SWE201 - CSE215 Object Oriented Programming Midterm and Final Project 2022-2023 Fall

Project Description V1: 08.11.2021

Project Delivery: End of Finals

Please read this document at least 2 times

Multi-Threaded Robust Responsive Web Crawler Design

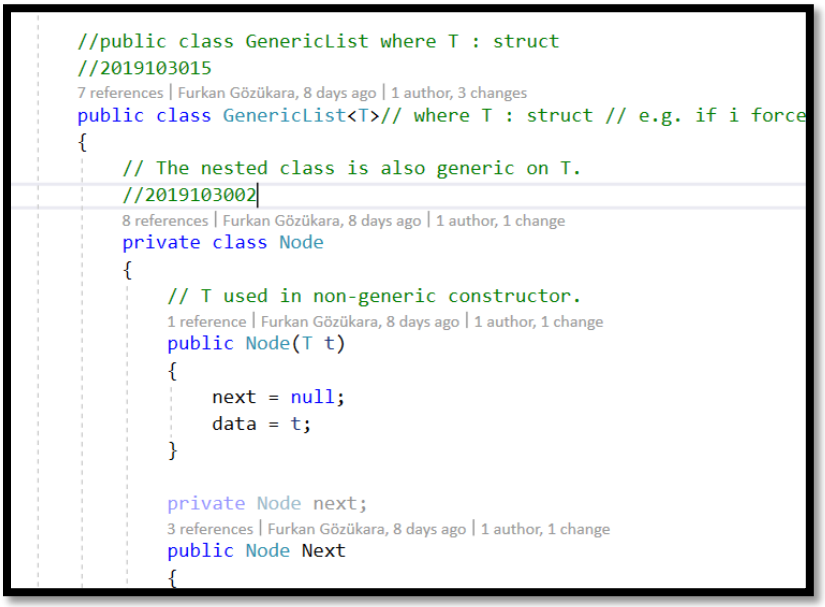
There will be no exams for both midterms or finals. Therefore, your score will be 100% given from this project. In this project, you are supposed to develop a desktop-based multi-threaded web crawler using some sort of database. You are not bound to use a certain programming language (you can use c#, java, etc.) but it has to have an interface and has to be a desktop application. Your score will be determined by how many of the necessary features you have achieved to develop.

1. Supporting crawl of multiple websites at the same time
2. An option to be able to set crawl external URLs or not
3. An option to be able to set how many threads will the software use to crawl each website
    a. This is load balancing and very important. Let's say we have 10 different websites to crawl. So the software should be able to start let's say 20 crawling tasks on each website
    b. You need to be able to display how many threads are crawling each website at any given time

4. Using some sort of database to keep track of crawled URLs, waiting to crawl URLs, failed URLs, retry counts, last crawl date, title, description of the crawled URLs, and inner text of crawled URLs.
    a. Preferred databases are MSSQL (Developer Edition), Oracle, MYSQL, PostgreSQL, and such
5. No errors should ever crash the software and all of the errors must be handled and logged gracefully
6. The user interface has to be fluent and responsive no matter how many crawling tasks are started
7. Start, pause and stop buttons on the user interface
8. An input box to add new root URLs to the crawling task
9. A screen to display the running tasks count for each website. This can be like multi-tab or a new WPF screen opened via button click
    a. In this case, it should also show other task statuses. For example for c#: Canceled, Created, Faulted, RanToCompletion, Running, WaitingForActivation, WaitingForChildrenToComplete, WaitingToRun
10. A screen to display the latest crawled URLs and errors if happened
11. A screen to display run time, the average number of crawled URLs per minute, successfully crawled page count, failed to crawl page count, if HTTP errors happened count of status codes, etc.
12. A button to close the software without any data loss. The same behavior is expected when the user closes the software via the X button
13. Software must be able to continue from wherever it left so you have to save the software current state and the historical state
14. Three (3) times failed to crawl URLs must be disabled for 24 hours and retried again later

