

1. Класове и обекти.

Класът представлява описание на тип, включващ едновременно данни и функции, които ги обработват. Данните се наричат **член-променливи**, а функциите – **член-функции**.

Дефиницията на един клас включва **декларация на класа** и **дефиниции на член-функциите**.

Декларацията на класа има следния синтаксис:

```
class име-на-клас {
    // Декларации на член-променливи
    // Декларации на член-функции
};
```

Дефиницията на член-функция има следния синтаксис:

```
тип-връщан-резултат име-на-клас::име-на-функция(списък с параметри)
{
    // Тяло на функцията
}
```

За да се създаде **обект**, се използва името на класа като спецификатор за типа. Следният ред декларира обект **obl** от тип **triangle**:

```
triangle obl;
```

Достъпът до **public** компонентите на даден обект се осъществява по 2 начина – чрез името на обекта и името на компонентата разделени с точка или чрез указател към обекта и името на компонентата:

```
obl.face();
obl.show("triangle1");
```

```
triangle ob2;
triangle *p;
p = &ob2;
p->face();
p->show("triangle2");
```

Конструкторът на един клас се извиква всеки път, когато се създава обект от този клас. И поради това всякаква инициализация, която трябва да се извърши за даден обект, може да се изпълни автоматично от функцията конструктор.

Конструкторът има същото име като името на класа, към който принадлежи, и не притежава тип на връщан резултат. Общият вид на конструктора е:

```
име-на-клас::име-на-клас(списък формални аргументи)
{
    //Тяло на конструктор
}
```

Деструкторът се извиква автоматично при разрушаването на обект от класа. Деструктора може да извежда съобщение за разрушаване на обекта или да освобождава заделената за обекта динамична памет.

Деструкторът има същото име като класа, но предшествано от символа `~`. Деструкторът не може да връща резултат и не може да има аргументи. Общият вид на деструктора е:

```
~име-на-клас::име-на-клас()
{
    //Тяло на деструктор
}
```

Задача 1. Да се дефинира клас за фигура триъгълник. Член-променливите на класа са дължините на трите страни на триъгълника: a , b , c . Въвеждането на трите страни на триъгълника да става чрез конструктор без параметри. Член-функциите включват функции за: изчисляване на лицето на триъгълника `face()`; извеждане на информация за страните на

триъгълника `show()`. Да се състави главна функция `main()`, която тества класа `triangle`, като създава обект от този клас, указател към обект от същия клас и извиква член-функциите на класа, веднъж посредством името на обекта и втори път посредством указателя към обекта.

```
#include <iostream.h>
#include <math.h>
class triangle {          //Декларация на клас triangle
private:
    double a, b, c;      //Член-данни на класа
public:
    triangle();          //Декларация на конструктор
    double face();      //Декларация на член-функция
    void show(char *);  //Декларация на член-функция
    ~triangle()         //Дефиниция на деструктор
    {
        cout << "\nDestructing object triangle!\n";
    }
};
triangle::triangle()    //Дефиниция на конструктора
{do
    { cout << "\n\nEnter three values for the sides of triangle:\n";
      cin >> a >> b >> c;
    }
    while(!((a>0) && (b>0) && (c>0) && ((a+b)>c) && ((a+c)>b) && ((b+c)>a)));
}
double triangle::face() //Дефиниция на член-функция
{
    double p, s;
    p = (a+b+c)/2;
    s = sqrt(p*(p-a)*(p-b)*(p-c));
    return s;
}
void triangle::show(char *name) //Дефиниция на член-функция
{
    cout << "Sides of the triangle " << name << ":\n";
    cout << "a=" << a << ",b=" << b << ",c=" << c;
}
void main()
{
    triangle ob1;        //Създаване на обект ob1 от клас triangle
    double s;
    s = ob1.face();      //Извиква се член-функция face() за обект ob1
    ob1.show("triangle1"); //Извиква се член-функция show() за обект ob1
    cout << "\nThe face of triangle1 is s=" << s;

    triangle ob2;
    triangle *p;        //Декларира се указател към обект от клас triangle
    p = &ob2;           //Указателят се установява да сочи към ob2
    s = p->face();       //Извиква се член-функция face() за обект ob2
    p->show("triangle2"); //Извиква се член-функция show() за обект ob2
    cout << "\nThe face of triangle2 is s=" << s;
}
```

Задачи за самостоятелна работа:

Задача 1:

Създайте клас, който описва часово време (Time) със следните характеристики:

- час, със стойности от 0 - 23 (24-часови формат)

- минути, със стойности от 0 - 59
- секунди, със стойности от 0 - 59

Класът съдържа следните член - функции:

- За задаване на стойности за час, минути и секунди, като се прави проверка за съответния диапазон от позволени стойности.
- За отпечатване на времето в два формата:
чч:мм:сс - например: 13:24:07
чч:мм:сс AM/PM - например: 13:24:07 PM

Демонстрирайте функционалността на класа чрез създаване на обект в `main()`.

Задача 2:

Създайте клас, който описва обекта работник със следните характеристики: социален номер, име, години трудов стаж, текуща длъжност и списък от взимани заплати (тип данни - масив).

Забележка: Нивото на достъп до член - променливите да е `private`.

Класът има два конструктора:

Инициализира социалният номер и годините трудов стаж с нулева стойност.

Инициализиране на текуща длъжност да се въведе от клавиатурата.

Класът съдържа следните член - функции:

Задаване и получаване на стойности за член-променливите.

За пресмятане на средно аритметичната стойност от заплатите.

За намиране на минимална заплата.

Демонстрирайте функционалността на класа чрез създаване на обект в `main()`.

Задача 3:

Създайте клас `Line`, който чертае линия на екрана, чрез изобразяване на линията с помощта на `*`. Съхранете дължината на линията в `private` променлива `Len` от тип `Integer`. Нека конструкторът на `Line` приема един параметър: дължината на линията. Нека конструкторът да изчертава линията и да съхранява нейната дължина. Създайте деструктор, който да изтрива линията.