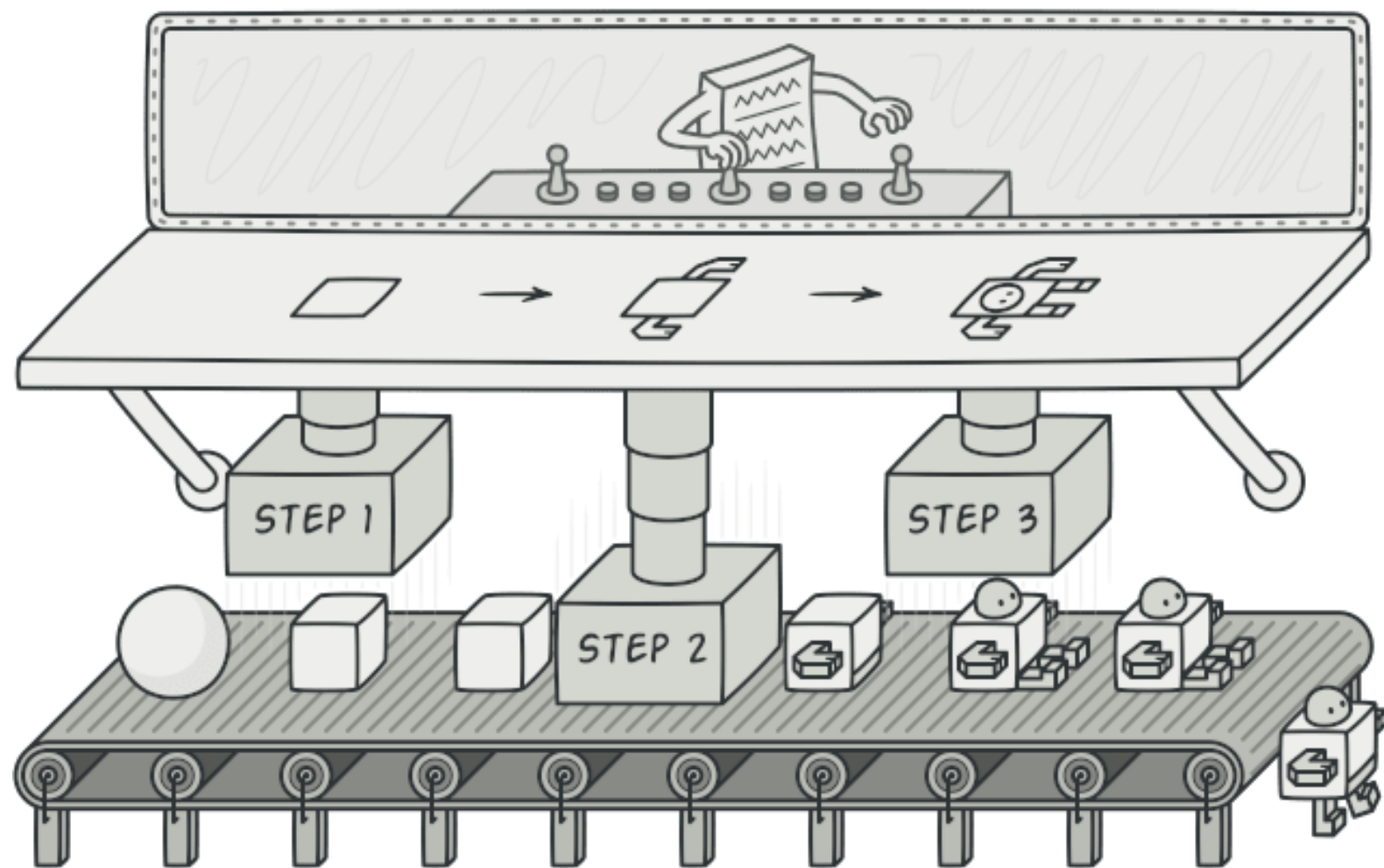




BUILDER

Alon jala tirta segara

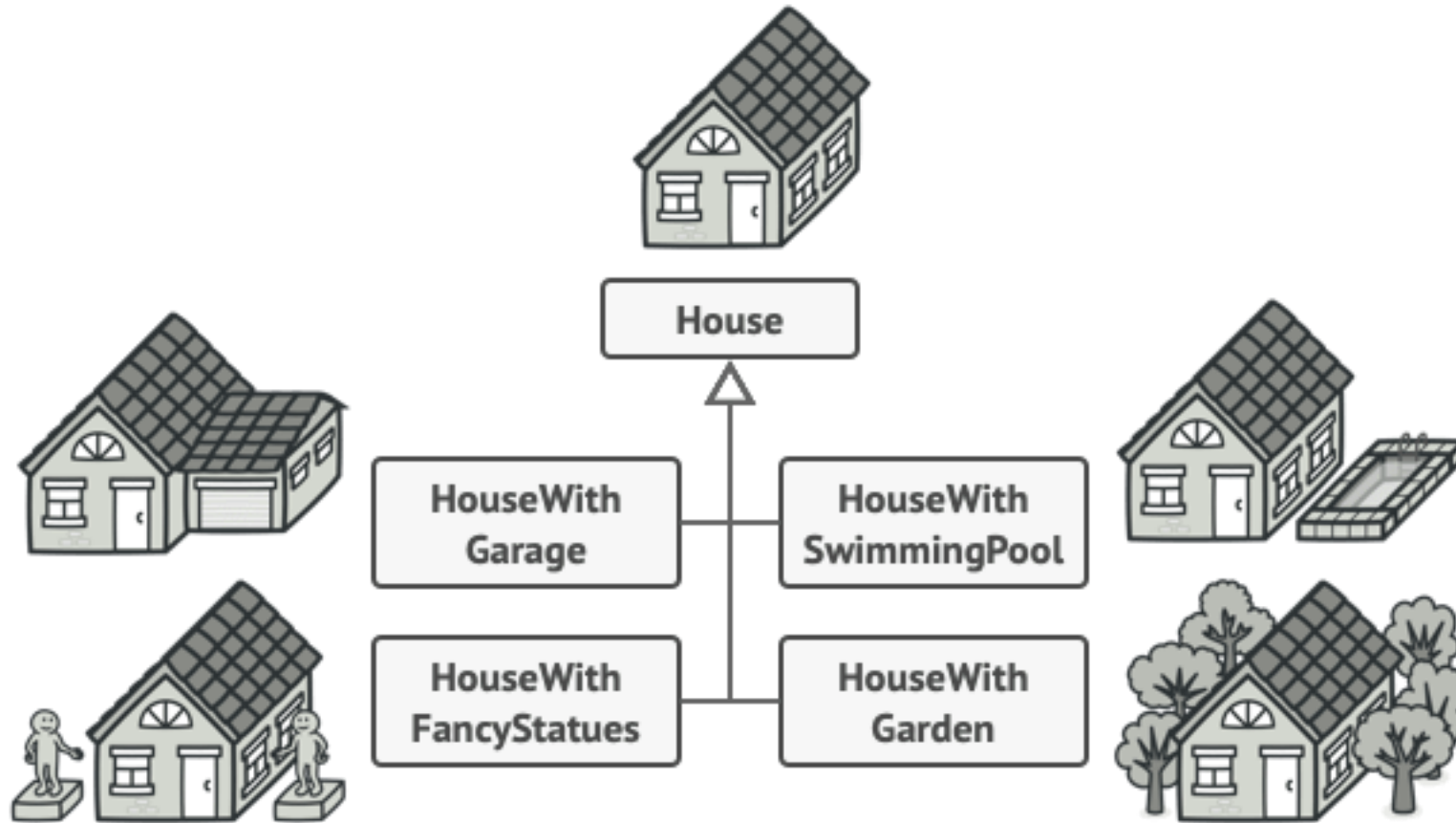


DEFINISI FORMAL

