1. Паттерны разделяют на порождающие, поведенческие и структурные
2. Не понял вопроса)) Ну можно в трейт вынести основные методы синглтона, необходимые для его создания, и переиспользовать трейт в других синглтонах.
3. Фабричный метод реализуется через общий интерфейс создаваемых объектов, и «создателя» с методом (абстрактный класс), который будет создавать определенные объекты (через Фабричный метод), реализующие общий интерфейс, в зависимости от конкретного создателя. И уже для создания конкретного целевого подкласса интерфейса используется метод-создатель подкласса-создателя. Паттерн Фабрика позволяет похожим образом создавать не один объект, а семейство связанных объектов.
4. Двусвязные списки spl – списки, в которых элементы связаны с соседями в обоих направлениях. Хороши для стеков/очередей, т.к. очень «дешево» позволяют итерировать, добавлять, удалять элементы.
   Массивы spl – отличаются от обычных массивов меньшим использованием памяти, но не столь удобны в использовании.
   Кучи spl – древовидные структуры. Каждый узел больше или равен своим потомкам.
5.
```
interface MyInt {
    public function funcI();
    private function funcP(); //методы в интерфейсе должны быть только public
}

class A {

    protected prop1; //переменная должна начинаться с $
    private prop2; //переменная должна начинаться с $

    function funcA(){
       return $this->prop2; //если исправить переменную в объявлении, ошибки нет

    }

}

class B extends A {

    function funcB(){

       return $this->prop1; //если исправить переменную в объявлении, ошибки нет
    }

}

class C extends B implements MyInt { //не реализован метод интерфейса funcI

    function funcB(){
        return $this->prop1; //если исправить переменную в объявлении, ошибки нет
    }
    private function funcP(){
          return 123;
    }

}

$b = new B();

$b->funcA();

$c = new C();

$c->funcI(); //данная фукнция не реализована
```