**КЛАС**

Декларация на клас:

class име на клас {

privatе //функции и променливи

public //функции и променливи

} списък от обекти;

Списък от обекти не е задължително да се декларира веднага с декларацията на класа, това може да стане и по-нататък. Функциите и променливите, които са декларирани в рамките на декларацията на един клас, се наричат членове на този клас. По подразбиране private са частни, т.e. те са достъпни само за други членове на този клас.За да се декларират публично членовете на един клас се използва ключовата дума public, следвана от двоеточие. Те са достъпни както за членовете на този клас, така и за всички други части от програмата съдържаща този клас.

Пример 1:

1. class myclass {
2. int a; //това е private променлива
3. public:
4. void set\_a(int num); //public функция set\_a, декларирана с прототипа си
5. int get\_a(); //public функция get\_a, декларирана с прототипа си
6. };

Функциите, декларирани като част от клас се наричат член-функции.

Тъй като променливата а е private, тя не е достъпна извън myclass. Но тъй като set\_a() и get\_a() са членове на myclass, то те имат достъп до а. Тъй като set\_a() и get\_a() са public, те могат да бъдат извиквани от всяка част на програмата която съдържа myclass. Въпреки, че set\_a() и get\_a() са декларирани от myclass те не са дефинирани.

**Дефиниране на член функция**

За да дефинирате дадена член-функция, трябва да свържете типа на класа с името на функцията. Това се прави, като разделяте името на класа и името на функцията с двойно двуеточие(::) Двойното двоеточие :: се нарича оператор за определяне областта на видимост.

Пример: Дефиниране на функциите set\_a() и get\_a()

1. void myclass::set\_a(int num)
2. {
3. a = num;
4. }
5. int myclass::get\_a()
6. {
7. return a;
8. }

**ОБЕКТИ**

Обектът е инстанция на даден клас. Обектът притежава идентичност (уникален екземпляр). Класът дефинира както интерфейса, така и реализацията на множество обекти. Класът определя поведението на всички негови обекти. След като обектът се създаде, той не може да променя класа към който принадлежи. Създаване на обект - за да създадете обект, използвайте името на класа като спецификатор за типа.

Пример: Следващият ред създава 2 обекта от тип myclass

myclass ob1, ob2;

Пример 1 :Напишете програма, която задава стойности на член-променливата а на класовете ob1 и ob2, след като извежда стойностите на а за всеки обект.

1. #include <iostream>
2. using namespace std;
3. class myclass {
4. int a; //частна променлива
5. public:
6. void set\_a(int num); //декларира функция set\_a
7. int get\_a(); //декларира функция get\_a
8. };
9. void myclass::set\_a(int num) //дефиниране на член функцията set\_a()
10. {
11. a = num; //тяло на функцията set\_a
12. }
13. int myclass::get\_a() //дефиниране на член функцията get\_a()
14. {
15. return a; //тяло на функцията get\_a
16. }
17. int main() //главна функция
18. {
19. myclass ob1, ob2; //декларира обектите ob1 и ob2
20. ob1.set\_a(10); //задава на а от ob1 стойност 10
21. ob2.set\_a(99); //задава на а от ob2 стойност 99
22. cout << ob1.get\_a() << endl; //изкарай на екран 10
23. cout << ob2.get\_a() << endl; //изкарай на екран 99
24. return 0;
25. }

Запомнете: Всеки обект от даден клас притежава свое собствено копие от всяка променлива, декларирана в този клас.

Пример 2: В тази програма а е декларирана в public частта на myclass и така обръщението към нея е възможно от която и да е част на програмата

1. #include <iostream>
2. using namespace std;
3. class myclass {
4. public:
5. int a; //сега а е public
6. // така, че set\_a() и get\_a() не са необходими !!!
7. };
8. int main()
9. {
10. myclass ob1, ob2;
11. ob1.a = 10; //сега имаме директен достъп до а
12. ob2.a = 99; //сега имаме директен достъп до а
13. cout << ob1.a << "\n";
14. cout << ob2.a << "\n";
15. return 0;
16. }

Пример 3: Програмата създава клас card, който има структура на библиотечен картон. Класа съхранява името, автора и броя на наличните екземпляри от книгата. Името и автора са променливи от тип низ, а броя на наличните екземпляри – целочислен тип. Тук е използвана public член-функция store, за да съхрани информация за книгата, и public член-функция show, която показва информацията. Добавянето на функцията main демонстрира класа.

