一、decltype意义

有时我们希望从表达式的类型推断出要定义的变量类型，但是不想用该表达式的值初始化变量（如果要初始化就用auto了）。为了满足这一需求，C++11新标准引入了decltype类型说明符，它的作用是选择并返回操作数的数据类型，在此过程中，编译器分析表达式并得到它的类型，却不实际计算表达式的值。

二、decltype用法

1.基本用法

```
int getSize();

int main(void)
{
 int tempA = 2;

 /*1.dclTempA为int*/
 decltype(tempA) dclTempA;
 /*2.dclTempB为int，对于getSize根本没有定义，但是程序依旧正常，因为decltype只做分析，并不调用getSize，*/
  decltype(getSize()) dclTempB;

 return 0;
}
```

2.与const结合

```
double tempA = 3.0;
const double ctempA = 5.0;
const double ctempB = 6.0;
const double *const cptrTempA = &ctempA;

/*1.dclTempA推断为const double（保留顶层const，此处与auto不同）*/
decltype(ctempA) dclTempA = 4.1;
/*2.dclTempA为const double，不能对其赋值，编译不过*/
dclTempA = 5;
 /*3.dclTempB推断为const double * const*/
 decltype(cptrTempA) dclTempB = &ctempA;
 /*4.输出为4（32位计算机）和5*/
 cout<<sizeof(dclTempB)<<" "<<*dclTempB<<endl;
 /*5.保留顶层const，不能修改指针指向的对象，编译不过*/
 dclTempB = &ctempB;
 /*6.保留底层const，不能修改指针指向的对象的值，编译不过*/
 *dclTempB = 7.0;
```

3.与引用结合

```
int tempA = 0, &refTempA = tempA;

/*1.dclTempA为引用，绑定到tempA*/
decltype(refTempA) dclTempA = tempA;
/*2.dclTempB为引用，必须绑定到变量，编译不过*/
decltype(refTempA) dclTempB = 0;
/*3.dclTempC为引用，必须初始化，编译不过*/
decltype(refTempA) dclTempC;
/*4.双层括号表示引用，dclTempD为引用，绑定到tempA*/
 decltype((tempA)) dclTempD = tempA;

 const int ctempA = 1, &crefTempA = ctempA;

 /*5.dclTempE为常量引用，可以绑定到普通变量tempA*/
 decltype(crefTempA) dclTempE = tempA;
 /*6.dclTempF为常量引用，可以绑定到常量ctempA*/
 decltype(crefTempA) dclTempF = ctempA;
 /*7.dclTempG为常量引用，绑定到一个临时变量*/
 decltype(crefTempA) dclTempG = 0;
 /*8.dclTempH为常量引用，必须初始化，编译不过*/
 decltype(crefTempA) dclTempH;
 /*9.双层括号表示引用,dclTempI为常量引用，可以绑定到普通变量tempA*/
 decltype((ctempA)) dclTempI = ctempA;
```

4.与指针结合

```
int tempA = 2;
int *ptrTempA = &tempA;
/*1.常规使用dclTempA为一个int *的指针*/
decltype(ptrTempA) dclTempA;
/*2.需要特别注意，表达式内容为解引用操作，dclTempB为一个引用，引用必须初始化，故编译不过*/
decltype(*ptrTempA) dclTempB;
```

三、decltype总结

decltype和auto都可以用来推断类型，但是二者有几处明显的差异：1.auto忽略顶层const，decltype保留顶层const；2.对引用操作，auto推断出原有类型，decltype推断出引用；3.对解引用操作，auto推断出原有类型，decltype推断出引用；4.auto推断时会实际执行，decltype不会执行，只做分析。总之在使用中过程中和const、引用和指针结合时需要特别小心。