

PCS4 Exam – Jan 2017

Date: January 20, 2017

Starting time: 12.45 h, duration 150 minutes.

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 using facebook, e-mail, skype, Dropbox, gsm or whatever.

GRADING:

| Assignment | | 1 | 2 | 3 | 4 | total |
|-------------|--|----|----|----|----|-------|
| Max. points | | 20 | 30 | 24 | 26 | 100 |

THE APPLICATION

Many times young people have marvelous ideas but not enough money to realize their dreams. A company wants to help these young people by collecting money so they can start their own startup-company to realize their dreams.

By crowdfunding, persons who would like to help these young starters, can donate money. Every startup-company has a threshold (an amount of money that is minimally needed to start). If the sum of all their donations is equal to or bigger than their threshold, they can start their startup-company.

Of course, the donators will get a discount for buying things of the startup's they sponsored (but that is beyond the scope of this exam).

The company would like to have an application to keep track of the startups and the donated money. The application also keeps track of thresholds, i.e. a minimum amount of money that a startup should have to start his/her company.

There are 2 companies interested in young startups. One of them is CapitolStudio, who is interested in music. This company is interested in all startups that have "music" in their description.

The other company, named RedFlat, is interested in startups who have a threshold of at least 10.000 euro.

The application consists of 3 windows: a window for the company itself (of type FormCrowdfunding), a window for CapitolStudio and a window for RedFlat. At startup all windows are created. It is possible that these windows overlap each other. Feel free to position them on the screen wherever you want.

Furthermore there are some other classes with already some code in it (see startup-project).

PRELIMINARY REMARKS:

- 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.
- The given classes already contain some code. It will be your task to add more code to these classes, if needed. You may add whatever you need. It will also be your task to add more classes, if needed.

The next picture shows the FormCrowdfunding -window:

Crowdfunding

about sorting

sort on total donations sort on number of donations/name

about startups and donations

name startup: Trickster

description:

website:

threshold:

name donator: Beyonce

amount donation: 2000

add a startup

add donation

about assignment 3

about assignment 4

show linked list

show all startups

label7 label8 label9

Moonlite, donations: 650, threshold: 7000, a bedtime story projector for mobile phone
ForeverMine, donations: 500, threshold: 10000, a laptop out of building blocks
Trickster, donations: 13900, threshold: 15000, interactive 3D-movie with music
Brave, donations: 1500, threshold: 8000, musicshow

As you can see in the above picture the FormCrowdfunding window has several buttons. The functionality of some buttons has been partially implemented. It will be your task to complete the functionality for these buttons and to add functionality for the other buttons. For the details, see the assignments.

By the way, to help you in testing your code, we added a method

`private void addSomeTestingStuff()`

which will add some startups and donations as shown in the picture above. Using this method results in the situation above. Using this testing-method is up to you. You may also skip it or add your own testing stuff. It will not cost you points and it will not give you points.

Assignment 1 (10+10 pts).

Consider the two buttons with text "sort on total donations" and "sort on number of donations/name" on this window.

In the pcs4-lessons you learned 4 ways to sort elements in a list. First you did it yourself. Then we had 3 ways: by using IComparable, Comparison or IComparer.

Assignment 1a:

Pushing the button with text "Sort descending on total donations " must sort the startups from the startup with the highest total amount of donated money to the startup with the lowest total amount of donated money.

As you can see in the FormCrowdfunding-file this button-click is already implemented by calling:

```
this.cf.SortOnTotalDonations();
```

The problem is solved by implementing the method

```
public void SortOnTotalDonations()
```

in the Crowdfunding-class.

Give an implementation by using one of the last 3 ways (so by using IComparable, Comparison or IComparer).

Assignment 1b:

Pushing the button with text " sort on number of donations/name " must sort the startups by "number of donations – name". The startups must be sorted by number of donations. If startups have the same number of donations, then they should be sorted alphabetically on name.

Again, as you can see in the FormCrowdfunding -file this button-click is already implemented by calling:

```
this.cf.sortOnNumberOfDonations_Name();
```

Make it work by using one of the last 3 ways, but **not** in the same way as you did in the former assignment (not doing with one of the other ways will be rewarded with zero points).

Assignment 2 (30 pts).

The buttons with text "add a startup" and "add donation" are partly implemented.

Implemented is the following functionality:

- Clicking the button with text "add a startup" first checks if there is already a startup with name as specified in the corresponding textbox, because every startup-object must have a unique name. If there is no startup with the specified name, a startup-object is created and added to the list of startups. If there is a startup with the specified name, a proper message is displayed.
- Clicking the button with text "add donation" will check if there is a startup-object with name as specified in the corresponding textbox. If so, a donation-object is created and

added to the list of donations of that startup. If there is no startup with the specified name, a proper message is displayed.

Remember, there are 2 other companies that should be notified if a certain startup reached its threshold.

Capitol Studio is notified everytime a startup, that has "music" as substring in its description, reaches its threshold.

RedFlat is notified everytime a startup, that has a threshold of at least 10.000 euro, reaches its threshold.

