1. Cung cấp cả 2 loại class và object pattern: **Adapter**
2. Trong các mẫu thiết kế **class cha** đóng vai trò: **như** **1 interface**
3. Tạo ra một ứng **dụng multi-flatform**: **kết hợp adater và abstract factory**
4. Tăng cường **trách nhiệm**: sử dụng tính thừa kế - **proxy, decorator, composite**
5. Muốn **tương tác với một họ** trong nhiều họ đối tượng: **Abstract factory**
6. Chuyển nhiệm vụ cho một class con**: factory method**
7. Mối quan hệ phổ biến nhất trong các mẫu thiết kế: **Thừa kế**
8. Sinh ra các đối tượng và **ít sự khác biệt**: **Prototype**
9. Xác định động các đối tượng xử lý request: **Chain of responsibility**
10. Tập trung các hành vi giống nhau, tránh trùng lặp: Teamplate
11. Mẫu thiết kế có **lược đồ class giống nhau: Stragety và state**
12. Tránh các **lệnh số học** có khả năng xảy ra nhiều trường hợp: **Strategy**
13. Đa số mẫu thiết kế thuộc **creational**: **Class pattern**
14. Muốn định nghĩa sự phụ thuộc **1-n,** có nghĩa khi 1 cái thay đổi => sự cảnh báo đến những cái còn lại: **Observer**
15. Đa số mẫu thiết kế **behavior**: Object **pattern**
16. Mẫu thiết kế **builder**: **class pattern**
17. Dễ dàng **cá nhân hóa** menu-bar: **command**