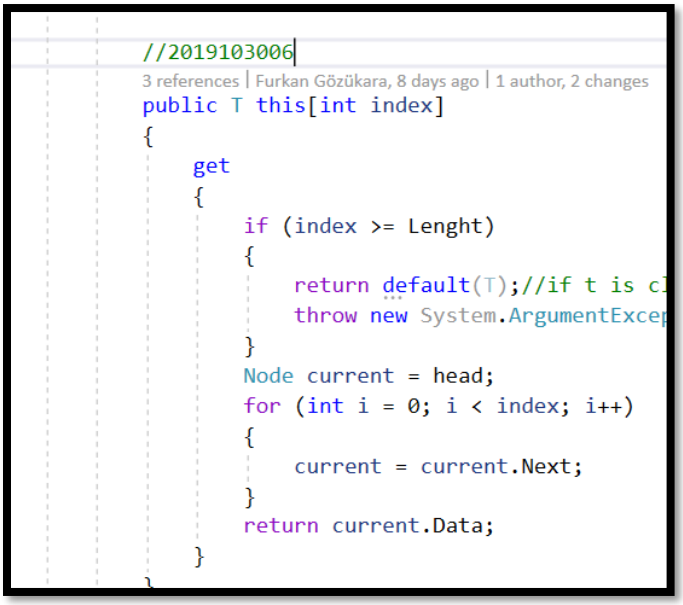
Expected Object-Oriented Features

In your software, you are supposed to write the given codes below in the respective code blocks. I will search and check the given below codes to see whether you have done the requested feature or not. Several example usage

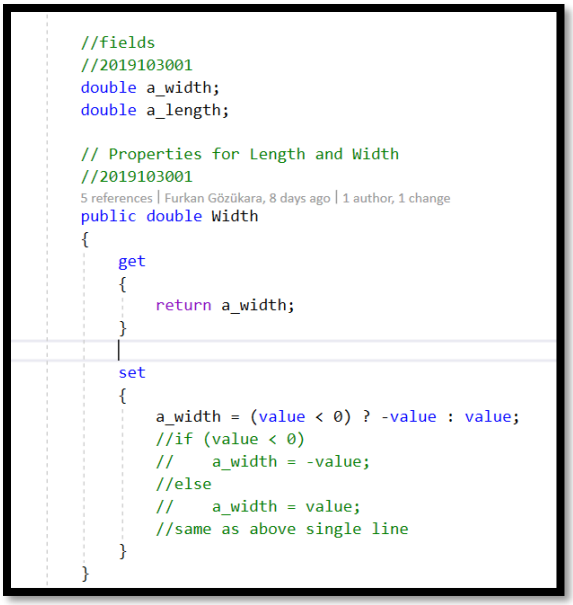


```
//public class GenericList where T : struct
//2019103015
7 references | Furkan Gözükara, 8 days ago | 1 author, 3 changes
public class GenericList<T>// where T : struct // e.g. if i force
{
    // The nested class is also generic on T.
    //2019103002
    8 references | Furkan Gözükara, 8 days ago | 1 author, 1 change
    private class Node
    {
        // T used in non-generic constructor.
        1 reference | Furkan Gözükara, 8 days ago | 1 author, 1 change
        public Node(T t)
        {
            next = null;
            data = t;
        }

        private Node next;
        3 references | Furkan Gözükara, 8 days ago | 1 author, 1 change
        public Node Next
        {
```



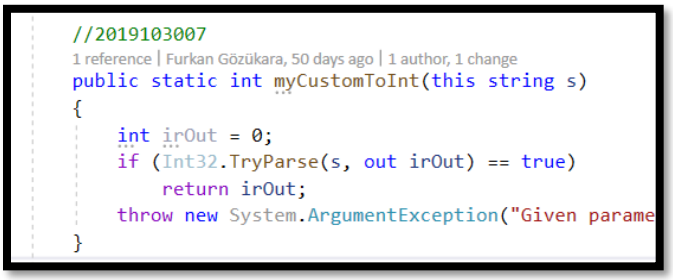
```
//2019103006
3 references | Furkan Gözükara, 8 days ago | 1 author, 2 changes
public T this[int index]
{
    get
    {
        if (index >= Lenght)
        {
            return default(T);//if t is cl
            throw new System.ArgumentExcep
        }
        Node current = head;
        for (int i = 0; i < index; i++)
        {
            current = current.Next;
        }
        return current.Data;
    }
}
```



```
//fields
//2019103001
double a_width;
double a_length;

// Properties for Length and Width
//2019103001
5 references | Furkan Gözükara, 8 days ago | 1 author, 1 change
public double Width
{
    get
    {
        return a_width;
    }

    set
    {
        a_width = (value < 0) ? -value : value;
        //if (value < 0)
        //    a_width = -value;
        //else
        //    a_width = value;
        //same as above single line
    }
}
```



```
//2019103007
1 reference | Furkan Gözükara, 50 days ago | 1 author, 1 change
public static int myCustomToInt(this string s)
{
    int irOut = 0;
    if (Int32.TryParse(s, out irOut) == true)
        return irOut;
    throw new System.ArgumentException("Given parame
}
```

