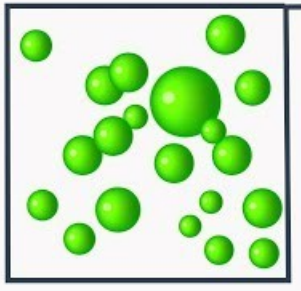
Семинар 3 по ООП

Модификатори за достъп, класове, конструктори

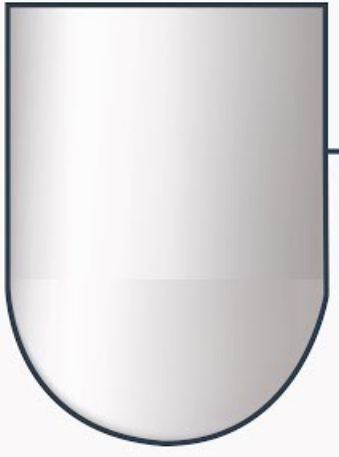
Енкапсулация

- Един от четирите основни принципа на ООП.
- Начин за скриване на данните на класа с цел потребителя да няма директен достъп до състоянието на променливите му.
- Реаллизира се чрез модификаторите за достъп

ENCAPSULATION



Variable or State



Method or Behavior

Модификатори за достъп

- Те са 3: public, private и protected
- В този курс ще използваме основно public и private
- Public – прави съответния елемент публичен. Той може да се достъпи от всякъде. От други класове, от наследници и от самия клас
- Protected – втори по рестриктивност. Елементът може да се достъпи от самия него или от наследници
- Private – най-рестриктивен. Дава достъп до елемента само в рамките на текущия клас

Класове

- Разлика със структури:
- Структурите се използват основно за групиране на данни, докато класовете – за по-сложна абстракция и евентуално наследяване
- Елементите на класовете по подразбиране са private, а на структурите - public

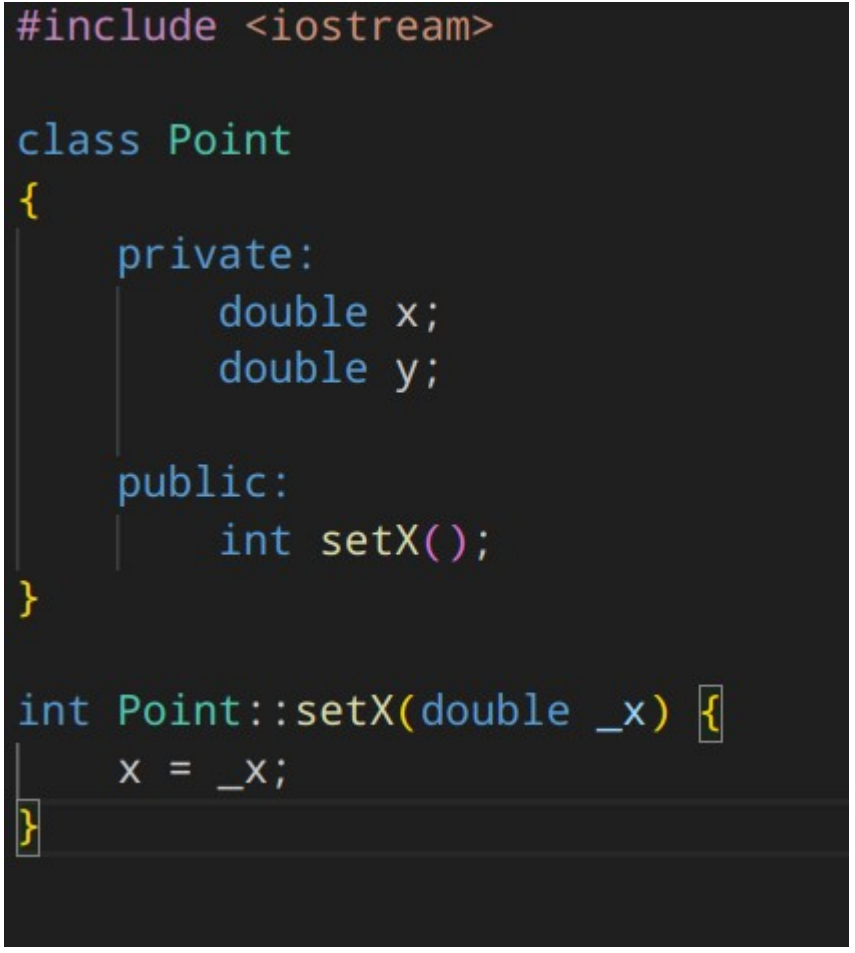
Пример

```
#include <iostream>

class Point
{
    public:
        double x;
        double y;
        int setX();
}

int Point::setX(double _x) {
    x = _x;
}
```

Този пример води до лоши резултати на контролни :)

- Нарушен е основен принцип на ООП-то, а именно енкапсулацията. Не трябва да имаме директен достъп до състоянието на променливите.

Как всъщност трябва да е



```
#include <iostream>

class Point
{
    private:
        double x;
        double y;

    public:
        int setX();
}

int Point::setX(double _x) {
    x = _x;
}
```

Конструктори

- Функции, които се викат при създаване на обект, който е инстанция на даден клас.

И за какво са ни?

- Чрез конструкторите, ние инициализираме стойности на полетата, при създаване на нова инстанция на клас.
- Има два вида конструктори: Конструктор с параметри и конструктор по подразбиране

Пример

```cpp
#include <iostream>

class Point
{
    private:
        double x;
        double y;

    public:
        Point();
        Point(double _x, double _y);
};
```

```cpp
Point::Point()
{
    x = 0;
    y = 0;
}

Point::Point(double _x, double _y)
{
    x = _x;
    y = _y;
}
```

Задача 1

- Напишете клас триъгълник с полета 3 страни, конструктор по подразбиране и с 3 параметъра за всяка страна. Проверете дали параметрите са валидни, ако не са сложете стойности по подразбиране. Реализирайте методи за обиколка и лице.

Задача 2

- Тези дни беше Трети март. Свят празник за нашата мила родина, но по-важното почивни дни :D. В тези дни много хора решават да посетят различни места и в частност, Велико Търново. Един турист се характеризира с име, град, от който е, номер на колата, коефициент на стреса, преживян от цените на сувенирите на Самоводската и естествено кое кръстовище ще затапи. Напишете клас Tourist, който има полета за дадените опции. Има конструктор по подразбиране. И конструктор с параметри. Да се реализира и метод, който извежда, кое кръстовище ще затапи и времето, за което ще го. Времето се смята по следната формула: 0.8 * коефициент на стрес + ако номерът не започва с ВТ още 4 минути.

Задача 3

- Да се реализира клас StaticString, който съдържа низ с дължина най-много 255 елемента и следните методи. PrintString() - изпечатва низа, concat, който приема друг низ и ги конкатенира, ако сумарният им размер не надвишава лимита, конструктори по подразбиране и с параметър.

Задача 4

- Марти е изключително флеймнат от затварянето на улиците и затапването на алтернативите им от туристи. Ще разширим задача 2 като сега ще направим клас TouristPool, който съдържа масив с до 200 туристи и метод, който извежда всички кръстовища, които ще бъдат затапени и времето, за което ще бъдат. Напишете и метод input, който въвежда масива от клавиатурата.