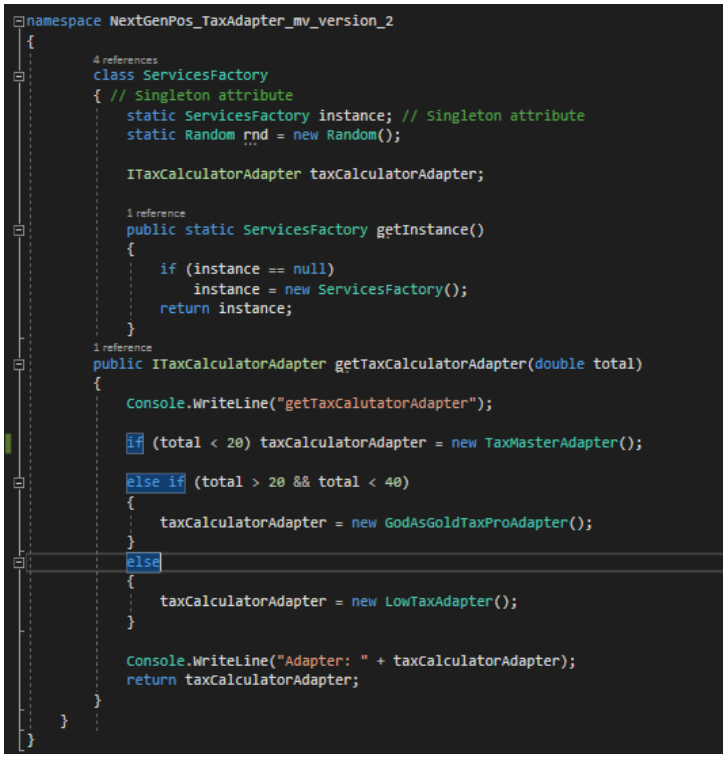
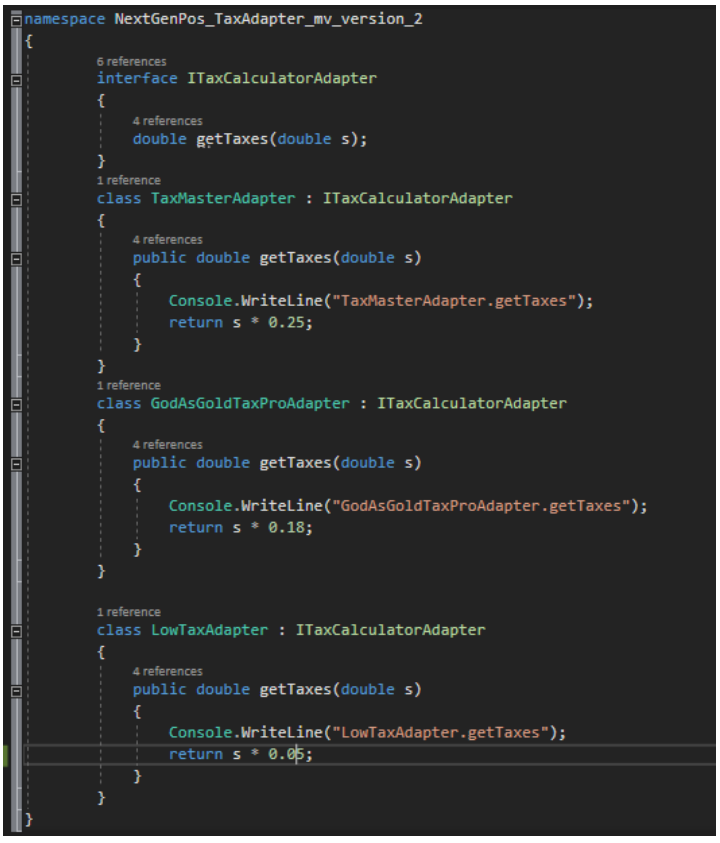
NextGenPos_TaxAdapter



```
namespace NextGenPos_TaxAdapter_mv_version_2
{
    4 references
    class ServicesFactory
    { // Singleton attribute
        static ServicesFactory instance; // Singleton attribute
        static Random rnd = new Random();

        ITaxCalculatorAdapter taxCalculatorAdapter;

        1 reference
        public static ServicesFactory getInstance()
        {
            if (instance == null)
                instance = new ServicesFactory();
            return instance;
        }
        1 reference
        public ITaxCalculatorAdapter getTaxCalculatorAdapter(double total)
        {
            Console.WriteLine("getTaxCalutatorAdapter");

            if (total < 20) taxCalculatorAdapter = new TaxMasterAdapter();

            else if (total > 20 && total < 40)
            {
                taxCalculatorAdapter = new GodAsGoldTaxProAdapter();
            }
            else
            {
                taxCalculatorAdapter = new LowTaxAdapter();
            }

            Console.WriteLine("Adapter: " + taxCalculatorAdapter);
            return taxCalculatorAdapter;
        }
    }
}
```

