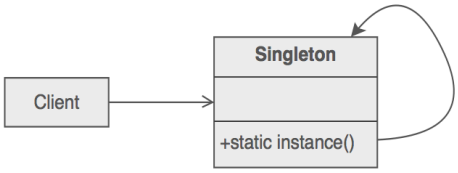
Creational design patterns:

כללי אצבע:

1. ניתן לשלב patterns לעבוד ביחד. יש מקרים בהם Prototype or Abstract Factory מתאים לבד. לפעמים Abstract Factory יכול להחזיק אובייקטים מסוג Prototype אשר אותם הוא משכפל ומחזיר למשתמש.
2.

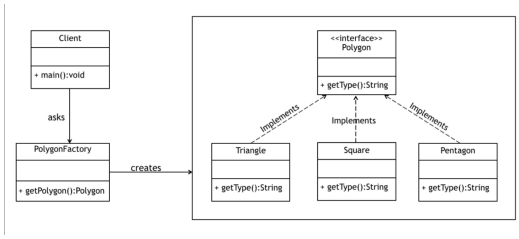
Singleton:



שימושי כאשר נרצה להגדיר אובייקט עם אינסטנס אחד בלבד בכל הפרוייקט.
ישנם מספר מימושים של singleton, במימושים יש לסים לב לדברין הבאים:

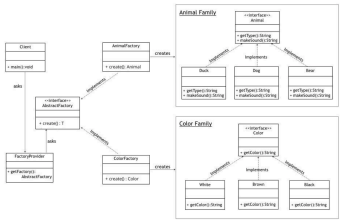
- איתחול האובייקט בצורה eager vs lazy.
- איתחול האובייקט בצורה Thread safe.
- לא לאפשר ליצור אינסטנס נוסף של האובייקט על ידי reflection.
- לא לאפשר ליצור אינסטנס נוסף של האובייקט על ידי serialization.

Factory:



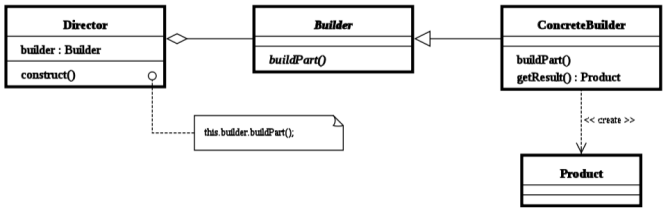
איתחול של אובייקטים באופן מפורש הופך את השינוי שלהם לקשה יותר, ועושה coupeling גבוה בין האובייקטים.
ב factory אופן האתחול של האובייקט מוסתר מהמשתמש.
מימוש: יוצרים ממשק שהאובייקטים שנרצה ליצור יממשו אותו. ה Factory יכיל Map של האובייקטים, וכאשר נרצה ליצור אובייקט מסוג מסויים נעזר ב Map.

Abstract Factory:



זה Factory of factories.

:Builder



תבנית ליצירת אובייקט שמכיל שדות רבים שחלקם שדות חובה וחלקם שדות רשות.
במקרה כזה המימוש הנאיבי של יצירת האובייקט יהיה על ידי שימוש בקומפוזיציה של בנאים, כלומר לעשות בנאי לכל אחד מהמקרים. או להשתמש ב Setters.
הפתרון לכך הוא יצירת אובייקט ריק, והוספת השדות החשבוים לאותו אובייקט.
מימוש: הבנאי של האובייקט יכיל את השדות שהן חובה, ושאר השדות יאותחלו באופן של קומפוזיציה, עד שהאובייקט יבנה על ידי קריאה לפונקציה build()
פונקציות הקומפוזיה יחזירו בסוף הפוקנציה את this, בכל הן יאפשרו שירשור של פונקציות בצד של המשתמש.
ספרייה: ב lombok, ניתן להגדיר builder על מחלקה על ידי האנוטציה builder@.