BDSA - Assignment 01

Alaa Abdul-Al (alia)
Shuja Hussain (shhu)
Mohammad Hasham (mhas)
Gustav Sølvsteen Olesen (guol)

September 24th, 2021

1 C#

The code can be found on the following Github repo:
https://github.com/Alaa0129/AS02.git

1.1 Constructs

A record is also a reference type, just like classes. To put it simple, the difference of the latter is, first of all that a record is just a class on drugs. Elaborating on this, it gives some extra pre-coded features, which a class cannot provide. First of all, a record is only readonly, which means you, inside the record itself, cannot change the initial values given to it, hence it is immutable. When creating a record, and giving it some parameters, it actually does several things behind the scenes. First of all it initialises the parameters, as well as creating get; and init; methods and a constructor. All this just in one line. There are also many features made easier in records compared to classes. To name one of them, the ToString() method for records is built-in, which means you can print out information about a record, by just saying:

```
var s2 = new ImmutableStudent(1,"John","Doe",
Status.Active,new DateTime(2021, 09, 22, 00, 00, 00),
new DateTime(2024, 06, 24, 00, 00, 00),new
DateTime(2024, 06, 22, 00, 00, 00));

Console.WriteLine(s2);
```

Records are not always the best option to choose. Normally a class is the most natural one. However, an example of when to use records, is when dealing with external data that does not change. An example hereof is the weather.

Struct is, compared to the previously mentioned constructs, a value type. This means a struct has the same type as an int for example. As a result of this, you do not pass references to objects, but rather copies. Meaning if you want to modify a value, it would not be possible. A structure type almost has the same capabilities as a class. However with some exceptions, where one of these is that you cannot inherit from other classes or structs.

2 Software Engineering

2.1 Exercise 1

Use cases are used during requirement elicitation and analysis to depict the functionality of the system. They are used when describing the behaviour of the system from the users point of view.
A scenario is an instance of a use case. Here, a scenario describes a concrete set of actions. Scenarios are used when you want to specify an action in depth. It focuses more on the understandability.

2.2 Exercise 2

2.3 Exercise 3

2.4 Exercise 4

2.5 Exercise 5

2.6 Exercise 6