "</h5>",
"<hr>"

The class MultiLineItem

- has a field to store the lines of the body-text of this MultiLineItem-object.
- Has a constructor with 4 parameters,
- Has a method addLine(string aLine) to add one line to the body-text, and
- Should have an implementation for the method ToHtml(), which should return a list of strings. These strings should be build according to the html-format. These strings should hold the information:
"<h2>", then the title, then "</h2>",
"<p>",
For every string in the list lines the text "", then the string, then "",
"</p>"
"<h5>Posted by: ", then the personName and the datePosted, then "</h5>",
"<hr>"

Implement the classes Item, SingleLineItem and MultiLineItem.

Assignment 2: The WebPart-class (15 points)

The class WebPart has a list to store its items. Furthermore it has a title (for instance: " Pinkpop 2015, latest news ").

WebPart
<ul style="list-style-type: none">- myItems : List<Item>- title : String
<ul style="list-style-type: none">+ WebPart (String t)+ GetAllItems () : List<Item>+ GetItem (int id) : Item+ AddItem (Item i) : bool+ GetListOfAllSingleLineItems () : List<Item>

The constructor and methods should be implemented as follows (you do not need to copy all these annotations to your Visual-Studio-project):

```

    /// <summary>
    /// creates a WebPart-object with title t
    /// Its list myItems is empty
    /// </summary>
    /// <returns></returns>
    public WebPart(String t)

    /// <summary>
    /// returns a list of all items
    /// </summary>
    /// <returns></returns>
    public List<Item> GetAllItems()

    /// <summary>
    /// If the webpart-object has an item with Idnr equal to id,
    /// then it returns that Item-object with Idnr equal to id.
    /// If not, it returns null
    /// </summary>
    /// <param name="id"></param>
    /// <returns></returns>
    public Item getItem(int id)

    /// <summary>
    /// If the webpart-object does not have an item with Idnr equal to
i.Idnr,
    /// then i is added to the list and it returns true.
    /// Otherwise nothing is added and it returns false
    /// </summary>
    /// <param name="i"></param>
    /// <returns></returns>
    public bool AddItem(Item i)

    /// <summary>
    /// returns a list of all items, which are of type SingleLineItem
    /// </summary>
    /// <returns></returns>
    public List<Item> GetListOfAllSingleLineItems()

```

Implement the class WebPart.

By the way, if you need more members in a certain class than showed in the class diagrams, you are free to add them.

Assignment 3: The user interface Form1.

Now we start with the class Form1. To make it easy for you to test your own code, we added already some commented lines. It is about the variable `myWebPart` and the method `PutSomeTestingStuffInTheWebPart()` to add some items in `myWebPart`. If you uncomment these lines of code, you get the situation as in the former example. You may use it, but you may also throw it away; it is up to you to use it or not.

Assignment 3a: The "Show all" button (8 points)

Add a variable `myWebPart` of type `WebPart` to it and create it at startup (Hint: uncomment some lines).

Implement the code for the "Show all"-button in such a way that it shows the title of `myWebPart` and shows information of all its items in the listbox.

There are several ways in how to show it: you may show every item as exactly one line in the listbox, or you may show every item on several lines, or you may show it as strings build according to the html-format (using the `ToHtml`-method).

How you show it, that is up to you.

Assignment 3b: The "Add an item" button (15 points)

Implement the button-click of the "Add an item"-button.

You should add a single-line-item or a multi-line-item, depending on which radiobutton is checked.

You can use the textboxes on the misty-rose-colored panel on the right side of the window to specify the fields of the newly to be added item.

For a new single-line-item, you need all textboxes.

For a new multi-line-item, you do not need the textbox for a line of text, since after creating a new multi-line-item, this new item has no lines yet.

Be aware: For the newly to be added item: the `idnr` must be unique: it is not allowed that `myWebPart` contains items with the same `idnr`.

You may assume that the text in textbox `tbId` can be converted to an integer without problems. You do not need to check it.

Assignment 3c: The "Add a line to a Multiline-item" button (10 points)

Implement the button-click of the "Add a line to a Multiline-item"-button.

Clicking this button should add a line (from the textbox `tbLineOfText`) to the Multiline-item with `idnr` equal to the number in textbox `tbId`. If not possible, it should display a proper message.

Assignment 3d: The "Show all Singleline-items" button (5 points)

Implement the code for the "Show all Singleline-items"-button in such a way that it shows information of all singleline-items in the listbox. Like in assignment 3a: how you show it, that is up to you.

Assignment 3e: The "Export to html-file" button (12 points)

Clicking the "Export to html"-button should result in:

- The app pops up a `saveDialog`.
- The user can choose a filename.
- The app saves the information of `myWebPart` in a file with the chosen filename. This file contains the title of `myWebPart` and all its news-items. The layout of the generated file must be according to the html-standard, as described earlier.

Implement the button-click of the "Export to Html"-button. Your code has to handle all kind of possible exceptions.

Hint: if you choose to save it to a file with extension ".html", you can easily open the file with your browser by double-clicking on it.

Assignment 4: Extra (10 points) You can skip this assignment if you earned the bonus points

Add two buttons to the user interface with texts "save" and "load" on it. Clicking the save-button results in saving the object myWebPart to a file. Clicking the load-button results in retrieving the object myWebPart from the contents a file.

--- END OF THE EXAM ---