

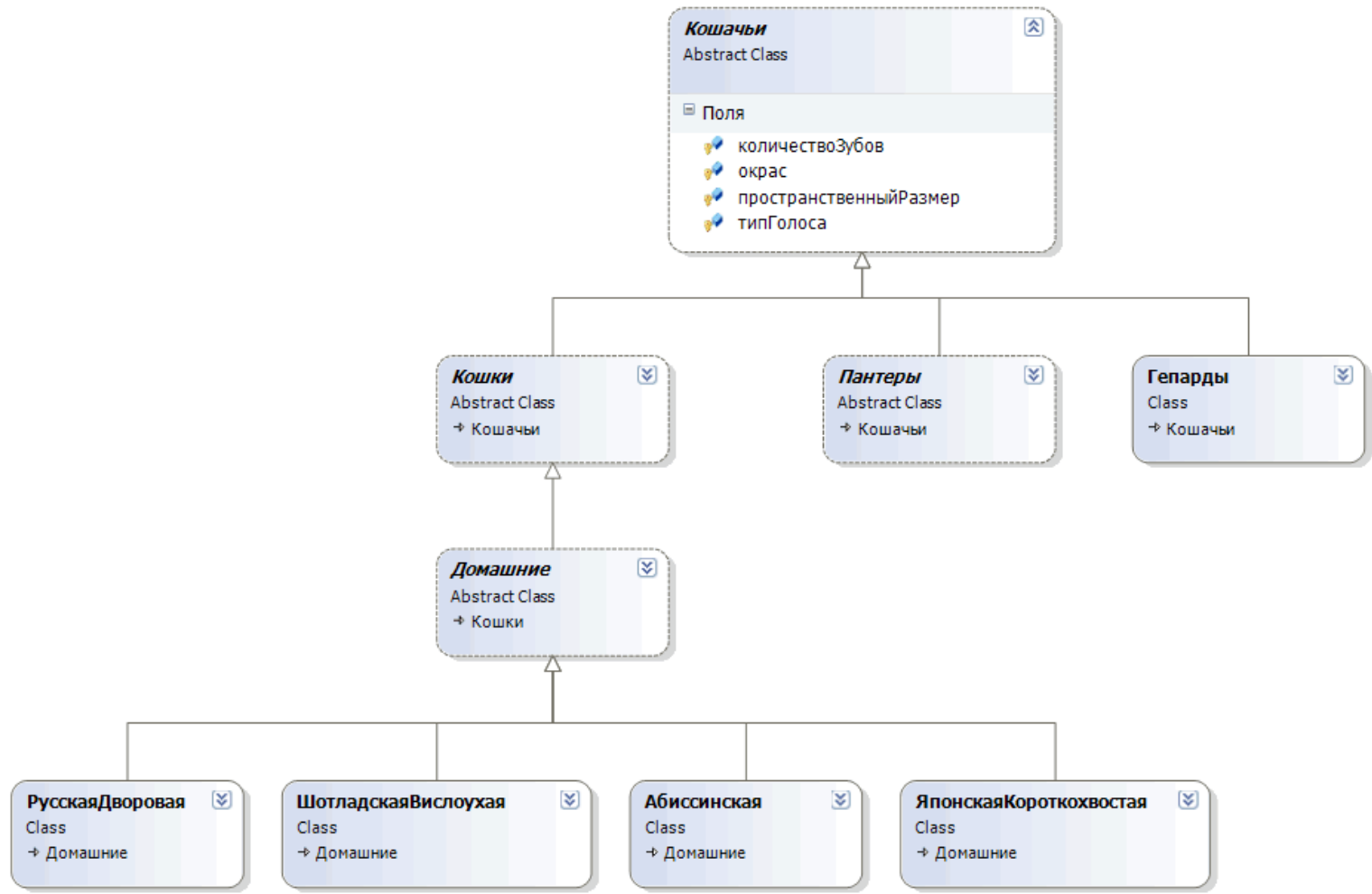




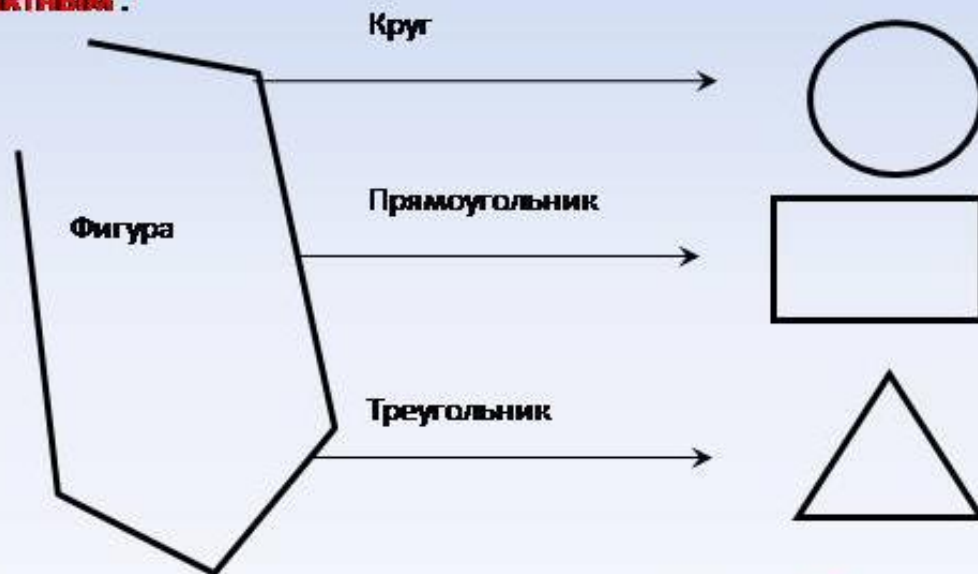
Абстрактные классы в C++

ТЕМА 2.10

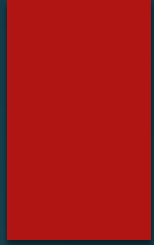
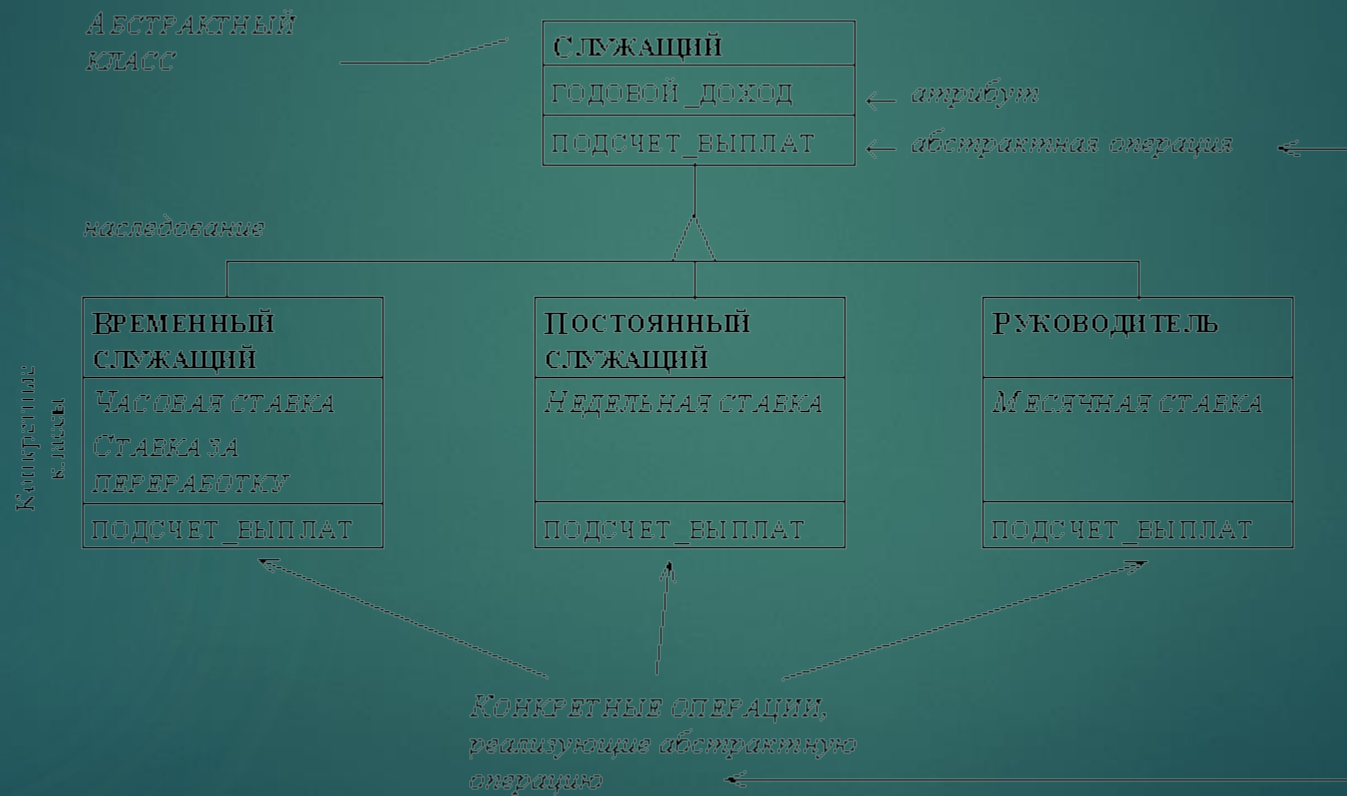
- 
- 
- ▶ **Абстрактный класс** в объектно-ориентированном программировании — базовый класс, который не предполагает создания экземпляров. Абстрактные классы реализуют на практике один из принципов ООП — полиморфизм. Абстрактный класс может содержать (и не содержать) абстрактные методы и свойства. Абстрактный метод не реализуется для класса, в котором описан, однако должен быть реализован для его неабстрактных потомков.






