СТРУКТУРЫ, КЛАССЫ

*Struct Person {*

*char Name[50]*; // <- поля структуры

*int date\_b;*

*long int passport;*

*}*

*…  
Person A;*

*A.date = 2001;*

*A.passport =”111111”*

typedef struct Person // создание типа данных Person

(!) При создании структуры C++ автоматически увеличивает количество ячеек памяти до степени двойки.

(!) Чем изначально отличались классы от структур? В классах помимо полей были еще методы.

*struct student {*

*char \*Name;*

*int A[10];*

*double avg();*

*}*

*…*

*student N;*

*…*

*cout << N.avg()* //функция сработает не для всех студентов, а конкретно для одного студента N

*double student:: avg() {*

*double sum=0;*

*for (int i=0; i<10; sum+=A[i++]);*

*return sum/10;*

*}*

(!) Поле/класс типа public может предоставить доступ к данным снаружи

(!) В структуре по умолчанию все поля открыты, а в классе закрыты

СПЕЦИАЛИЗИРОВАННЫЕ МЕТОДЫ

1. DESTRUCTOR

* может подчищать данные вне области видимости
* его нужно определять, если в процессе работы программы выделялась дополнительная память

1. CONSTRUCTOR

* может иметь аргументы

*student (char\* str, int\* a) {*

*Name=new char[256];*

*for(int i=0; str[i]!=’\0’; i++) {*

*Name [i]=str[i];*

*for (int i=0; i<10; i++)*

*A[i]=a[i];*

*}*

(!) operator - функция, определенная для всех классов (Например, *A.operator = (B)* )

(!) *student (const student &a)* - конструктор копий

Копирование также можно осуществить: A=B

(!) Перегрузка равенств:   
A!=B аналогично !(A==B)

A<B сравнимо !(A>B)&&(A!=B)