Разширен конспект за упражнението по
Програмиране (C++)
на 25.10.2021 година
за специалност Софтуерно инженерство, редовно, I курс

Кирил Иванов

⧩ *Основни преговаряни понятия и подтеми*

◊ Термините информация и данна се използват с много значения, включително и в съвременната наука.

В компютърното програмиране:

— информация най-общо наричаме всяко отражение (образ) на нещо от действителността;

— данна наричаме знакова конструкция, подлежаща на обработка, а самите обработки избираме така, че данната да съответства на някаква информация (да назовава информацията).

Компютри обработват данни без възможност да проследяват връзките данна ↔ отражение на действителността ↔ част от действителността. Използването на думата компютър в друг смисъл излиза извън рамките на компютърната информатика.

◊ оператор, операнд, операция;

◊ именувана константа и изброен тип (enumerated type; т. е. деклариран чрез `enum`);

◊ цикъл по диапазон;

◊ генериране на случайни числа.

⧩ *Основни въвеждани понятия и идеи*

▷ базова схема на фон Нойманова архитектура: централен процесор; регистър (клетка памет, вградена в процесорно устройство); оперативна (вътрешна) памет; периферни устройства; портове (буферна памет, работеща със скоростта на оперативната) между централния процесор и периферните устройства); шина („сноп“ линии, по които физически едновременно се предават цифри, по една цифра по всяка линия); шина за адреси, шина за данни и шина за управляващи сигнали; адрес (редица от цифри, т. е. – цяло неотрицателно число); байт (минимална адресируема клетка памет в оперативната памет или в портовете);

▷ необходимост от използване на адреси (например за междинни данни, създавани в процеса на изпълнение) в езиците от ниско и от високо ниво;

▷ четири сегмента от памет, предоставяни на изпълняваната програма – сегмент с памет за изпълним код, сегмент за статична памет (за статични данни), сегмент за стек (паметта от него наричаме още стекова памет) и сегмент за динамична памет (за данни, появяващи се едва по време на изпълнението);

▷ указател;

▷ качествена разлика между адрес (цяло неотрицателно число) в машинния език и указател (адрес, асоцииран с тип на данната, намираща се на този адрес) в език от ниво над машинното;

▷ създаване и унищожаване на динамични данни, константа `NULL` (показани са и в ***Пример 5.1***);

▷ строга типизация при указателите (например типът `int*` е несъвместим с `long*`), но на синтактично ниво е допустимо явно преобразуване от един към друг тип на указател (като понякога това няма смисъл), например `(int*)(long*)...` или `(short*)(long*)...`;

▷ основана идея за предефинирането на оператори (operator overloading) в C++ (илюстрирана в ***Пример 5.10***);

Предефинирането на оператори е възможно, когато *поне единият* операнд е от изброим тип (дефиниран чрез enum) или от тип клас.

▷ преимущество на предоставянето в езика на някое действие чрез оператор, вместо чрез команда;

▷ псевдоним (показан е и в ***Пример 5.2***);

▷ `double const` по смисъл е еквивалентно на `double const`;

▷ В една и съща декларация, която въвежда едновременно няколко имена, названието на тип се отнася за всички имена, но всеки използван оператор се отнася само за едно име.

Например в декларацията

```
double d = 1.125, * u1 = & d, * * u2 = & u1;
```

името `double` се отнася и за `d`, и за `u1`, и за `u2`, а `*` се отнася само за `u1` и `* *` се отнася само за `u2`.

▷ за да бъде някой указател константен, трябва ключовата дума `const` да се използва между оператора * и името на указателя (например в `int const * u, * const c;` указателят `u` може да се променя, а указателят `c` е константа, но и двата указателя „сочат“ константа от тип `int`);

▷ масив, елемент на масив, индекс или индекси на елемента на масив, дължина на масив, едномерен и многомерен масив, размерност на масив;

▷ в едномерния масив индексът е точно броя на елементите в масива, разположени преди назоваванията чрез индекса (така индексът `n` в `ar[n]` означава, че в масива, назоваван с името `ar`, има точно `n` елемента преди `ar[n]`);

▷ начини за инициализация на масив (показани са в ***Пример 5.6***);

▷ безразмерен масив; терминът се отнася само за синтактичния начин за задаване на дължината на масива, но това е масив от същия вид, както при явно назоваване на дължината;

▷ Операторът `sizeof` за име, декларирано с квадратни скоби, дава стойност броя на байтовете, заети от всички елементи на масива (това е показано в ***Пример 5.6***).

Например след

```
int iar[10];
long long LLar[10];
```

изразът `sizeof(iar)` има стойност 40,

а изразът `sizeof(LLar)` има стойност 80.

