Opgave 1 – Værksted

Denne opgave omhandler et antal samarbejdende klasser til håndtering af informationer til et bilværksted.

Du skal implementere fire klasser:

`Vaerksted` – overordnet klasse, som bl.a. indeholder containere af de øvrige klasser.

`Bil` – med informationer om kundernes biler.

`Mekaniker` – med informationer om værkstedets mekanikere.

`ArbejdsSeddel` – med data om arbejde, som mekanikerne har udført på bilerne.

Klassen `Vaerksted`

Nedenfor er vist et uddrag af et forslag til headerfil:

```
class Vaerksted
{
public:
            Vaerksted();
            Vaerksted(string adresse, string ejer);
            string getAdresse();
            string getEjer();
            void addBil(Bil &enB);
            void addMekaniker(Mekaniker enM);
            void addArbejdsSeddel(ArbejdsSeddel &enA);
            ~Vaerksted();
private:
            string adressen;
            string ejeren;
            // Her tilføjes containerne og evt. attributter til at holde styr
            // på antallet af elementer i containerne
            // biler - fx en vector eller et array med objekter af klassen Bil
            // mekanikere - fx en vector eller et array med objekter af klassen Mekaniker
            // arbejdsSedler - fx en vector eller et array med objekter af klassen
            //                 ArbejdsSeddel
};
```

Klassen `Bil`

indeholder attributterne `regNr` og `ejer`, begge af typen `string`.

Klassen `Mekaniker`

indeholder attributten `navn` af typen `string`.

Klassen `ArbejdsSeddel`

indeholder referencer til et objekt af klassen `Bil` og til et objekt af klassen `Mekaniker`. Derudover attributterne `datoen` og `timerne`, som angiver datoen for arbejdets udførelse og antal anvendte timer. Begge er af typen `int`.

Når du har implementeret klasserne, skal nedenstående testdriver kunne køres uden fejl.

```
int main()
{
            Vaerksted vS("Nymarken 104, 5330 Munkebo", "Lars Peter Hansen");

            Bil b1("MA39604", "Knud Pedersen");
            Bil b2("MH40136", "Lis Fugl");
            Bil b3("MA45647", "Herluf Jensen");
            vS.addBil(b1);
            vS.addBil(b2);
            vS.addBil(b3);

            Mekaniker m1("Poul");
            Mekaniker m2("Per");
            vS.addMekaniker(m1);
            vS.addMekaniker(m2);

            ArbejdsSeddel a1(b1, m2, 7, 20200503);
            ArbejdsSeddel a2(b3, m1, 4, 20200512);
            ArbejdsSeddel a3(b3, m2, 3, 20200514);
            ArbejdsSeddel a4(b1, m1, 5, 20200516);
            ArbejdsSeddel a5(b3, m2, 2, 20200518);
            vS.addArbejdsSeddel(a1); // Det antages, at arbejdssedler
            vS.addArbejdsSeddel(a2); // registreres i kronologisk orden
            vS.addArbejdsSeddel(a3);
            vS.addArbejdsSeddel(a4);
            vS.addArbejdsSeddel(a5);

}
```

Du skal nu implementere følgende tre metoder i Vaerksted-klassen

1

`void Vaerksted::ingenRegning()`

Metoden skriver (på `cout`) navnene på de bilejere, som ingen regning skal have på baggrund af de udfyldte arbejdssedler. I dette tilfælde *Lis Fugl.*

2

`void Vaerksted::mekanikerTimer()`

Metoden udskriver en liste på `cout` med navnene på mekanikerne og det samlede antal timer, de hver især har arbejdet ifølge arbejdssedlerne. En linje per mekaniker.

3

`string Vaerksted::senesteArbejde()`

Metoden returnerer en `string` indeholdende datoen for det senest udførte arbejde på formen DD/MM/YYYY. I dette tilfælde ”18/05/2020”.

Opgave 2 – Find ord

Skriv en metode med følgende signatur:

```
bool ordITekst(string tekst, string ord)
```

Metoden returnerer `true`, hvis parameteren `ord` er indeholdt i parameteren `tekst`. Hvis ikke, returneres `false`.

Opgave 3 – Find største sum

Skriv en metode med følgende signatur:

```
int largestSumOfTwo(int ar1[], int ar2[], int size)
```

Metoden skal returnere summen af de to største tal i begge arrays. Kaldt med de to nedenstående arrays og med size = 5, returneres 110.

```
int ar1[] = { 17,33,44,11,9 };
int ar2[] = { 22,66,1,35,22 };
```