//实例113 动物对象进化（继承）

#include <iostream>

using namespace std;

class animal\_ancestor //基类

{

public:

animal\_ancestor(){}

~animal\_ancestor(){}

void respire()

{

cout << "可以呼吸" << endl;

}

};

class ape:public animal\_ancestor //由基类派生的子类

{

public:

ape(){}

~ape(){}

void move()

{

cout << "可以行走" << endl;

}

};

class human:public ape

{

public:

human(){}

~human(){}

void miss()

{

cout << "可以想念某人" << endl;

}

};

void main()

{

human \_human;

cout << "人类";

\_human.respire();

\_human.move();

\_human.miss();

}

//员工月薪发送（多态）

#include <iostream>

using namespace std;

class salary

{

public:

salary(){}

~salary(){}

virtual void pay(){}

private:

};

class member1:public salary

{

public:

member1(double a){\_total = a;}

~member1(){}

void pay()

{

cout << "每月给员工 1 的薪水为：" << \_total << "元" << endl;

}

private:

double \_total;

};

class member2:public salary

{

public:

member2(double a){\_total = a;}

~member2(){}

void pay()

{

cout << "每月给员工 2 的薪水为：" << \_total << "元" << endl;

}

private:

double \_total;

};

int main()

{

member1 m1(4000.0);

member2 m2(10000.0);

salary \*sa1 = &m1;

salary \*sa2 = &m2;

sa1->pay();

sa2->pay();

return 0;

}

//115 家族性格遗传（纯虚函数）

#include <iostream>

using namespace std;

class xingge

{

public:

xingge(){}

virtual void talk() = 0;

virtual void kind\_hearted() = 0;

private:

};

class child1:public xingge

{

public:

child1(){}

void talk(){cout << "沉默寡言" << endl;}

void kind\_hearted()

{

cout << "热心肠" <<endl;

}

};

class child2:public xingge

{

public:

child2(){}

void talk(){cout << "碎碎念" << endl;}

void kind\_hearted()

{

cout << "热心肠" <<endl;

}

};

int main()

{

xingge \*xg;

child1 c1;

child2 c2;

xg = &c1;

cout << "child1 的性格" << endl;

xg->talk();

xg->kind\_hearted();

xg = &c2;

cout << "child2 的性格" << endl;

xg->talk();

xg->kind\_hearted();

return 0;

}