**Aluno: Erivaldo Barros**

Pesquise sobre padrões de projeto e escolha um para apresentar e descrever o seu funcionamento. Além disso, explique quais as vantagens e desvantagens comparados a outros e mostre suas referências.

**Factory Method**

Serve para definir uma interface para criar um objeto, mas deixa as subclasses decidirem que classe instanciar, ou seja, ela irá permitir que uma classe delegue a responsabilidade de instanciação para as subclasses.

O *Factory Method* deve ser utilizado quando:

* Uma classe não pode antecipar a classe de objetos a ser criada;
* uma classe quer que suas subclasses especifiquem os objetos a serem criados;
* classes delegam responsabilidades para uma dentre outras várias subclasses auxiliares, e você quer localizar qual subclasse auxiliar é delegada.

**Participantes**

* **Product** (Document)
* Define a interface de objetos que o método fábrica cria.
* **ConcreteProduct (MyDocument)**
* Implementa a interface de Product.
* **Creator (Application)**
* Declara o método fábrica o qual retorna um objeto do tipo *Product*. *Creator* pode também definir uma implementação por omissão do método *factory* que retorna por omissão um objeto *ConcreteProduct*.
* Pode chamar o método *factory* para criar um objeto *Product*.
* **ConcreteCreator (MyApplication)**
* Redefine (*override*) o método fábrica para retornar uma instância de um *ConcreteProduct*.

Uma desvantagem em potencial dos métodos fábrica é que os clientes podem ter que fornecer subclasses da classe *Creator* somente para criar um objeto *ConcreteProduct* em particular. Usar subclasses é bom quando o cliente tem que fornecer subclasses a *Creator* de qualquer maneira, caso contrário, o cliente deve lidar com outro ponto de evolução.

**Exemplo do Factory Method com Creator**

// “Creator”

abstract class Document {

    //Fields

    protected ArrayList pages = new ArrayList();

    //Constructor

    public Document(){

        this.CreatePages();

    }

    //Properties

    public ArrayList Pages{

        get { return pages; }

    }

    //Factory Method

    abstract public void CreatePages();

}

**Padrões relacionados**

***Abstract Factory*** é frequentemente implementado utilizado o padrão *Factory Method*. O exemplo na reação de Motivação no padrão *Abstract Factory* também ilustra o padrão *Factory Method*.

*Factory Methods* são usualmente chamados dentro de ***Template Methods***.