9. ***对普通函数和虚函数进行的区别实验如下：***

#include<stdio.h>

class A{

public:

int aaa = 233;

public:

**//virtual void func(int a,char b){printf("A\n");}**

void func(int a,char b){printf("A\n");}

};

class B:public A{

public:

int aaa = 344;

void func(int a,char b){printf("B\n");}

int func(){printf("BB\n");}

};

int main(){

A zjs1;

B zjs2;

A\* pa = &zjs2;

B\* pb = &zjs2;

pa->func(1,1); //A //用加粗的那句话就显示为B

pb->func(1,1); //B

pb->func(); //BB

printf("%d\n",pa->aaa);

printf("%d\n",pb->aaa);

A\* paa = &zjs1;

//B\* pbb = &zjs1;

paa->func(1,1); //A

return 0;

}

结论：1.父类指针可以指向子类但反之不行

2.父类指针只能访问子类中的父类成员（变量和函数都是这样）

3.子类指针可以访问子类和父类中的成员

4.子类和父类的成员（再次提醒成员包括变量和函数）都可以重名（注意此处的函数重名是指完全相同，而不仅仅是函数名称相同，完全相同是指函数名、返回值类型、参数个数以及类型全部相同），因为他们处在两个域之中，很难相互干扰

5.想要让这两个域产生交互的唯一办法就是用虚函数（没有虚变量）

6.只有父类的函数可以用虚函数，子类函数用虚函数只对子类的子类起作用，而不能对其父类起作用

7.父类的虚函数将在被子类继承时改写（如果子类有与之同类型且同参数且同名的成员函数的话。这里要注意，只是同名的两个函数并不一定是同一个函数，只要类型和参数不同，那么他们就是两个完全不同且没有联系的函数）成子类成员函数的内容，注意，父类的这个虚函数并未消失，只是函数体遭到了改写，这样唯一的作用就是，用父类指针调用父类域下的该函数时，能够实现，和用子类指针调用子类域下的该函数，相同的效果而已。当然，如果子类没有定义相同函数，就不会改写父域下的该虚函数。

9.对于函数来说，如果父域子域中存在相同函数，那么子类指针会优先调用子域中的该函数，如果存在父域中有而子域中没有的函数，那么子类指针会调用父域中的该函数；对于父类指针来说，它只会调用父域中的函数，子域中的所有东西对它来说都是透明的。

10.对于变量来说，调用规则与函数完全相同。

11.经常会使人困惑的一个点就在于，函数不仅可以相同，还可以重名而不相同，也就是重载，比如父域中有一个func，子域中有一个重载了的func而没有与父域func相同的func，此时，在用子类指针调用func时，我们希望的是它根据参数列表去调用父域中的func，但是很可惜这是做不到的，编译器会报错，无论把父域func写成实虚函数都是如此。我的理解就是，编译器不够智能，它在将子类指针从子域重定向到父域时只会根据函数名判断，不回判断两个重名函数是不是真的相同，一旦找到了子域中的重名函数，它就不再往父域中重定向了，因此就会出现函数参数传递错误之类的报错提示。详见下列代码：

#include<stdio.h>

class A{

public:

int aaa = 233;

void func(int a,int b){

printf("A\n");

}

};

class B:public A{

public:

int aaa = 344;

int func(int a){

printf("B\n");

}

};

int main(){

B b;

A\* pa = &b;

B\* pb = &b;

printf("%d\n",pa->aaa);

printf("%d\n",pb->aaa);

pa->func(1,2);

pb->func(1,2);

}//会报错的，但是你把B中的func删掉再执行pb->func(1,2)就会通过。

12.一句话总结：父用父，子用子与父，子用时有重名（注意是重名）先用子，没有重名再用父，这个重定向不智能，只能根据变量或函数的名称重定向，这对于变量是足够的，但是对于函数来说却考虑不到参数列表和返回类型这两点。虚函数只有一作用——让子域中的，与父域中相同的函数（注意这里不是重名了，是想通了）覆写父域中的函数。

***那么纯虚函数在父子类之间又如何呢？***

有纯虚函数的类叫做抽象类，抽象类不能定义对象

抽象类也有自己的域，抽象类里面除了纯虚函数，也可以由普通函数和虚函数

抽象类中的普通函数和虚函数与子类的关系分析方式同上

抽象类中纯虚函数约束子类，要求子类必须实现这个函数

这个函数在抽象类里只有声明没有实现，在子类中实现，这个函数比较特殊，无论是用父类指针还是子类指针都可调用这个函数