

Глобални (статични) - използват статични променливи
Локални - локални, променливи пром. за да се при комп.
Динамична - "временно" пром. която може да се яви по време от изпълнението на програмата
Управляем ход - при което се пада конспект, достига се функцион работи

ООП - програмна парадигма - която е една програмна система се моделира като набор от обекти, които взаимодействат помежду си

Namespace - инструмент за избягване на конфликти на имена
 - пространство от имена

namespace ns {
 void f() {}
 int global = 9;
 }
 Номен: namespace A {
 namespace B {
 f() {}
 }

Използване : using namespace ns; - въвежда всичко от namespace-a в м. а може на namespace-името код. Това може използването без използването на от за функцията
 using ns::f; - въвежда използването на дадените имен на namespace-a в дадения scope, без да въвежда използването на ::
 ns::f; - въвежда еднократно дадените имен на namespace-a
 :: - оператор за резултат

! Можемо използваме namespace-ове, свързвайки имената с имената на имената, които ще се компилира, за да ги използваме

Енумерации - дефинирани от програмиста

Enum тип данни който е ограничен до дадените от стойности, представяващи специално дефинирани константи (**enum-кони**)

! Всички enum-кони отговарят на едно число
 - ако не е даден стойност, то то е нула
 - ако на първия не е дадена стойност, то е 0

enum Color {
 red = 110,
 blue = 113,
 orange = 114,
 purple = red + blue; 113
 }

enum е **unscoped** - enum-кони са неща като глобални променливи, т.е. ще може достъп до тях от цялата

enum Color {
 orange
 }
 няма да се компилира

При unscoped enum няма имплицитно преобразуване от enum-кони към число

Пример: enum Color {
 orange = 1
 }

enum Animal {
 cat = 1
 }

Color c1 = Color::orange;
 Animal a1 = Animal::cat;

if (c1 == a1) {}
 // не се компилира

В повечето случаи това е нежелателно поведение, защото бързо се съпоставят отговорите на един и същи стойност, не забравяйте всички резултати

Вместо това може да се използва **enum class**

enum class Color {
 orange
 }
 enum class Fruit {
 orange
 } с OK

- ако искаме да присвоим стойността на enum-кони на променлива от конкретен тип, тогава трябва да им дадем експлицитен тип

! Имплицитно на променлива от тип enum (sizeof(enum)) - константата на имплементацията на променлива, в която дадените от стойности се използват enum-кони

Структури - последователност от полета, които се правят в определена ред

- тип-данните могат да бъдат от различни тип
 - когато са, от един тип представяне в паметта с едно и също като при един масив, разликата е само семантика

struct Point {
 int x = 0;
 int y = 0;
 }
 Point* ptr = new Point {3, 4};
 ! delete ptr;

Декларация на инстанция

Point p; // на тип-данните се присвояват по-големи стойности

Point p = {1, 3} // p.x = 1 p.y = 1

Point p;
 p.x = 2;
 p.y = 3;

Достъп до елементи : <името на инстанцията> . <името на елемент>

Може да създаваме динамични обекти :

Point* ptr = new Point();

Достъп до елементи :

(*ptr).x = 5;
 ptr->y = 10; } еквивалентни

Възможности на структури

struct Line {
 Point beg;
 Point end;
 };

Масиви от структури

Point arr[10]; // създава масив с 10 инстанции от тип Point

Point* ptr = new Point[10]; // създава указател към масив от инстанции от тип Point в heap-а

- големина на масив от инстанции е

sizeof(A) * sizeof(int) - защото

Правилна във функции

Структурата е негов тип, затова използване за създаване константи. Това е важно, защото, ако не използваме по-големи стойности (когато само не адреси и много по-големи)

f(Point& t)

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

double getDist(const Point& g, const Point& s)

f(100);
 f(const 100);
 f(A& ref);
 f(const A& ref);
 f(A* ptr);
 f(const A* ptr);

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности

Важно е, когато няма да променяме инстанцията във функцията да я поставим по-големи стойности