Øvelse 1.

a.

Lav i et nyt program program (eget valg, console, winforms etc.) en abstrakt klasse Shape med properties X og Y (double) og passende constructors. Lav en constructor, der initialiserer X og Y, og en parameterløs contructor. Den parameterløse skal initialisere til 1,1 og gøre dette ved at kalde den anden constructor.

b.

Lav underklasser til Shape:

- Circle, med yderligere property Radius (double), tilret constructor, så den også tager radius med (og kalder Shape's constructor)
- Rectangle, med yderligere properties Length og Width (double) og tilret ctor til at inkludere disse.
- I begge klasser skal ToString() overrides, så den også udskriver de ekstra properties, som ikke er med i Shape-klassen

Test i en main-metode. Lav en liste af Shape-objekter, tilføj nogle til denne og gennemløb den med en løkke og udskriv oplysninger om hvert objekt.

c.

Lav en abstrakt metode public double Area() i Shape.

Lav implementationer af denne i underklasserne.

Udskriv Area i løkken fra før.

d.

Vi ønsker en Move-metode, der flytter en shape (altså en Circle eller Rectangle). Move-metoden skal tage en XChange og en YChange værdi, og disse skal bruges som ændringerne til X og Y properties, dvs. de skal lægges til. Negative Change-værdier vil så give mindre X og Y værdier. Hvor skal denne metode ligge?

e. Nu vil vi lægge nogle regler ned på Move-metoden.

- for Rectangle må X og Y ikke blive negative. Dvs. vi forestiller os, at X og Y er det øverste venstre hjørne af rektanglet, der altid ligger vandret/lodret.

- for Circle ligger X og Y i centrum af cirklen. Cirklen må ikke komme ud på negative X og Y værdier, heller ikke kanten af cirklen. Så man må ikke komme længere ud end at hele cirklen ligger rent på positive værdier, dvs. man skal tage højde for radius.

Hvor og hvordan skal disse regler implementeres? Det kan jo ikke være i Shape-klassen.

Øvelse 2. struct øvelse (hvis det går hurtigere med øvelse 1 end forventet)

Denne opgave laves i et nyt program (eget valg, console, winforms etc.)

Implementer en struct kaldet *Time*, som repræsenterer et tidspunkt 00:00:00-23:59:59. Internt gemmes tidspunktet som et antal sekunder i et privat felt. F.eks. svarer 01:23:20 til 5000 sekunder i det private felt.

Så du kan erklære den sådan : `struct Time {…}`

I de foregående opgaver blev det foreslået at oprette en fil pr. klasse. Her kan I nøjes med at skrive det hele i en cs-fil. Dvs. du kan have din struct defineret i samme namespace som din main metode.

Erklær en int, der indeholder antal sekunder siden midnat i din struct, kald den f.eks. secondsSinceMidnight.

Erklær tre properties, som sætter og returnerer henholdsvis timer, minutter og sekunder (i det følgende antages det, at de hedder Hour, Min og Sek). De skal så også opdatere antal sekunder siden midnat – her et eks. på hvordan Hour kan se ud – prøv at forstå hvad der sker?:

```
public int Hour
            {
                set
                {
                    int hour = value;
                    int temp = _secondsSinceMidnight % 3600;
                    _secondsSinceMidnight = (temp + hour * 3600) % _maxSeconds;

                }

                get { return _secondsSinceMidnight / 3600; }


            }
```

Skriv en constructor, der kan tage tidspunktet som en string på formatet ”hh:mm:ss” (f.eks. ”01:23:20”). Altså altid 3 gange to cifre adskilt af kolon. Du skal så i gang med at bruge substring og parse for at få fat i timer, minutter og sekunder:

Brug SubString() i string klassen til at trække de enkelte værdier ud. Omform til tal med Int32.Parse().

Skriv en anden constructor, der tager 3 int som parametre, nemlig timer, minutter og sekunder og så sætter værdierne

Lav desuden en ToString(), der returnerer tidspunktet på førnævnte format. *Bemærk at signaturen skal indeholde ”override”, dvs. den skal være*

```
public override string ToString()
```

Test din struct, og prøv f.eks. følgende kode i main:

```
Time t1, t2;
t1=new Time("08:30:00");
```

```
t2=t1;
t2.Hour=t2.Hour+2;

Console.Write(t1.ToString());

Console.Write(t2.ToString());
```

Test også fejl i input ved oprettelse og ændring.