Податоци од тип објект и класа

1)Објект е сложена структура на податок кој се опишува со два дела
Објект=Податоци+Методи (функции и процедури)

2)Класа е проширена податочна структура што содржи не само податоци, туку ги содржи и функциите што се применуваат врз тие податоци. Класа е множество на објекти што делат исти карактеристики, однесување, релации и функции.

Примери:
1) класа – ученик-податоци: име, презиме, низа со оценки по предмети,
методи – функција за пресметка на среден успех : збирот на оценките се дели со бројот на предмети;
објект – ученик од III-9 клас,или со име и презиме
2) класа – продавница-податоци: име, цена и количество за секој артикал, **методи** – функција за пресметка на попис во продавницита ...
објект – аптека , маркет и сл.
3) класа – возило , објект- автомобил, авион, воз и сл.

Пристап до класите:
public - членовите од класата се достапни за сите функции во програмата
private - членовите од класта се достапни само за функциите од иста класа
protected - членовите од класата се на располагање за сите функции од иста класа и поврзани функции на класи кои се добиени од почетната класа, а не се видливи за други функции

Дефинирање на класа во C++:

```
#include <iostream>
using namespace std;
class име на класата
{
  обезбедување на пристап до класата:
  листа на атрибути и однесување/функции;
};
```

Конструктори и деструктори

При креирање на објект во некоја класа, за негово користење потребно е да се предвиди (дефинира) т.н. конструктор, кој е всушност функциија –членки во класата и треба да го има истото име како што е името на класата. Основната разлика меѓу конструкторите и другите функции во класата е тоа што конструкторот не враќа никаква вредност и секогаш до него има пристап од типот на public.
Ако во класата не е дефиниран конструктор од самиот програмер, тогаш компајлерот во C++ набавува default constructor – без параметри, кој се повикува при самото дефинирањето на објект во некоја класа. (пр. ucenik objucenik;)

Бидејќи со самото дефинирање на класата-објектот се зафаќа меморија и доколку не ни треба класата –објектот, тој треба да се избриши, а тоа се прави со функција – **деструктор,** а тоа се прави со следната команда: **име на класата::~име на класата().** Пример ако сакаме да ја избришиме класата ucenik тогаш деструкторот ќе биде: **ucenik::~ucenik()** како наредба/команда во главниот дел од програмата.

Примери на програми со класи:

1) Класа со едноставна функција – членка, што нема влезни податоци и не враќа вредност, туку само печати податоци (за тоа се користи резервираниот збор – void):

```
#include <iostream>
using namespace std;
class  Pozdrav
{
public:   //пристап до класата
  void  Poraka()   // функција за порака
{
cout << "Klasa so ednostavna funkcija so poraka!"<< endl;
}
};   // завршува дефинирањето на класата
int  main()  //главна дел на програма (главна функција)
 {
   Pozdrav  objPozdrav;   //креирање на објект  со име objPozdrav од класата Pozdrav
  objPozdrav.Poraka ();  // повикување на функција – членка на класа
  return 0;
 }
```

2) Класа со функција – членка на која и требаат влезни вредности (параметри, аргументи-податоци) за да испечати некаков резултат:

```
#include <iostream>
using namespace std;
class  Kvadrat
{
public:  //пристап до класата
  void  Poraka(int broj)  // функција со порака што ќе печати квадрат на број
{
   cout << "Kvadratot na brojot e:"<< broj*broj<<endl;
}
};
int  main()
 {
  int br;
  cout<<”Vnesete broj”;
  cin>>br;
  Kvadrat  objKvadrat;  // креирање на објект  со име objKvadrat од класата Kvadrat
  objKvadrat.Poraka (br);   // повикување на функција – членка на класа
  return 0;
 }
```

3) Програма со која се пресметува плоштината, периметарот и должината на дијагоналата на правоаголник. При тоа, се креира класа правоаголник, која има податоци за должината и ширината на правоаголникот и три функции за пресметување на плоштината, периметарот и должината на дијагоналата.

```
#include <iostream>
#include <cmath>
using namespace std;
class Pravoagolnik
{
   public:
      float  visina, sirina;
```

```
      float  plostina(float visina, float sirina)
      {
          return visina * sirina;
      }

     float  perimetar(float visina, float sirina)
      {
          return  (2*(visina + sirina));
      }

      float  dijagonala(float visina, float sirina)
      {
         return (sqrt(pow(visina,2) + pow(sirina,2));
     }
};
int  main()
{
      Pravoagolnik   mal;
      float  v,s;
      cout<<"Vnesete sirina i  visina na pravoagolnikot"<<endl;
      cin>>s>>v;
      mal.sirina=s;  mal.visina=v;
      cout<< " Plostinata na pravoagolnikot e " << mal.plostina(v,s) << endl;
      cout<< " Perimetarot na pravoagolnikot e " << mal.perimetar(v,s) << endl;
      cout<< " Dijagonalata  na pravoagolnikot e " << mal.dijagonala(v,s) << endl;
  return 0;
}
```

