ОБЕКТНО ОРИЕНТИРАН ПОДХОД ЗА ПРОГРАМИРАНЕ. КЛАСОВЕ. ОБЕКТИ
Наследяване.

Обектно ориентирано програмиране т.е начин на организиране на изчисленията.

1.Първи принцип на ООП-действие се активира чрез съобщение, което се предава на обект. Съобщението кодиразаявка с допълнителна информация, необходима за нея. Получателят приема със съобщението и отговорността за действието. В отговор се изпълнява метод, за да се удовлетвори заявката.

2.Втори принцип на ООП-всички обекти са екземпляри на клас. Кой метод ще се активира в отговор на едно съобщение се определя от класа на получателя и всички обекти от даден клас изпълняват един и същ метод за подобни съобщения.

Йерархия-по-долните класове имат своиствата на по-горните класове.

Принцип на наследяването – знанията за по общата категория са валидни и за по-специфичните.

3.Трети принцип на ООП- класовете могат да се организират в йерархична структора. Класът наследник наследява свойствата на класа родител. Абстрактен клас-клас, който се използва само за създаване на подкласове т.е. няма преки екземпляри.

Основни характеристики на ООП

1.Всичко е обект.

2.Изчислението се извършва от обекти, комуникиращи по между си, давайки заявки други обекти да извършват действието. Обектите комуникират по между си като приемат и получават съобщения. Съобщението е заявка за действие, заедно с аргументи, необходими за изпълнението на заявката.

3.Всеки обект има собственс памет, която се състои от други обекти.

4.Всеки обект е екземпляр от даден клас, който представлява групиране на подобни обекти.

5.Класът е склад за поведението,асоциирано с един обект т.е. всички обекти, които са екземпляри на един и същ клас магат да изпълняват едни и същи действия.

6.Класовете са организи-рани в дървовидна структора, наречена йерархия. Паметта и поведението, асоциирани с екземпляр на един клас автоматично са достъпни до всеки клас, по-надолу в йерархията т.е. се наследяват.

Видове абстрактни механизми:

Процедурни- позволяват извършване на операциите в цикъл;

Блокови структори- инициализираните в блок променлививажат смоза него.

Модули- пространството на имената се разделя на две : public(публични)- достъпни и за програмите извън модула, private (частни)- достъпни само за модула. Чрез модулите моигат да се правят екземпляри т.е. има само един стек.

Абстрактни типове данни – типове данни, декларирани от програмистта, който могат да

се обработват подобно на системно дефинираните типове данни.. Модулите са техника за реализация на абстрактни типове данни . за реализацията им са нужни:

-да експортираме дефиниции на тип;

-да представим множество от операции за обработка на екземпляри от типа;

-да защитим данните, така че да могат да се обработват чрез представените процедури;

-да създаваме много екземпляри на типа.

*Класове и обекти*

Чрез обектите адекватно се представят понятията и обектите сот реалния свят и обектната област. Класовете са типове състоящи се от компоненти, които са данни или функции. Функциите, принадлежащи на класовете наричаме методи. На един дефиниран клас могат да се създават екземпляри, наречени обекти.

Дефинирането на класове се състои от декларация на класа и дефиниране на неговите методи. Общ вид на декларацията:

class[]

{

;

```
}
```

Имената на класовете трябва да са уникални. Имената на компонентите са локални.

```
class Person

{

char name[50];

unsigned age;

public:

void getdata();

void display();

}

void Person::getdata()
```

```
{

coutname;

coutage;

}

void Person::display()

{

cout
```