Assignment 1 – Bookshop revisited

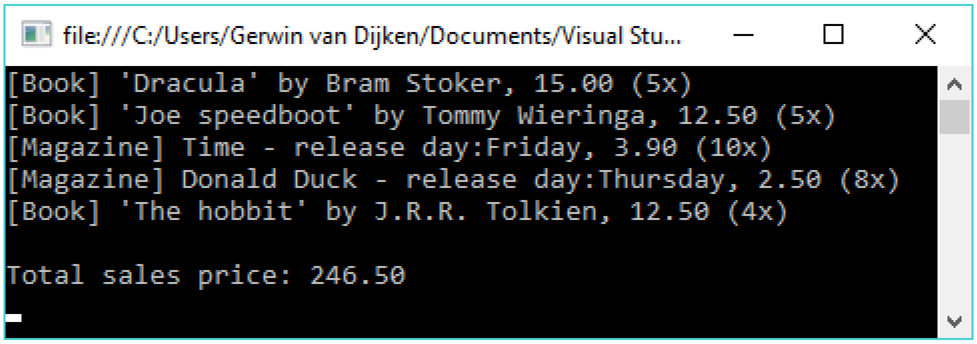
We will revisit the Bookshop of last week, improving it by using properties (instead of public members).

a) Modify (abstract) class BookStoreItem by converting the public members (title and price) into properties. Add an extra property 'Count' that contains the number of copies of a bookstore item. Also add a 'calculated' (read only) property 'TotalPrice', that returns the total sales price of all available copies.

b) Modify the classes Book and Magazine by converting the public members into properties. Property Author (class Book) must be a read only property, since the author of a book does not change.

c) Modify method 'PrintCompleteStock' of class BookStore: instead of using the price of each item, use the 'total price' (of all copies of an item). Furthermore, make sure the list with items is private, we don't want the list to be accessible by other classes.

d) In de Start method, create some books and magazines, and add these to the bookstore. You can use the screenshot below, to check your own program (pay special attention to the total price of the complete stock).



```
[Book] 'Dracula' by Bram Stoker, 15.00 (5x)
[Book] 'Joe speedboot' by Tommy Wieringa, 12.50 (5x)
[Magazine] Time - release day:Friday, 3.90 (10x)
[Magazine] Donald Duck - release day:Thursday, 2.50 (8x)
[Book] 'The hobbit' by J.R.R. Tolkien, 12.50 (4x)

Total sales price: 246.50
```

Assignment 2 - Cinema

In this assignment we will create some classes for a reservation system, for booking cinema tickets. We don’t implement a complete systeem, only the classes Customer, Ticket and Reservation.

class Customer

Of each customer, we want to record the name, the age and the date of registration. Young customers can not book all movies. Furthermore, customers get a discount when they have been registered for at least a year.

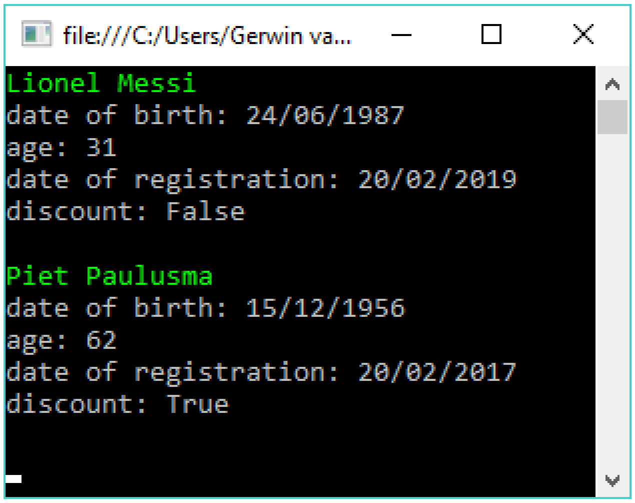
a) Create a property for name and date of registration. Date of registration is the same as the date when the Customer object is created, and can not changed afterwards. The name is only updated when a valid name is given (not empty).

b) Since the age of a customer will of course change, we will not store the age it self, but the date of birth. Create a property DateOfBirth, allowing only valid values (date of birth must be in the past, and must be a realistic date).

c) Create a calculated int-property Age that uses the date of birth. Create a calculated bool-property Discount that uses the date of registration. For both properties, use DateTime.Today to get the current date.

d) In the Start method, check if all properties work correctly for several customers. Some customers will have had their birthday already this year, other customer will have it later this year. Also check incorrect input (like an empty name, or a ‘future’ date of birth), and what happens when you try to change the (read only) date of registration.

