

Вопрос 1 2 вида наследования:

1. is a

```
class Parent {
public:
int x;
protected:
int y;
private:
int z;
};

class Kid: public Parent {
public:
int func() {
    return x;
    return y;
    return z; // is inaccessible
}
};

int main() {
    Kid kid;
    kid.x;
    kid.y; // is inaccessible
    kid.z; // is inaccessible
}
```

x в Kid – **public**, y в Kid – **protected**, z недоступен для Kid

2. is implemented with

```
class Parent {
public:
int x;
protected:
int y;
private:
int z;
};

class Kid: private Parent { //now private
public:
int func() {
    return x;
    return y;
    return z; // is inaccessible
}
};

int main() {
    Kid kid;
    kid.x; // now is inaccessible
    kid.y; // is inaccessible
    kid.z; // is inaccessible
}
```

x в Kid – **private**, y в Kid – **private**, z недоступен для Kid

Вопрос 2 Тогда первым аргументом в `std::ostream& operator<<()` будет автоматически передаваться этот класс, а должен быть поток вывода.

Вопрос 3 Чистая виртуальная функция объявляется так: `virtual Type pure() = 0;`. Она не определена, её обязаны определить дети. Абстрактный базовый класс – это класс, имеющий хотя бы одну виртуальную функцию. Его экземпляр нельзя создать.