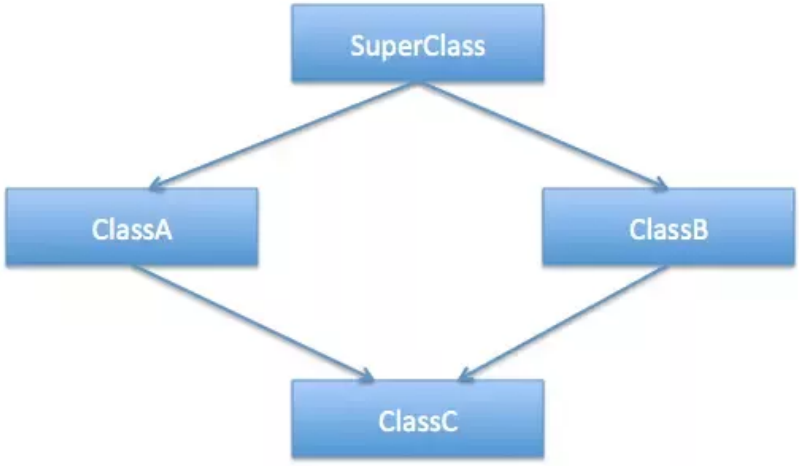
Seminar 13
Multiple and virtual inheritance in short

1. Multiple inheritance.

- In C++ classes can inherit as many other classes as needed.
- The inherited classes' constructors are called in the order in which they are inherited.
- The inherited classes' destructors are called in the exact opposite order.
- If needed members with the same names in different inherited classes can be distinguished by explicitly stating the class' name followed by "**::**".
  Example: `... Derived2::fun(); ...`

2. The diamond problem.

- In certain cases a duplication of a base class' memory can occur inside a derived class, for example in the classic infamous **diamond problem**.



In this case **ClassC** will have **SuperClass**' members duplicated, and thus the compiler won't know which copy to use.

- In C++ this problem can be solved by inheriting the base class **virtually**.

3. Virtual inheritance.

- Adding the **virtual** keyword before an inheritance will mark it as virtual and the derived classes will be "linked" to only **one instance** of the base class.
  No duplication of memory will occur, but the classes' memory will increase.
  This depends on the compiler, but usually it's implemented using virtual pointers and virtual tables.
  In the diamond problem ClassA and ClassB can virtually inherit SuperClass to solve the duplication of memory.
  Example:
  ```
  class ClassA : virtual public SuperClass { ... }
  class ClassB : virtual public SuperClass { ... }
  ```