e) Create a separate method PrintCustomer (with a Customer parameter) to print a customer. You can see an example of the output below.



```
Lionel Messi
date of birth: 24/06/1987
age: 31
date of registration: 20/02/2019
discount: False

Piet Paulusma
date of birth: 15/12/1956
age: 62
date of registration: 20/02/2017
discount: True
```

class Ticket

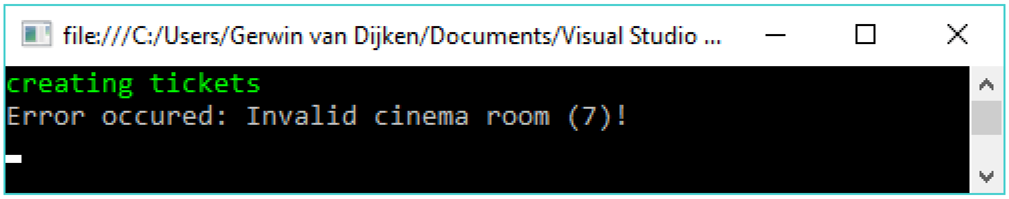
Class Ticket has the following properties: (string) MovieName, (int) CinemaRoom, (DateTime) StartTime, (decimal) Price, (int) MinimumAge.

a) Create the properties and make sure that the following restrictions are met:
- the name of the movie is not empty;
- there are 5 cinema rooms: 1, 2, 3, 4 and 5;
- each movie starts on the hour or on the half hour;
- the minimum age can be one of the following values: 0, 6, 9, 12, 16, where 0 means ‘for all ages’;

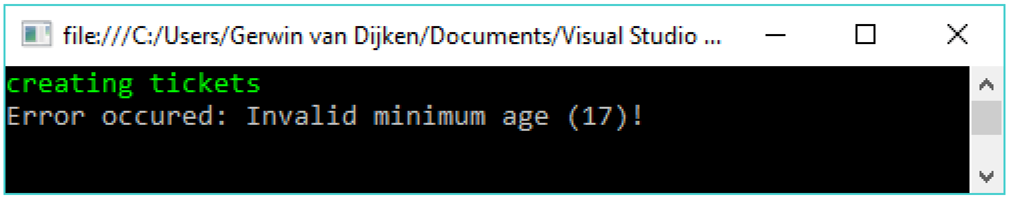
(throw an exception if an invalid value is used for these properties)

b) Create a calculated property Discount (in class Ticket), that indicates if a discount will be given for the ticket. A discount is given for all movies on a Monday or Tuesday. *(use property DayOfWeek of class DateTime)*

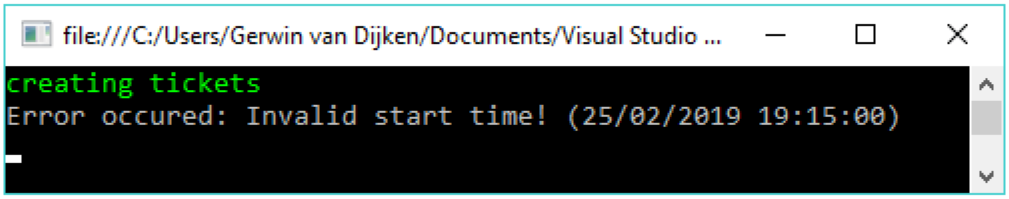
c) In the Start method, check if all properties work correctly for several tickets. Check the following cases: an empty movie name, an invalid cinema room, and invalid minimum age and an invalid start time.


file:///C:/Users/Gerwin van Dijken/Documents/Visual Studio ...

```
creating tickets
Error occured: Invalid cinema room (7)!
```


file:///C:/Users/Gerwin van Dijken/Documents/Visual Studio ...

```
creating tickets
Error occured: Invalid minimum age (17)!
```


file:///C:/Users/Gerwin van Dijken/Documents/Visual Studio ...

```
creating tickets
Error occured: Invalid start time! (25/02/2019 19:15:00)
```

