

КЛАСОВЕ И ОБЕКТИ

1. Цел на упражнението

Усвояване на правилата и принципите на проектиране, деклариране, дефиниране и използване на класове и обекти в C++.

2. Декларация на клас

Класовете в C++ са типове, състоящи се от множество компоненти, които могат да бъдат данни и функции. За разлика от основните типове, класовете не са създадени предварително, а се дефинират от програмиста в зависимост от конкретните нужди. Декларирането на класове е еквивалентно на добавяне на нови типове към езика.

Класът е основната програмна единица в C++. Описанието на клас има следния синтаксис:

```
class <име_клас>
{
    public:           // секция на глобалните данни
                     // декларации на данни и функции

    protected:      // секция на данните с ограничен достъп
                     // декларации на данни и функции

    private:         // секция на локалните данни
                     // декларации на данни и функции
};
```

Обикновено членовете-данни се намират в секция **private**, а членовете-функции (методите) - в секция **public**. Механизмът, който ограничава достъпа до членовете на класа, се нарича "скриване на информацията".

Членове-данни (атрибути): Членовете, които описват класа се дефинират като обикновени променливи, но не могат да се инициализират явно.

Членове-функции (методи): Методите се дефинират непосредствено в класа или в тялото на класа се посочва техният прототип. В последния случай дефиницията на функциите е извън класа, като тя трябва да бъде разположена в изходния файл след декларацията на класа.

Когато методът се дефинира извън класа, в заглавието на описваната функция трябва да се посочи името на класа, за който тя се отнася.

```
<тип_резултат> <име_клас> :: <име_функция> ( <списък_параметри> )
{
    <тяло на функцията>
}
```

Методите на даден клас осигуряват множеството операции за обектите от този клас. Операциите се наричат общодостъпен интерфейс (public interface) на класа. Дали класът е добре конструиран, зависи от пълнотата и ефективността

на неговите методи. Методите условно се делят на: управляващи, инструментални, помощни и методи за достъп. Разделението не е част от езика C++, начин на мислене при проектиране на класове.

Вторият основен механизъм, заложен в C++, е, че класът представлява един абстрактен тип данни. Класът се състои от декларативни описания, като реализацията на членовете-функции е изнесена извън класа. Това не е направено само за удобство. Основното е, че комуникацията с представителите на класа (обектите) се осъществява единствено чрез членовете-функции. Ако се промени реализацията на един член на класа, това няма да се отрази на комуникацията с другите представители на класа. Вторият основен механизъм се нарича капсулиране - структурата от данни, която формира класът и операциите върху тези данни, са затворени в една езикова конструкция.

3. Обекти

След като даден клас е деклариран, могат да бъдат създавани негови екземпляри (конкретни представители), които се наричат обекти. Отношението между клас и обект в езика C++ е същото, каквото е отношението между тип и променлива. Създаването на обекти от даден клас е аналогично на дефинирането на променливи от даден тип.

<име_клас> <име_променлива>;

Достъпът до данните на един обект се осъществява изключително чрез неговите методи. Тъй като имената на тези функции са еднакви за всички обекти от даден клас, при използването им трябва да се посочва и името на обекта, за който се отнасят. Обръщението към член-функция (метод) на даден обект има следния синтаксис:

<име_обект>.<име_функция>(<фактически параметри>);

Достъпът до компонентите на обектите може да се осъществява чрез указател:

<име_обект> -> <име_функция>(<фактически параметри>);

4. Задачи за изпълнение

Задача 1:

Да се декларира клас **Person**. Членове-данни (атрибути) на класа са: името на лицето и неговата възраст. Методите на класа включват функции за: въвеждане информация за лицето (**setData**); извеждане на информация за лицето (**display**). Да се дефинират методите на класа. Да се състави главна програма за тестване на дефинираните методи.

Да се създаде проект с име **EX31**.

Задача 2:

Да се декларира клас **ComplexNum** (комплексно число). Членове данни на класа са реалната и имагинерната част на числото. Методите на класа включват функции за: въвеждане на елементите на числото; събиране на две комплексни числа; умножение на две комплексни числа; изчисляване на модул на комплексно число; извеждане елементите на числото. Да се дефинират

методите на класа. Да се състави главна програма за тестване на дефинираните методи.

Да се създаде проект с име EX32.

Задача 3:

Да се декларира клас **Time** (астрономическо време). Членове-данни на класа са: час (*hour*), минути (*minute*), секунди (*second*). Методите на класа включват функции за: въвеждане елементите на класа (да се предвиди контрол на входните данни, ако данните не са допустими, въвежда се 0); извеждане в универсален формат - чч:мм:сс; извеждане в стандартен формат - чч:мм:сс AM/PM. Да се дефинират методите на класа. Да се състави главна програма за тестване на дефинираните методи.

Да се създаде проект с име EX33.

Задача 4:

Да се декларира клас **SalesPerson** (продавач).

Членове-данни на класа са: име, месечните продажби в продължение на една година (масив от 12 елемента от тип *double*). Методите на класа включват функции за: въвеждане елементите на класа (въвеждане сумата на продажбите за 12-те месеца); извеждане общата сума на продажбите за 12-те месеца. Да се дефинират методите на класа. Да се състави главна програма за тестване на дефинираните методи.

Да се създаде проект с име EX34.

Задача 5:

Да се декларира клас за обслужване на екранни прозорци. Да се предвидят членове-данни за съхраняване размерите на прозореца, масив от символи, запълващ екрана и указател към текущата позиция на курсора. Да се предвидят методи за инициализиране данните на обекта и за запълване на прозореца със зададен фонов символ. Да се добавят инструментални методи, поддържащи движение на курсора до конкретна точка от екрана: *move()*, *forward()*, *back()*, *up()*, *down()*. Да се предвидят методи за достъп, осигуряващи потребителски достъп до скритите данни на класа. Да се дефинират функции, които да записват отделен символ или символен низ от текущата позиция на курсора. Да се дефинира набор предикатни функции, които връщат истина при съвпадение на текущия символ със зададен символ или ако от текущата позиция започва символен низ, съвпадащ със зададен низ или ако зададен екран съвпада с текущия.

Да се дефинира масив от обекти от дефинирания клас и се състави програма за задаване на N елемента от този масив с различни данни. Да се тества съставената програма.

Задача 6:

Да се декларира клас **NIZ** (символен низ) с членове-данни максимална дължина и указател към началото на символния низ и методи за определяне дължината на низа, отстраняване на определен символ от низа, слепване (конкатенация) със зададен символен низ.

Да се дефинира външна за класа функция за сравняване на два обекта от дефинирания клас.