Namn: Mojtaba Sarvari

Kod A

Koden är ett exempel på Factory pattern. Koden använder sig av implementering av interface vid skapande av objekt och returnerar en extension av objektet vilket i detta fall returneras en hamburgare

```
namespace RestaurantManagement.Restaurants
{

    public class FastFoodRestaurant : Restaurant
    {
        public override IMeal PrepareMainCourse()
        {
            return new Hamburger();
        }
    }
}
```

Kod B

Koden är ett exempel på Singleton. Koden har ett gemensamt interface som används av klasserna och varje instance på klasser går att att ha endast en gång. Man kan se i klasserna att en instance instatieras och returneras sedan

```
namespace Greeter.Types
{

    public class LocklessGreeter : BaseGreeter
    {
        private static readonly LocklessGreeter InstanceValue = new LocklessGreeter();

        static LocklessGreeter()
        {
        }

        private LocklessGreeter()
        {
        }

        public static LocklessGreeter Instance
        {
            get
            {
                return InstanceValue;
            }
        }
    }
}
```

Namn: Mojtaba Sarvari

Kod C

Koden är ett exempel på Observer pattern. Klassen sorting person känner till sina observer och objekt är sparade i listor som sedan hämtas och uppdateras enligt bild 2.

```
namespace Library.SortingExample
{
    public class SortablePersons
    {
        private readonly List<Person> _persons;

        public SortablePersons(ISort sort)
        {
            Sort = sort;
            _persons = new List<Person>();
        }

        public ISort Sort { get; set; }

        public void Add(Person person)
        {
            _persons.Add(person);
        }

        public void Clear()
        {
            _persons.Clear();
        }

        public List<Person> Sort()
```

```
{
    public static class SortingExecutor
    {
        public static void Execute()
        {
            ConsoleExtension.WriteSeparator("Sorting example");

            ISort sort = new SortByFirstName();
            var sortablePersons = new SortablePersons(sort);

            sortablePersons.Add(new Person("Dennis", "Ritchie", 1941));
            sortablePersons.Add(new Person("Linus", "Torvalds", 1969));
            sortablePersons.Add(new Person("Tim", "Berners-Lee", 1955));
            sortablePersons.Add(new Person("Larry", "Page", 1973));
            sortablePersons.Add(new Person("Anders", "Hejlsberg", 1960));
            sortablePersons.Add(new Person("Bjarne", "Stroustrup", 1950));

            Console.WriteLine("Sorting persons by first name.");
            foreach (Person person in sortablePersons.Sort())
            {
                Console.WriteLine(person.ToString());
            }

            sortablePersons.Sort = new SortByYearOfBirth();

            Console.WriteLine("\nSorting persons by year of birth.");
            foreach (Person person in sortablePersons.Sort())
            {
                Console.WriteLine(person.ToString());
```

Namn: Mojtaba Sarvari

Kod D

Koden är ett exempel på Adapter pattern. Koden känns igen av en konstruktor som tar en instans av en annan gränssnittstyp. När adaptern tar emot ett anrop till någon av kod D:s metoder, översätter den parametrar till json format och dirigerar sedan anropet för det lindade objektet, se exempel nedan:

```
namespace Library.MovieBroadcasterExample
{

    public class Broadcast : IBroadcaster
    {
        private readonly MovieRegistry _movieRegistry;
        private readonly ThirdPartyBroadcaster _thirdPartyBroadcaster;

        public Broadcast(MovieRegistry movieRegistry)
        {
            _movieRegistry = movieRegistry;
            _thirdPartyBroadcaster = new ThirdPartyBroadcaster();
        }

        public void BroadcastToExternalPartners()
        {
            string jsonMovies = ConvertRegistryMoviesToJson();

            _thirdPartyBroadcaster.Broadcast(jsonMovies);
        }

        private string ConvertRegistryMoviesToJson()
        {
            XDocument xmlMovies = _movieRegistry.GetAll();

            Console.WriteLine("Movies from internal registry...");
            Console.WriteLine(xmlMovies);

            IEnumerable<Movie> modelMovies = xmlMovies
```