**Design Patterns. Abstract Factory**

**Task:**

Розробити додаток, який представляє магазин комп'ютерної техніки. На думку замовника, однією з найбільш потребуючих функцій програми буде можливість швидкого створення конфігурації системного блоку.

Для спрощення задачі припустимо, що до складу конфігурації системного блоку входять:

* бокс (Box);
* процесор (Processor);
* системна плата (MainBoard);
* жорсткий диск (Hdd);
* оперативна пам'ять (Memory).

Для кожної з цих складових визначимо потрібно описати абстрактний клас. Конкретні моделі складових визначати шляхом успадкування від абстрактного базового класу.

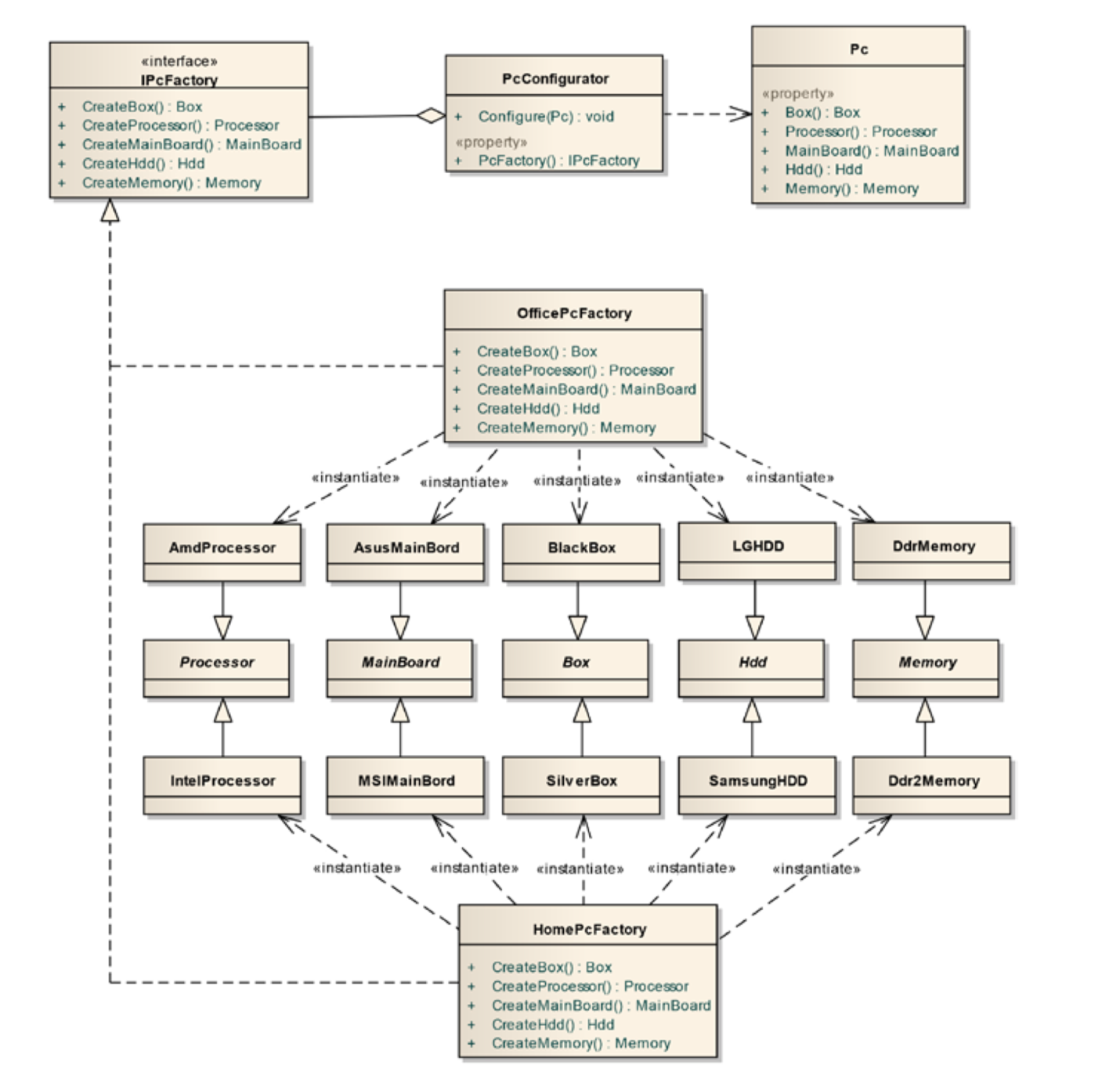
Клас “Pс”, який представляє конфігурацію системного блоку містить в собі всі компоненти та методи Get/Set (див. Class Diagram).

Припустимо, що наша програма повинна створювати конфігурації ПК двох типів: ігрові та офісні.

З логічної точки зору, важливо, що б всі складові заданої конфігурації системного блоку працювали злагоджено, тому відповідальність за їх створення треба покласти на один клас-фабрику. Ця фабрика повинна наслідувати абстрактний клас PсFactory (див. Class Diagram). Зауважу, що методи цього класу повертають посилання на класи абстрактних продуктів.

Для створення наборів компонентів даних конфігурацій визначимо класи конкретних фабрик OfficePсFactory і GamePсFactory (див. Class Diagram). У кожному з create-методів цих класів створюється об'єкт конкретного класу продукту, відповідного типу конфігурації.

Визначимо клас PсConfigurator, що відповідає за конфігурування об'етка типу Pc обраним сімейством складових (див. Class Diagram). Клас PсConfigurator приймає екземпляр конкретної фабрики і за допомогою її методів створює складові персонального комп'ютера. При цьому слід зауважити, що PсConfigurator працює з абстрактим класом фабрики, тобто він нічого не знає про конкретні фабрикі конфігурацій і конкретних складових. В цьому і проявляється вся сила абстрактної фабрики - конкретну фабрику можна визначати на етапі виконання програми, і при цьому клієнтський код (в даному випадку PсConfigurator) не залежить від конкретних фабрик або конкретних продуктів.



*Class Diagram*