Суперкласс, который объявляет структуру определенной абстракции **без** предоставления полной **реализации** хотя бы одного **метода** называется - **абстрактным**.




Любой **класс**, который содержит один или более **абстрактных методов**, должен быть также объявлен как **абстрактный**.




- 
- 
- ▶ На языке программирования C++ абстрактный класс объявляется включением хотя бы одной чистой виртуальной функции, типа `virtual _сигнатура_функции_=0;`, которая, как и другие, может быть заменена.

- 
- ▶ Важной особенностью абстрактных классов является то, что не существует ни одного объекта данного класса. Вместо этого абстрактный класс служит в качестве базового для других производных классов. Причина, по которой абстрактный класс не может быть использован для объявления объекта, заключается в том, что одна или несколько его функций-членов не имеют определения. Тем не менее, даже если базовый класс является абстрактным, все равно можно объявлять указатели или ссылки на него, с помощью которых затем поддерживается полиморфизм времени исполнения.

- ▶ `Abstract abstract;`// создание ссылки на абстрактный класс
- ▶ `Impl impl = new Impl();`//создание объекта класса-наследника
//абстрактного класса
- ▶ `abstract = impl;`
- ▶ `abstract.doSmth;`//запускаем на исполнение метод, описанный
//в `Abstract` и реализованный в `Impl`

- 
- ▶ Абстрактные классы активно используются для создания т.н. пользовательского интерфейса. Ссылаясь через абстрактный класс можно “прятать” реализацию функции, предоставляя доступ к функционалу. То есть, наследование от абстрактных классов также помогает в осуществлении важной парадигмы объектно-ориентированного программирования – инкапсуляции.

- 
- ▶ Абстрактные классы являются важной частью современного объектно-ориентированного программирования, особенно касательно создания API. В более современных языках, как C# и Java, идея абстрактных классов как интерфейсов получила продолжение в так называемых интерфейсах, классах, только объявляющих методы, но не реализующих их.