Абстрактные классы

C++

Что это

Иногда возникает необходимость определить класс, который не предполагает создания конкретных объектов. Например, класс фигуры. В реальности есть конкретные фигуры: квадрат, прямоугольник, треугольник, круг и так далее. Однако абстрактной фигуры самой по себе не существует. В то же время может потребоваться определить для всех фигур какой-то общий класс, который будет содержать общую для всех функциональность. И для описания подобных сущностей используются абстрактные классы.

Определение

Абстрактные классы - это классы, которые содержат или наследуют без переопределения хотя бы одну чистую виртуальную функцию. Абстрактный класс определяет интерфейс для переопределения производными классами.

Чистые виртуальные функции

Это функции, которые не имеют определения. Цель подобных функций - просто определить функционал без реализации, а реализацию определят производные классы. Чтобы определить виртуальную функцию как чистую, ее объявление завершается значением "=0".

```
class Shape
{
public:
    virtual double getSquare() const = 0;      // площадь фигуры
    virtual double getPerimeter() const = 0;   // периметр фигуры
};
```

Определение

Класс Shape является абстрактным, потому что он содержит как минимум одну чистую виртуальную функцию. А в данном случае даже две таких функции - для вычисления площади и периметра фигуры. И ни одна из функций не имеет никакой реализации.

При этом мы не можем создать объект абстрактного класса:

Применение

Добавим класс Прямоугольник, наследник класса Фигура. Переопределим в нем функции из родительского класса

```
class Rectangle : public Shape  // класс прямоугольника
{
public:
    Rectangle(double w, double h) : width(w), height(h)
    { }
    double getSquare() const override
    {
        return width * height;
    }
    double getPerimeter() const override
    {
        return width * 2 + height * 2;
    }
private:
    double width;    // ширина
    double height;   // высота
};
```

Применение

При создании классов-наследников все они должны либо определить для чистых виртуальных функций конкретную реализацию, либо повторить объявление чистой виртуальной функции. Во втором случае производные классы также будут абстрактными.