**Data Structures Advanced with C# - Regular Exam**

# Kubernetes – 100 pts

Kubernetes, also known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications.

You are given a skeleton with a class **Controller** that implements the **IController interface.**

This **Controller** works with **Pod** entities. All **Pod** entities are identified by a **unique Id**.

The **Pod** entity contains the following properties:

* **Id** – string
* **ServiceName** – string
* **Port** – integer
* **Namespace** – string

Implement the following functionalities to make **Controller** fully operative:

* **void Deploy(Pod pod)**: This method is used to add a pod for deployment. It takes a Pod object as an argument and adds it to the system for execution.
* **bool Contains(string podId)**: This method checks whether a pod with a specific ID exists in the system. It returns true if a pod with the given ID is present, and false otherwise.
* **int Size()**: This method returns the total count of all the pods that are currently deployed.
* **Pod Get(string podId)**: This method retrieves a specific pod from the system based on its ID. If a pod with the provided ID exists, it returns the corresponding Pod object. If no such pod exists, it **throws an ArgumentException**.
* **void Uninstall(string podId)**: This method is used to remove a pod from the system completely. It removes any reference to the pod, effectively deleting it from the system. If there is no pod with the provided ID, it **throws an ArgumentException**.
* **void Upgrade(Pod pod)**: This method is used to upgrade an existing pod. It takes a Pod object representing the updated configuration and applies it to the corresponding pod in the system. **If a pod with such ID does not exist, then it's deployed**.
* **IEnumerable<Pod> GetPodsInNamespace(string namespace)**: This method returns an enumerable **collection of all the pods that belong to a specific namespace**.
* **IEnumerable<Pod> GetPodsBetweenPort(int lowerBound, int upperBound)**: This method returns an enumerable **collection of pods that fall within a specified port range (inclusive)**.
* **IEnumerable<Pod> GetPodsOrderedByPortThenByName()**: This method returns an enumerable collection of all pods **ordered first by their port in descending order** and **then by their names in ascending order**.

**NOTE: If all sorting criteria fails, you should order by order of input. This is for all methods with ordered output.**

* 1. **Kubernetes – Performance – 50 pts**

For this task you will only be required to submit the **code from the previous problem**. If you are having a problem with this task you should **perform detailed algorithmic complexity analysis** and try to **figure** **out** **weak** spots inside your implementation.

For this problem it is important that other operations are **implemented** **correctly** according to the specific problems: **add**, **size**, **remove**, **get** etc… Also, make sure you are using the correct data structures. ☺

You can submit code to this problem **without full coverage** from the previous problem, **not all test cases** will be considered, only the **general** **behaviour** will be important, **edge** **cases** will mostly be ignored such as throwing exceptions etc…