

## Ćwiczenia 5 i 6 - wzorce projektowe - cd.

E1. Przeanalizuj kod dostępny na stronie <https://github.com/Apress/design-patterns-c-sharp/blob/master/BuilderPattern/Program.cs>

W projekcie stwórz kolekcję dyrektorów i różne poćwicz różne możliwości tworzenia obiektów.

E2. Dla poniższej klasy zaimplementuj wzorec budowniczego:

```
class Burger
{
    private int mSize;
    private bool mCheese;
    private bool mPepperoni;
    private bool mLettuce;
    private bool mTomato;
}
```

E3. Stwórz przykład prezentujący różnice między `string` a `StringBuilder`.

E4. Wykorzystaj wzorec budowniczego by stworzyć klasę do testowania obiektów. Wskazówka: <https://medium.com/@arleypadua/builder-pattern-applied-to-testing-60e009c427c6>

E5. Wykonaj co najmniej dwa projekty ilustrujące wzorec Fabryki Abstrakcyjnej:

Przydatne strony:

- <https://github.com/VanHakobyan/DesignPatterns/tree/master/Abstract%20Factory>
- <https://github.com/suyashsatav/Design-Patterns-csharp/tree/master/DP.AbstractFactoryPattern>
- <https://github.com/verloka/DesignPatterns/tree/master/src/CreationalPatterns/AbstractFactory/AbstractFactoryCSharp>
- <https://github.com/Fork501/Design-Patterns-Abstract-Factory/tree/master/C%23>

E6. Wykonaj co najmniej dwa projekty ilustrujące wzorec Metody Wytwórczej:

Przydatne strony:

- <https://github.com/xsoheilalizadeh/DesignPatterns/tree/master/Creational/FactoryMethod>
- <https://github.com/verloka/DesignPatterns/tree/master/src/CreationalPatterns/FactoryMethod/FactoryMethodCSharp>
- <https://github.com/alexandra-valkova/DesignPatterns/blob/master/CreationalDesignPatterns/AbstractFactory/Program.cs>

E7. Wykonaj co najmniej jeden projekt ilustrujący wzorec Adapter:

- <https://github.com/anupavanm/csharp-design-patterns-for-humans#-adapter>
- <https://github.com/verloka/DesignPatterns/tree/master/src/StructuralPatterns/Adapter/AdapterCSharp>
- <https://github.com/suyashsatav/Design-Patterns-csharp/tree/master/DP.AdapterPattern>
- <https://github.com/hamzaak/designpatterns/tree/master/adapter>
- <https://github.com/alexandra-valkova/DesignPatterns/tree/master/StructuralDesignPatterns/Adapter>

E8. Wykonaj co najmniej jeden projekt ilustrujący wzorec Dekorator:

- <https://github.com/anupavanm/csharp-design-patterns-for-humans#-decorator>

- <https://github.com/VanHakobyan/DesignPatterns/tree/master/Decorator>
- <https://github.com/verloka/DesignPatterns/tree/master/src/StructuralPatterns/Decorator/DecoratorCSharp>

E9. Wykonaj co najmniej jeden projekt ilustrujący wzorzec Fasada:

- <https://github.com/anupavanm/csharp-design-patterns-for-humans#-facade>
- <https://github.com/VanHakobyan/DesignPatterns/tree/master/Facade>
- <https://github.com/anupavanm/csharp-design-patterns-for-humans-examples/blob/master/FacadePattern/Program.cs>
- <https://github.com/hamzaak/designpatterns/tree/master/facade>
- <https://github.com/alexandra-valkova/DesignPatterns/tree/master/StructuralDesignPatterns/Facade>
- <https://github.com/verloka/DesignPatterns/tree/master/src/StructuralPatterns/Facade/FacadeCSharp>

E10. Wykonaj co najmniej jeden projekt ilustrujący wzorzec Kompozyt:

- <https://github.com/verloka/DesignPatterns/tree/master/src/StructuralPatterns/Composite/CompositeCSharp>
- <https://github.com/anupavanm/csharp-design-patterns-for-humans#-composite>