Capitol Studio must be notified on `frmCapitolStudio`, and RedFlat on `frmRedFlat`.

The `FormCrowdfunding` -class has 3 variables:

```
private Crowdfunding cf; //the crowdfunding-object that has a list of startups
private FormCapitolStudio frmCapitalStudio; //the window for Capitol Studio
private FormRedFlat frmRedFlat; //the window for RedFlat
```

For Capitol Studio, the class `FormCapitalStudio` has a method

```
public void ShowStartupInMyListbox(Startup startup)
```

to show some information of startup in the listbox on his window (is already implemented).

For RedFlat, the class `FormRedFlat` has a method

```
public void AlertMe(Startup su)
```

to show information about su blinking on her window (is already implemented).

Furthermore, for every startup `st` that reaches its threshold, some information must be displayed in the red area at the bottom of the main window. For this purpose we have the method

```
public void congratulations(Startup st)
```

In the `FormCrowdfunding`-class (is already implemented).

Assignment:

Adjust the code in such a way, that the above mentioned functionality works correctly.

You must give a solution by using events. An implementation without using events will not be rewarded.

Example as in the picture above:

Consider the situation as depicted in the screenshot above.

When the "add donation"-button is clicked, a donation, as specified in the textboxes on the main window, should be added. So in this case a donation of 2000 euro, donated by Beyonce, must be added to startup Trickster. That means that the total amount of donations exceeds Trickster's threshold of 15.000 euro. So information about Trickster must be displayed in the red area at the bottom of the main window. Since its description contains "music", RedFlat must be informed. And because of and its threshold is bigger than 10.000 euro, Capitol Studio must also be informed.

Assignment 3 (12 + 12 pts).

Assignment 3a:

The class Startup has a method

```
public int getSumOfDonationsAbove(int limit, int n)
```

The method will return the sum of the first n donations in the list that are bigger than limit.

Give a recursive implementation of this method (a non-recursive solution will not be rewarded).

You may assume that $n \geq 0$ && $n \leq \text{this.donations.Count}$.

Example:

If you use the method `addSomeTestingStuff()` the startup Brave has a list with 7 donations of resp 300, 150, 350, 100, 100, 400 and 100 euro.

Executing the method-call `getSumOfDonationsAbove(250, 3)` returns the value 650, since 2 of the first 3 donations are higher than 250 euro, and their sum is $300 + 350 = 650$.

P.S.: we do not ask to test this method. If you would like to test it, you may add buttons, textboxes or whatever you need to the groupbox with text "about assignment 3", but you will not get any points for it.

Assignment 3b:

Two consecutive donations in a list are twins if the sum of their amount is equal to 500. So, for the example in assignment 3a, we have:

- Consider the first two donations: `this.donations[0].Amount + this.donations[1].Amount = 300+150`, sum is not equal to 500, so not a twin
- Consider the next two donations: $150+350$, sum equals 500, so a twin
- $350+100$, sum is not equal to 500, so not a twin
- $100+100$, sum is not equal to 500, so not a twin
- $100+400$, sum equals 500, so a twin
- $400+100$, sum equals 500, so a twin

The class Startup has a method

```
public int getNumberOfTwins(int n)
```

The method will return how many twins there are in the first n donations in the list.

Give a recursive implementation of this method (a non-recursive solution will not be rewarded).

You may assume that $n \geq 0$ && $n \leq \text{this.donations.Count}$.

Example:

Executing the method-call `getNumberOfTwins(7)` returns the value 3.

Executing the method-call `getNumberOfTwins(6)` returns the value 2.

Executing the method-call `getNumberOfTwins(1)` returns the value 0.

P.S.: again, like in assignment 3a, we do not ask to test this method. If you would like to test it, you may add buttons, textboxes or whatever you need to the groupbox with text "about assignment 3", but you will not get any points for it.

Assignment 4 (18 + 8 pts).

For this assignment you need to implement a linked list. The items of the linked list must be objects, containing information about big donations. Such an item must at least have

- a field to store the name of the startup,
- a field to store the name of the donator,
- a field to store the amount of money of the donation,
- and a method that returns a string with information about these 3 fields.

We want to keep track of at most 10 of the most recent big donations. A big donation is a donation of 1000 euro or more.

In the linked list the most recent donation will be at the front (the beginning) of the linked list and the oldest will be at the end of the linked list. (the items in the linked list are ordered chronologically).

For every new donation, you must first check if it is 1000 euro or more. If so, then there are 2 options:

- if the linked list has less than 10 items, add an item about it at the front of the linked list,
- if the linked list has 10 items, add an item about it at the front of the linked list and remove the oldest item.

Assignment 4a:

Add classes to the project to get the linked list working as specified above.

There must be a method

`public void addDonationToLinkedList(String startupName, String donatorName, int amount)`
which is implemented as described above.

Assignment 4b:

Clicking the button with text “show linked list” shows information of the items of the linked list chronologically on the screen (the most recent first). Make it work!

End of exam.