Alexander Velikiy
V.A. Savel’ev, research supervisor
V.V. Tihonenko, language adviser
SHEI “National Mining University”, Dnipropetrovsk

## Using Multiple Inheritance in Programming

Before we talk about multiple inheritance, let's find out what is inheritance. As the name inheritance suggests an object is able to inherit characteristics from another object. In more concrete terms, an object is able to pass on its state and behaviors to its children. Multiple inheritance, when speaking of object-oriented programming (OOP) languages, is a term that refers to a class that inherits functionality from more than one parent class. The term cannot be fully grasped without understanding what a class is. A very brief definition of a class is the following one: a class is a set of programming instructions or code in an OOP language that describes the essence of an entity as well. Classes define all of the properties of an entity and contain all of the methods necessary for manipulating those properties.

Depending on the program that is being coded, there might be a need to write many classes that have things in common but need to remain distinct entities. This is where multiple inheritance comes into play. For example, technological advances allowing producing flying submarine or diving airplane would produce a hybrid product. That hybrid would have some properties of airplane has and some of the properties of a submarine, but it would be a unique product or entity. Multiple inheritance would easily allow the hybrid to inherit functionality from an airplane class and functionality from a submarine class at the same time without copying code. Another way of explaining multiple inheritance is to say that it makes possible more than one class to act as the base for other classes. There are some advantages in using multiple inheritance instead of simple copying the code from one or more classes to another. Bugs in programs often take time to be discovered, but when they are discovered, they need to be corrected as soon as possible. If code containing bugs is copied to several classes, programmers must make the same efforts to every copy of the code. When multiple inheritance is employed, however, he or she would have to work with the code of only the original class from which all of the subclasses inherit functionality.

Not all programming languages support multiple inheritance, even if they support object orientation. Some versions of some languages might be claimed to support it when in reality they support inheritance from only two classes. The classes from which other classes inherit functionality are often referred to as "parent" classes or "super classes," and the classes that inherit are called "child" classes or "subclasses." Programmers who are interested in working with classes in this way are strongly recommended to start from getting thorough knowledge of the basics of classes before attempting to take advantage of the power of multiple inheritance, because mistakes or poor coding practices in the parent class are passed down to the child class.