

**ПРАКТИКА ИСПОЛЬЗОВАНИЯ УКАЗАТЕЛЕЙ В C++***БНТУ, г. Минск**Научный руководитель: канд. техн. наук, доцент Дробыш А. А.*

Для начала надо определить, что понимается под понятием «указатель». Например на языке C++ указатель – это та же переменная, только инициализируется она не значением одного из множества типов данных в C++, а адресом некоторой переменной, которая была объявлена в коде ранее. Адрес переменной – это путь, по которому находится значение самой переменной. Он записывается в шестнадцатеричном виде. При этом при каждом упоминании переменной в коде программы, компилятор автоматически обращается к этому адресу. Указатели можно объявлять и инициализировать в любом месте кода программы. Для введения указателей в код программы используются следующие операторы:

\* – показывает значение переменной по заданному адресу. Если вы используете оператор \*, то вы занимаетесь операцией разыменования указателя;

& – показывает адрес переменной.

Процедура создания указателей на языке программирования C++ выглядит следующим образом:

- объявление переменной любого типа;
- с помощью оператора \* (звездочка) указываем компилятору, что будет использоваться тип данных – указатель;
- дальше указывается имя переменной;
- после знака равно передаем указателю адрес какой-то переменной, что делается с помощью оператора &(амперсанд).

Для передачи адреса какой-то переменной, от одного указателя другому, необходимо опускать оператор \* для одного указателя (от которого передаем второму). Например:

```
int a = 56;  
int *b = &a;  
int *c = b;  
cout << *c;
```

При разработке сложноструктурированных программ уместно использовать указатели в следующих случаях:

- Ссылочная семантика. Иногда может возникнуть необходимость обращения к объекту (вне зависимости от того, как под него распределена память), поскольку вы хотите обратиться в функции именно к этому объекту, а не его копии – то есть когда вам требуется реализовать передачу по ссылке. Но если вы можете обратиться к копии объекта, то и ссылку использовать нет необходимости, однако копирование объекта – это ресурсозатратная операция.

- Использование указателей при работе с массивами. Запись  $a[b]$  всегда эквивалентна  $*(a + b)$ . Таким образом, запись  $x[2]$  означает следующее:  $x[2]$  эквивалентно  $*(x + 2)$ ;  $x + 2$  относится к тем операциям, при которых имя массива преобразуется в указатель на его первый элемент.

- С помощью указателя массив нельзя передать как аргумент в функцию.

- Указатель на массив. Под фразой «указатель на массив» неформально понимают указатель на область памяти, в которой размещён массив, даже если тип у этого указателя неподходящий.

- Многомерные массивы. Они задаются с помощью двойных указателей. Например:

```
int **y = new int *[5];
for (int i = 0; i != 5; ++i)
{
    y[i] = new int[7];
}
```

- Зачастую при решении какой-либо задачи возникает потребность в использовании массива неопределённого размера, то есть размер этот заранее неизвестен. Здесь нам на помощь приходят динамические массивы – память под них выделяется в процессе выполнения программы.

- Операции над адресами. С помощью указателей можно осуществлять следующие арифметические операции:

- сложение и вычитание указателей с константой. Сложение и вычитание указателей с константой  $n$  означает, что указатель перемещается по ячейкам памяти на столько байт, сколько занимает  $n$  переменных того типа, на который он указывает;

– вычитание одного указателя из другого определяет, сколько переменных данного типа размещается между указанными ячейками.

– Инкремент перемещает указатель к следующему элементу массива.

– Декремент перемещает указатель к предыдущему элементу массива.

– Операции отношения. Указатели можно сравнивать таким же образом переменные и константы.

Итак, можно сделать вывод, что указатели широко применяются при написании кода на различных языках программирования, в частности на языке C++. Однако, наибольшую эффективность указатели приносят в умелых руках.

УДК 621.762.4

Кислушенко А. В.

## **ТРЕНДЫ В ТЕСТИРОВАНИИ В 2019 ГОДУ**

*БНТУ, г. Минск*

*Научный руководитель: канд. техн. наук, доцент Дробыш А. А.*

Профессия тестировщика претерпела множество изменений. В XX веке все началось с ручного тестирования и каскадного подхода. Сейчас автоматизация и развитие искусственного интеллекта привели к переменам в ИТ-индустрии и внесли новый смысл в профессию QA-инженера.

Изменились и задачи, которые стоят перед тестированием. Раньше все ограничивалось поиском максимального количества дефектов. Согласно World Quality Report 2018-2019, подготовленному компаниями Micro Focus, Capgemini и отделом Sogeti, современные тренды в тестировании основаны на стремлении улучшить пользовательский опыт. Удовлетворение потребителей – главная задача, которая движет развитием тестирования ПО.

QA-инженерам важно быть в курсе трендов и вовремя адаптироваться к ним, чтобы оставаться востребованными на рынке. Итак, давайте взглянем на тенденции в тестировании, на которые стоит обратить внимание в 2019 году.

– Переход бизнеса на цифровые технологии и использование API;