Homework:

Submit one Visual studio solution in your github repo - TARpe24/Homework/- name the project Homework_w7 and **for this week's** homework, check the box where it says, "Place solution and project in the same directory". LLMs like ChatGPT are not allowed but you can use google and various online documentation.

Practice:

Read: https://www.tutorialspoint.com/csharp/csharp_inheritance.htm, https://www.tutorialspoint.com/csharp/csharp_polymorphism.htm https://csharp-station.com/Tutorial/CSharp/Lesson08

Watch: https://www.youtube.com/watch?v=EiBCF7rYRtI

https://www.youtube.com/watch?v=pFCeRlr34CE

NB! If we make fields that need to be used in base class we use keyword internal

This means that our properties are accessible in derived classes but not in main method.

Practice exercise: try it and then see the solution in https://github.com/krkivimagi/HW7PracticeSolution

Please try it first yourself. This exercise uses many principles of OOP and is a good practice. **Do not submit this**.

Create an interface for Animal with methods:

- void MakeSound(no parameters)
- void Jump(no parameters)
- void PrintInfo(no parameters)
- void SetName(string name)

Create a class Animal where:

- MakeSound prints „animal is making sound"
- Jump prints: „animal (type) has jumped x times" and count the times
    - animal.Jump()- „Animal has jumped 1 times"
    - animal.Jump()- „Animal has jumped 2 times"
- PrintInfo: prints animal type and name
- SetName: sets name for animal

Create a class Dog where:

- MakeSound prints „Dog is barking"

- Name can be max 8 letters long

Create a class Cat where:

- MakeSound prints „Cat is meowing“
- Name can be max 4 letters long
- Cat can jump 3 times. If cat jumps three times then cat stops jumping and „Cat is tired, must sleep now.“ is printed.

Create a class Chiuaua where:

- MakeSound prints „Dog is barking“ and „with a really irritating voice“.

Homework task 1

Vaata: mõttega, tee näidisülesanded ise kaasa ja kirjuta paberile märkmeid:

1) https://learn.microsoft.com/en-us/shows/csharp-fundamentals-for-absolute-beginners/understanding-classes

2) https://learn.microsoft.com/en-us/shows/csharp-fundamentals-for-absolute-beginners/understanding-scope-and-accessibility-modifiers

3) https://learn.microsoft.com/en-us/shows/csharp-fundamentals-for-absolute-beginners/more-about-classes-and-methods

Vajadusel vaata juurde ka: https://www.youtube.com/watch?v=t2SPg6IuT3k

Otsi ise juurde lugemismaterjale/videosi kasutades märksõnadeks nt: "C# classes tutorial"

Tee: Vasta allolevad 2 küsimust; lisa vastused .pdf formaadis repose kausta.

Write answers as .pdf file to following questions. Add this file to Homework_w7 folder in your repo.

1) Kirjuta 2-3 lausega (oma sõnadega!) mida tähendavad järgnevad mõisted ning kuidas/kus neid kasutatakse:
a) klass (programmeerimises) b) klassi konstruktor c) objekt d) meetodite ülelaadimine

Write with 2-3 sentences (with your own words!) what is the meaning of following terms and how to use them:

a) class (in programming) b) class constructor c) object d) method overloading

2) Pane kirja millised 3 omadust (nimisõna, kirjeldavad objekti) ja millised 3 tegevust (tegevus, mida objekt teeb) võiksid sinu arvates olla järgmistel objektidel:

a) Robottolmuimeja b) Käsiaag c) Korstnapühkija d) Teetass

Write down which 3 properties (nouns that describe the object) and which 3 behaviors (activities done by the object) could describe the following objects:

a) Robot vacuum cleaner b) wood saw c) Chimney sweeper d) Tea cup

Homework task 2

Think this through! If you use good structure for classes then this exercise is very simple. If you use bad class design then this task is very long and complicated. Write structure on paper!

You work in a company that creates irons (*triikraud*). There are 3 types of ironing machines:

- Regular
- Premium
- Linen

All ironing machines have to have these methods (create an interface and implement it):

- Descale
- DoIroning
- UseSteam
- TurnOn
- TurnOff

Properties for the machine:

- There are differnt programs (temperature ranges) for different fabrics:

  200°C -230 °C : Linen program

  150°C -199 °C : Cotton program

  120°C -149 °C : Silk program

  90°C -119 °C : Synthetics program

Linen machine can do all of the programs; premium and regular machine cannot do linen program (their possible highest temperature is 199).

- For all machines there are two DoIroning methods:
  - Method which takes temperature as parameter and prints out the machine type and the program that is used (example: „DoIroning(170)“ - > „Premium machine is ironing with Cotton program“.) ○ Method which takes the program name as parameter and prints out the machine type and a random range from within the program temperature range (example:
    „DoIroning(„Cotton“)“ - > „Regular machine is ironing with 154 degrees.“)

- All machines need to be cleaned after they have been used for 3 times (after 3 times executing the „DoIroning“ method).
    - Regular and linen machines have to be cleaned manually (Descale method has to be called in main class). There should be a notification that „Machine has been used 3 times and needs cleaning“. You cannot use DoIroning method before the machine is cleaned.
    - Premium machine cleans itself (You do not need to call out Descale method from main class, it is called from within the DoIron method but ONLY when neccessary, ie its been used 3 times for ironing)
- Descale method sets the usage counter to 0 and prints „Machine is cleaned“.
- There is an option to use steam while ironing. Using steam only works for one ironing cycle. Example:

```
        myIron.UseSteam();
        myIron.DoIroning(); //irons with steam, then turns off the steam
myIron.DoIroning(); //irons without the steam
```

- Steam can be used only if the program temperature is at least 120 degrees. With linen program the steam should be always turned on automatically. If steam is used, the info should be printed out in DoIroning method (additional text „Ironing with steam“). If steam is not used, no info should be printed.
- If UseSteam is called out 2 times in a row the second time should print „Steam is already on“.
- Premium machine has a indicator light for letting the user know when you need to add water to the machine. If steam is used 2 times, the light goes on. Create a property that indicates if the light is on or off and if it is turned on then notify the user. Method for turning off the light is not neccessary.
- DoIron methods can only be called out with correct parameter values. If fase values are used, there should be a notification and ironing shuld not happen. Example:
    - DoIron(300) -> „Invalid temperature range for ironing“
    - DoIron(„underwear“) -> „We do not have a program for ironing underwear“
- In main method create different objects and call out the methods and try different scenarios.

Tips:

- **DO NOT** create duplicated code. Base code comes from the base class. In derived classes there should be ONLY custom logic.
- Think of which class should be the base class. Base class carries the methods and properties that are common to all classes.
- Think of all the properties that are needed.
- Start by creating base class. Then create derived classes.

- To understand when the iron needs cleaning we need to keep track of all the times when DoIroning methods were called out. Best option is to use a counter inside the iron class (do not do this in main method!).
- For all machines there should be also info about machine types (regular, premium, linen). Maybe it is good to set these values in constructors?
- Constructors are not specified in the task, you can use them according to your needs.
- Keyword for making properties in base class available to derived classes is „protected“
- Keywords for overwriting the methods are „virtual“ and „override“ (only virtual methods in base class can be overwritten in derived class). Make only those methods virtual that have to be overwritten!
- Think which methods should be called out from within other methods and which ones from the main class.
- Check https://msdn.microsoft.com/en-us/library/2dx6wyd4(v=vs.110).aspx method for generating the temp value. Are upper and lower values included in the generated random number?