Съответно и изразът `sizeof(iar)/sizeof(iar[0])`, и изразът `sizeof(LLar)/sizeof(LLar[0])` имат стойности 10.

▷ цикъл по диапазон с променлива псевдоним *може* да променя редицата, задаваща диапазона.

⧩ *Основни средства на езика C++, въвеждани в това занятие*

- унарни оператори `*` и `&` в декларация и в израз (показани са и в ***Пример 5.1*** и ***Пример 5.2***);
- оператори `new` и `delete` (показани са и в ***Пример 5.1***);
- константа `NULL` (показана е и в ***Пример 5.1***);
- оператори `==` и `!=` с операнди указатели (показани са и в ***Пример 5.1***);
- оператор `[]` в декларация и в израз (показан е и в ***Пример 5.6***);
- варианти и правила за инициализация на масив (показани са в ***Пример 5.6***);
- `sizeof(име_декларирано_с_квадратни_скоби)` (показано е в ***Пример 5.6***).

⧩ *Задачи*

• *Пример 5.1*

Примерът показва използването на указатели и операторите `new` и `delete` (без аритметиката с указатели, тя ще бъде показана след запознаването с масиви).

Файл с примера: `progr21_05_01_demo.cpp`

• *Пример 5.2*

Примерът показва работа с псевдоним.

Файл с примера: `progr21_05_02_demo.cpp`

• *Задача 5.3*

Да се създаде приложение, което въвежда три числа, а след това:

— създава *указатели* съответно към минималното и максималното от тях и *само* с помощта на тези указатели извежда минималния затворен интервал, съдържащ числата;

— създава *псевдоними* съответно към минималното и максималното от тях и *само* с помощта на тези псевдоними извежда минималния затворен интервал, съдържащ числата.

Файл с примерно решение: `progr21_05_03.cpp`

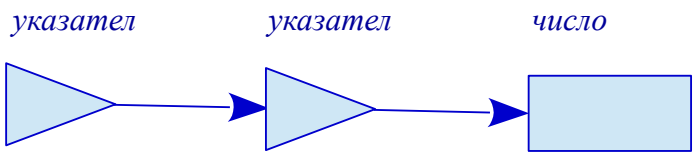
• ***Задача 5.4***

Да се създаде приложение, което въвежда две числа и ги извежда в нарастващ ред чрез една и съща команда за извеждане и в нея назовава двете числа *чрез една и съща* променлива от тип псевдоним на число.

Файл с примерно решение: `progr21_05_04.cpp`

• ***Задача 5.5***

Да се напише приложение, което създава следните три конструкции от данни от вида:



Първо, като и трите данни са назовани чрез имена в текста на приложението.

Второ, като десният указател и числото са създадени по време на изпълнение.

Трето, като пак десният указател и числото са създадени по време на изпълнение, но и двете динамични данни са константи.

И в трите случая да се извежда числото, като се назовава *само* чрез левия указател (чрез указателя към указател).

Файл с примерно решение: `progr21_05_05.cpp`

• ***Пример 5.6***

Примерът показва:

— различни начини за инициализиране на едномерен масив от цели числа;

— извеждане на масив;

— цикли for по индекс и по диапазон за обхождане на масив;

— получаване на дължината на масив (на броя на елементите му) чрез оператор `sizeof`.

Файл с примера: `progr21_05_06_demo.cpp`

• ***Задача 5.7***

Да се създаде приложение, което въвежда с контрол на стойността k цели, четни числа и ги извежда в ред, обратен на реда на прочитане.

Броят k да се зададе чрез константа в програмния текст.

Файл с примерно решение: `progr21_05_07.cpp`

• ***Задача 5.8***

Да се създаде приложение, което:

— *чрез цикъл по диапазон* записва в едномерен масив десет случайни числа, всяко от `-3` до `4`;

— извежда всички числа;

— извежда максималния брой пъти на генериране на едно и също число;

— извежда всички числа (допуска се многократно извеждане на едно и също число), които са генерирани максимален брой пъти.

Файл с примерно решение: `progr21_05_08.cpp`

• ***Задача 5.9***

Да се създаде приложение, което създава и извежда масив от 8 случайни цели числа, всяко от -3 до 1, а след това извежда *само по един път* всяко число, което се среща в масива.

Файл с примерно решение: `progr21_05_09.cpp`

• ***Пример 5.10***

Примерът показва предефинирането на операторите „`+`” и „`!`” за константите от конкретен изброим тип.

Файл с примера: `progr21_05_10_demo.cpp`