- Object Oriented Programming (General Approach no specific code)
- Classes and methods must be used for everything (General Approach no specific code)
- No duplicate code, or function ever (General Approach no specific code)
- Fields and properties usage in classes (2022110801)
- Default constructor and overloaded constructor usage in classes (Polymorphism) (2022110802)
- Method and operator overloading (Polymorphism) (2022110803)
- Namespace usage example in the project. For example, you can put a class in another namespace and use it in another class (2022110804)

- Both globally and locally exception handling. The errors must be logged properly (2022110805)
- This keyword usage example (2022110806)
- Method extension example with this keyword (2022110807)
- Static and non-static class usage (2022110808)
- Public and private class, variable, and method usage (2022110809)
- Class inheritance usage (2022110810)
- Enum usage (2022110811)
- Base and this keyword usage (2022110812)
- Method overriding with virtual methods and class inheritance (Dynamic Polymorphism) (2022110813)
- Abstract class usage (2022110814)
- Generic class usage (2022110815)
- Sealed class usage (2022110816)
- Reference passing vs value passing method usage (2022110817)
- Interfaces usage (2022110818)
- Constant variables (2022110819)
- Dictionary and HashSet (2022110820)
- Structs (2022110821)
- Tuples, Nested Tuples, Tuples with Methods (2022110822)
- Default parameters in methods (2022110823)
- Anonymous Method (2022110824)
- Conditional Operator (2022110825)
- LINQ (2022110826)
- Lock usage (2022110827)
- Concurrent Collections (ConcurrentBag) (2022110828)

- Async and Await (2022110829)
- Using Statement (2022110830)
- Reflections (2022110831)
- Serialization and Deserialization e.g. JSON (2022110832)
- StringBuilder (2022110833)
- Ref keyword in methods (2022110834)
- Yield Keyword (2022110835)
- IEnumerable and IEnumerator (2022110836)
- CultureInfo (2022110837)
- Destructors (2022110838)
- IsNullOrEmpty (2022110839)
- TimeSpan, StopWatch (2022110840)
- Timers (2022110841)
- Events (2022110842)
- Optional Arguments, Named Arguments, and Generic Arguments (2022110843)
- Params Keyword (2022110844)
- Local Functions or Nested Functions (2022110845)
- Delegates and Delegate Multicast and Generic Delegates (2022110846)
- Value Tuples, Value Nested Tuples, Value Tuples with Methods (2022110847)
- KeyValuePair (2022110848)
- NameValueCollection (2022110849)
- Generic List, Generic Dictionary, Generic SortedList, Generic SortedDictionary,
  Generic Stack, Generic Queue (2022110850)

From the above-requested features, you need to try to use as many as them in your application. The more you have used, the more points you will get. Every one of them has an abundant amount of examples on the Internet. If you use other than C# language, many of them still should be available in your chosen programming language such as Java. However, their names may change.

General Things That Need to be Considered

Every project will be controlled one by one. So if you cheat or bring a ready code, Project, you will get FF

There will not a final exam. Your grade will be based on this Project. Thus, take this Project very seriously and start working on it immediately. Look the internet for the parts that you do not know

Write your code with as many as possible explanations. What does that code block? Of course, put those explanations into your code with comment blocks such as // or /* */

Create an account on https://stackoverflow.com/ and ask there, or look for answers there.

When doing the Project you can email and ask me about the parts that you cannot solve.

My email is furkan.gozukara@toros.edu.tr

Also, use discord channel for communication

Latest Project delivery date: End of finals but you can also deliver at the makeup exams

1. **Each one of you will bring your computer and present your project on your computer.**
2. **The project has to be made in English.**
3. **When explaining your Project you need to explain every piece of your written code.**
4. **What does that particular code piece or that particular class or that particular method, etc. and how did you write it?**
5. **Moreover, you have to explain and show every feature of your application. So you will run your application and do an example of every feature such as start crawling, show crawling status, restart, pause, etc.**
6. **Finally ZIP or RAR your entire Project and upload it to your Google drive. After you get your Google drive upload link send me that link as email: furkan.gozukara@toros.edu.tr**
7. **When sharing your Google drive upload link as an email, make sure that your upload's visibility is set correctly. Set it as whoever has the link can view or download it so that I can download it.**
8. **Everyone who didn't fully participate in face-to-face lectures will be sending a Word or PDF document along with their final project files. This document will contain screenshots of your all comments on the lecture videos (all of the 14 lecture videos). Take screenshots with high resolution. Your comment has to be clearly readable. Those who fail to send this document will be counted as not attended to the lectures. Therefore they will get FF.**