

# Упражнение 1

## Групиране на данни чрез помощта на структури и обединения (unions)

### Структури

- Специален тип в C++, който ни позволява да групираме съвкупност от произволен тип данни, като всяка си има собствена памет, обединени от обща идея.

#### Синтаксис:

```
struct <type_name>
{
member_type1 member_name1;
member_type2 member_name2;
member_type3 member_name3;
...
...
} object_names;
```

Където:

- type\_name - задължително, ако нямаме object\_names
- object\_name - задължително, ако нямаме type\_name - (анонимни структури)

***За да бъде валидна дадена структура, то трябва поне една от двете точки да бъде изпълнена, но не задължително и двете.***

- Името на структурата трябва да обяснява за какво служи. Обикновено се приема името да започва с главна буква.
- Обръщение към даден член.
- Инициализация
- Заделяне на динамична структура

- Ако функцията няма да променя структурата, използваме винаги **const**

Задачи:

1. Напишете структура Point, съдържаща информация за координатите на точка в двумерното пространство.
2. Напишете функция, която въвежда координатите на дадена точка от конзолата.
3. Напишете функция, която извежда координатите на дадена точка на конзолата.
4. Напишете функция, която по подадени 2 точки като параметър, да връща разстоянието между тях.
5. Създайте масив от точки.
6. Напишете функции, които прочитат и извеждат масив от точки.

Допълнителна задача:

Създайте структура Vector, чрез създадената структура Point. Напишете функция за въвеждане/извеждане на вектор от/на конзолата, както и функции смятащи векторно и скалярно произведение.

## Union

- Специален тип в C++ наследен от C, който ни позволява върху една и съща памет да съществуват няколко променливи от произволен тип.

```
union SomeData
{
    int i;
    float f;
    char str[20];
};
```

```
union IP
{
    Int ip;
    char octet[4];
};
```