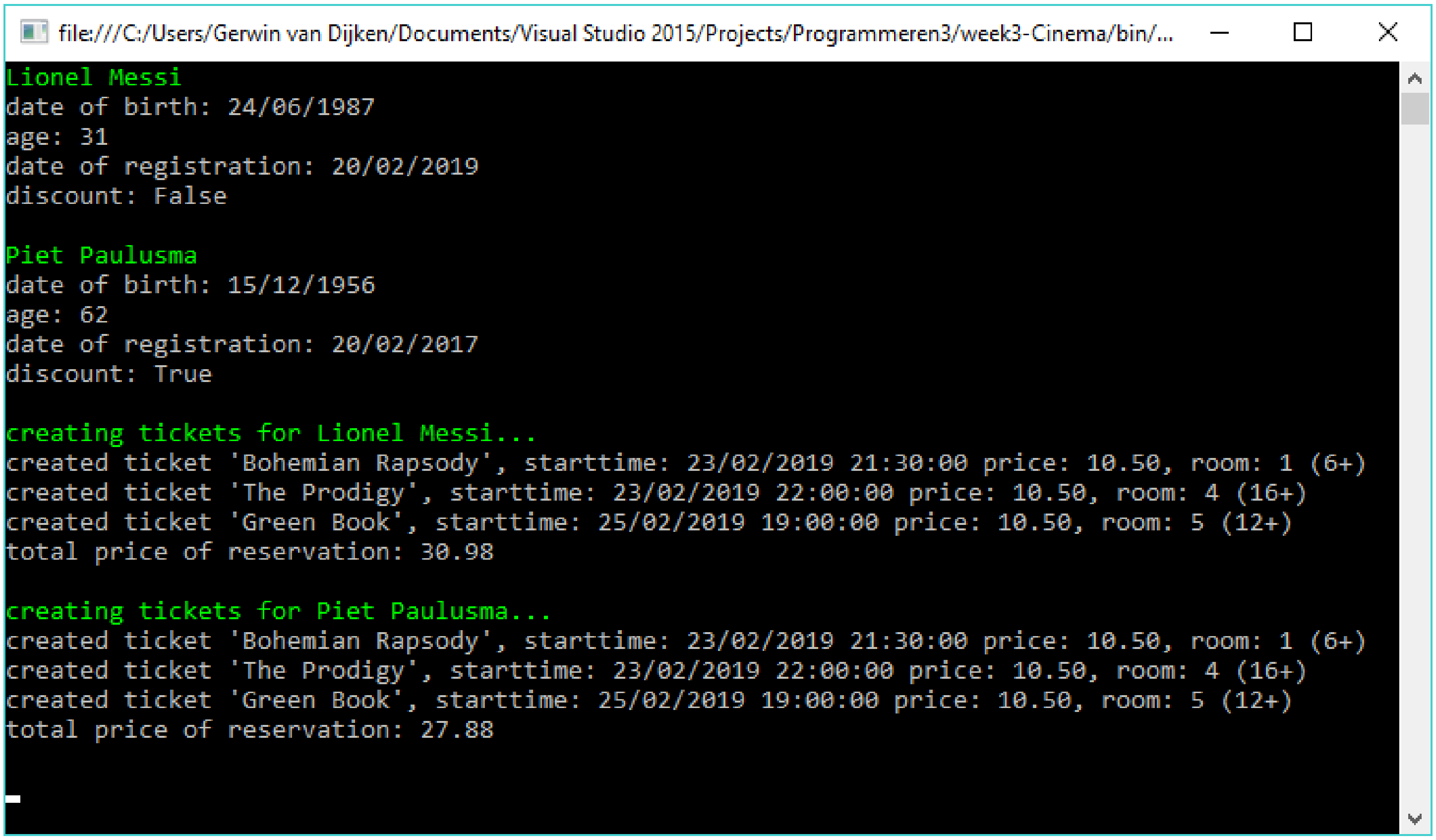
class Reservation

Class Reservering is linked to a customer, and can have several tickets.

a) Create property Customer (the customer of the reservation) and a property Tickets (a list containing all tickets of the reservation), tickets can be added to this list;

b) Create a calculated property TotalPrice, that returns the total price of all tickets in the list. This property checks the minimum age, and uses the discount of the tickets and the discount of the customer. The discount for a ticket is 5%. The discount for 'regular customers' *(registered for at least a year)* is 10% and is given on the total price (of all tickets);

c) In the Start method, create some reservations and check if the total price is calculated correctly. An example of the output is given below.
(the movie 'Green Book' is on a Monday: Februari 25th 2019, so a discount of 5% is applied; Piet Paulusma have been registered for more than a year, so a discount of 10% is applied for the complete reservation)



file:///C:/Users/Gerwin van Dijken/Documents/Visual Studio 2015/Projects/Programmeren3/week3-Cinema/bin/...

```
Lionel Messi
date of birth: 24/06/1987
age: 31
date of registration: 20/02/2019
discount: False

Piet Paulusma
date of birth: 15/12/1956
age: 62
date of registration: 20/02/2017
discount: True

creating tickets for Lionel Messi...
created ticket 'Bohemian Rapsody', starttime: 23/02/2019 21:30:00 price: 10.50, room: 1 (6+)
created ticket 'The Prodigy', starttime: 23/02/2019 22:00:00 price: 10.50, room: 4 (16+)
created ticket 'Green Book', starttime: 25/02/2019 19:00:00 price: 10.50, room: 5 (12+)
total price of reservation: 30.98

creating tickets for Piet Paulusma...
created ticket 'Bohemian Rapsody', starttime: 23/02/2019 21:30:00 price: 10.50, room: 1 (6+)
created ticket 'The Prodigy', starttime: 23/02/2019 22:00:00 price: 10.50, room: 4 (16+)
created ticket 'Green Book', starttime: 25/02/2019 19:00:00 price: 10.50, room: 5 (12+)
total price of reservation: 27.88
```