Her bliver mons'en udregnet udfra hvad det totale beløb der bliver modtaget fra sales.



```
namespace NextGenPos_TaxAdapter_mv_version_2
{
    6 references
    interface ITaxCalculatorAdapter
    {
        4 references
        double getTaxes(double s);
    }
    1 reference
    class TaxMasterAdapter : ITaxCalculatorAdapter
    {
        4 references
        public double getTaxes(double s)
        {
            Console.WriteLine("TaxMasterAdapter.getTaxes");
            return s * 0.25;
        }
    }
    1 reference
    class GodAsGoldTaxProAdapter : ITaxCalculatorAdapter
    {
        4 references
        public double getTaxes(double s)
        {
            Console.WriteLine("GodAsGoldTaxProAdapter.getTaxes");
            return s * 0.18;
        }
    }

    1 reference
    class LowTaxAdapter : ITaxCalculatorAdapter
    {
        4 references
        public double getTaxes(double s)
        {
            Console.WriteLine("LowTaxAdapter.getTaxes");
            return s * 0.05;
        }
    }
}
```

Her er der tilføjet en ny class "LowTaxAdapter"

```
3 references
public double getTotal()
{
    double dblTotal = 0;
    double dblSubTotal = 0;
    foreach (SalesLineItem lineItem in lineItems)
    {
        dblSubTotal = lineItem.getSubTotal();
        dblTotal = dblTotal + dblSubTotal;
    }

    ITaxCalculatorAdapter taxAdapter;
    taxAdapter = ServicesFactory.getInstance().getTaxCalculatorAdapter(dblTotal);
    tax = taxAdapter.getTaxes(dblTotal);
    taxes.Add(new TaxLineItem("Moms", 1, tax));

    return dblTotal;
}
```

getTotal() har jeg tilføjet noget mere kode der efter at have udregnet det fulde beløb, udregner mons'en

Monopoly

```
1 reference
public void TakeTurn(int roundCnt)
{
    if (jailTime == 1)
    {
        Console.WriteLine("Du er i Jail vent en omgang");
        jailTime = 0;
    }
    else
    {
        Cup cup = new Cup();
        cup.RollDice();
        MovePieceOnBoard(pce, cup.FaceValueTot);
        Console.Write(pce.SquareLocation.SquareName +
            " Slag: " + cup.FaceValueTot.ToString());
    }
}

1 reference
private double UpdateMoney(double sum)
{
    return playerSum += sum;
}

1 reference
private int jailTimeCounter()
{
    return jailTime = 1;
}
```

TakeTurn er opdateret med en if statement der tjekker om player landende på Jail feltet sidste runde.

UdateMoney methoden giver player 200kr hver gang de rammer start, implementering til når de passere start kommer på et senere tidspunkt.

JailTimeCounter methoden giver spilleren JailTime counteren + 1

```
public Board(int numSquares)
{
    Squares = new List<Square>();
    for (int i = 0; i < numSquares; i++)
    {
        Square sq = new RegularSquare();
        sq.SquareName= MonopolyConstants.SquareNames[i];
        sq.SquareNo = i;
        Squares.Add(sq);
    }
    squares[0] = new StartSquare();
    squares[0].SquareName = MonopolyConstants.SquareNames[0];
    squares[0].SquareNo = 0;

    squares[10] = new JailSquare();
    squares[10].SquareName = MonopolyConstants.SquareNames[10];
    squares[10].SquareNo = 10;
}
```

Board klassen er opdateret med beskrivelse af hvor Start og Jail square'en