4) Класа со податочни членки- (тоа се податоци – атрибути за сите објекти што припаѓа на класата, кои се дефинираат внатре во класата) и функциите-членки во класата set и get

```
#include <iostream>
using namespace std;
class  ucenik
{
public:
string Prezime,Ime; // декларирање  на податочните членки
float mat,mak,ang;  // декларирање  на податочните членки
// функција  set за поставување на податоци  внесени од тастатура  соодветно во објектот
void  setPodatoci(string P,string I,float o1,float o2, float o3)   {
   Prezime=P;
   Ime=I;
   mat=o1;
   mak=o2;
   ang=o3;
}
// функција get за добивање  на податочните членки од објектот
string getPrezimeIme()
{
   return Prezime+"  "+Ime;
}
float  Prosek(float  mat,float mak,float ang)
{
  float p;
  p=((mat+mak+ang)/3);
  return p;
}
};    // завршува дефинирањето на класата
```

```
int  main()
 {
  string P,I;
  float o1,o2,o3;
  ucenik  objucenik; // креирање на објект  со име objucenik од класата ucenik
  cout<<"Vnesi Prezime i ime na ucenikot"<<endl;
  cin>>P>>I;
  cout<<"Vnesi gi ocenkite po matematika, makedoski i angliski jazik"<<endl;
  cin>>o1>>o2>>o3;
  objucenik.setPodatoci(P,I,o1,o2,o3);   // повикување на функцијата set
  cout<<"Ucenikot  "<<objucenik.getPrezimeIme()<<"  ima  "<<objucenik.Prosek(o1,o2,o3)<< "  prosek";
  return 0;
  }
```

повикување на функција get

повикување на функција Prosek

5) Класа со име vraboten што содржи податочни членки : Prezime (string), Ime(string), Godvrab(int) и функции членки: set, get и функција членка што ќе пресметува работен стаж на вработениот во години.

```
#include <iostream>
#include <string>
using namespace std;
class  vraboten
{
public:
string Prezime,Ime;
 int  godvrab;
void  setPodatoci(string P,string I,int g)
{
   Prezime=P;
   Ime=I;
   godvrab=g;
}
string  getPodatoci()
{
   return Prezime+"  "+Ime;
}
  int  Staz(int godvrab)
{
  int p;
  p=2014-godvrab;
  return   p;
}
};

int  main()
 {
  string P,I;
  int g;
  vraboten  objvraboten;
  cout<<"Vnesi Prezime i ime na vraboteniot"<<endl;
  cin>>P>>I;
  cout<<"Vnesi ja godinata na vrabotuvanje"<<endl;
  cin>>g;
  objvraboten.setPodatoci(P,I,g);
  cout<< “Vraboteniot  “<< objvraboten.getPodatoci()<<"  ima  staz od "<< objvraboten.Staz(g)<<”godini”;
  return 0;
  }
```

6) Класа со име proizvod што содржи податочни членки : Imep (string), nab_cena(float) и функции членки: set, get и функција членка што ќе пресметува продажна цена на производот која е за 20% поголема од набавната цена.

```
#include <iostream>
#include <string>
using namespace std;
class  proizvod
{
public:
string Imep;
float  nab_cena ;
void  setPodatoci(string I, float nb)
{
  Imep=I;
  nab_cena=nb;
}
string  getPodatoci()
{
   return Imep;
}
int  cena (float  nab_cena)
{
    return   (nab_cena+0.2*nab_cena);
}
};

int  main()
 {
  string  I;
  float nb;
  proizvod   objproizvod;
  cout<<"Vnesi  go  proizvodot "<<endl;
  cin>>I;
  cout<<"Vnesi ja nabavnata cena"<<endl;
  cin>>nb;
  objproizvod.setPodatoci(I,nb);
  cout<< “Prodaznata cena na  “<< objproizvod.getPodatoci()<<"  e  "<< objproizvod.cena(nb);
  return 0;
  }
```