1. #include <iostream>
2. #include <cstring>
3. using namespace std;
4. class card {
5. char title[80]; // име на книгата
6. char author[40]; // автор
7. int number; // наличност
8. public:
9. void store(char \*t, char \*name, int num); //член-функция store
10. void show(); //член-функция show
11. };
12. void card::store(char \*t, char \*name, int num) //дефин. на чл.ф. store
13. {
14. strcpy(title, t);
15. strcpy(author, name); //тяло на чл.ф. store
16. number = num;
17. }
18. void card::show() //дефиниране на чл.ф show
19. {
20. cout << "Title: " << title << "\n";
21. cout << "Author: " << author << "\n"; //тяло на чл.ф. show
22. cout << "Number on hand: " << number << "\n";
23. }
24. int main() //сега демонстрираме класа
25. {
26. card book1, book2, book3; //създаваме обектите
27. book1.store("Dune", "Frank Herbert", 2); //извикваме функцията store
28. book2.store("The Foundation Trilogy", "Isaac Asimov", 2);
29. book3.store("The Rainbow", "D. H. Lawrence", 1);
30. book1.show(); //извикваме функцията show
31. book2.show();
32. book3.show();
33. return 0;
34. }

**Задача 1** Създайте клас, който да съдържа име и адресна информация. Съхранявайте информацията в символни низове, които са private членове на класа. Включете public член-функция, която да запазва името и адреса. Включете и public член-функция, която да извежда името и адреса (наречете тези функции store() и display().)

**Присвояване на обекти**

Един обект може да бъде присвояван на друг при условие, че и двата обекта са от един и същи тип.

Пример 4:

1. #include <iostream>
2. using namespace std;
3. class myclass {
4. int a, b;
5. public:
6. void set(int i, int j) { a = i; b = j; }
7. void show() { cout << a << ' ' << b << "\n"; }
8. };
9. int main()
10. {
11. myclass o1, o2;
12. o1.set(10, 4); /\*обектът o1 притежава член-променливите а и b със стойности 10 и 4\*/
13. o2 = o1; /\*присвоява o1 на o2, т.е ст-та на о1 а се присвоява на о2 и текущата ст-т на о1.b се присвоява на о2.b\*/
14. o1.show();
15. o2.show();
16. return 0;
17. }

**Предавне на обекти към функции**

Обектите могат да бъдат подавани като аргументи на функции по същия начин, по който се предават и всички други типове данни. Просто декларирайте параметъра на функцията като тип клас и след това използвайте обект от този клас като аргумент при извикването на функцията. Обектите, както и останалите типове данни по подразбиране се подават на функцията по стойност.

Пример 5: Тази програма демонстрира подаването на обект към функция

1. #include <iostream>
2. using namespace std;
3. class samp {
4. int i; //целочислена променлива i
5. public:
6. samp(int n) { i = n; }
7. int get\_i() { return i; }
8. };
9. int sqr\_it(samp o) //връща стойността на o.i на квадрат
10. {
11. return o.get\_i() \* o.get\_i(); //квадрат
12. }
13. int main()
14. {
15. samp a(10), b(2); //обекти а и b
16. cout << sqr\_it(a) << endl;
17. cout << sqr\_it(b) << endl;
18. return 0;
19. }

Връщане на обекти като резултат от функция

За тази цел първо трябва да дефинирате функцията като функция, която връща резултат от тип клас, също така като резултат трябва да върнете обект от декларирания тип, като използвате обикновената конструкция return.

Пример 6: Тази програма демонстрира функция, която като резултат връща един обект

1. #include <iostream>
2. #include <cstring>
3. using namespace std;
4. class samp {
5. char s[80];
6. public:
7. void show() { cout << s << endl; }
8. void set(char \*str) { strcpy(s, str); }
9. };
10. samp input() //връща обект от тип samp
11. {
12. char s[80];
13. samp str;
14. cout << "Enter a string: ";
15. cin >> s;
16. str.set(s);
17. return str;
18. }
19. int main()
20. {
21. samp ob;
22. ob = input(); //присвоява върнатия обект на ob
23. ob.show();
24. return 0;
25. }

**Задача 2**

Дефинирайте клас коли(car), който има следните променливи: марка, модел, година на производство, цвят, тип скоростна кутия(автоматик или ръчна), мощност на двигателя(в конски сили), тип на гориво(бензин/дизел/газ или газ/бензин),тегло и колко километра е извървяно с нея. Всички променливи да са в полето private, въведете тяхната стойност чрез get и set и я изведете.