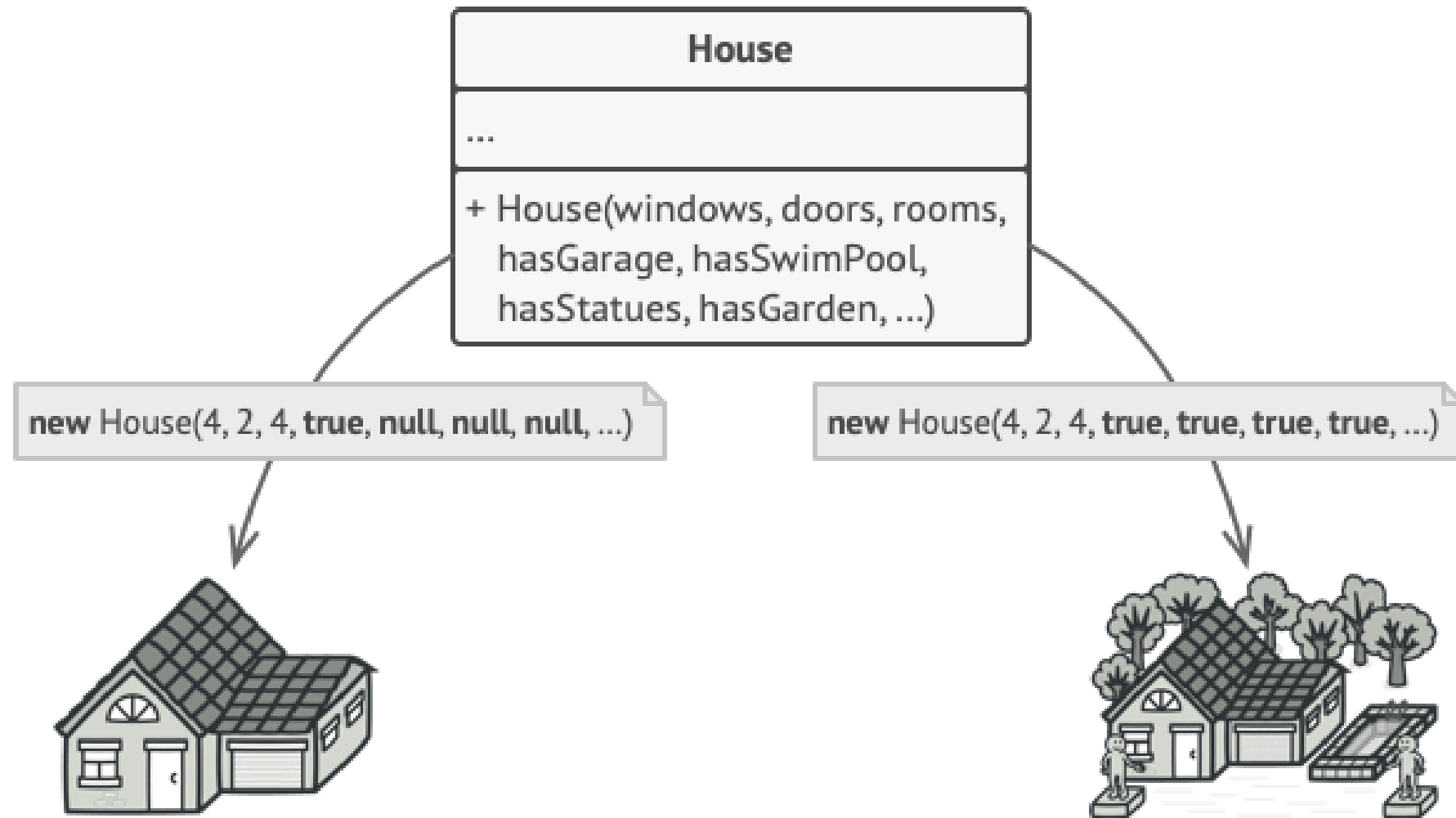
“

Builder adalah design pattern creational yang dikhususkan untuk membuat sebuah object yang kompleks tahap demi tahap. Pattern ini membuat kamu dapat menghasilkan berbagai tipe dan representasi dari sebuah object hanya dengan menggunakan basis kode yang sama.”

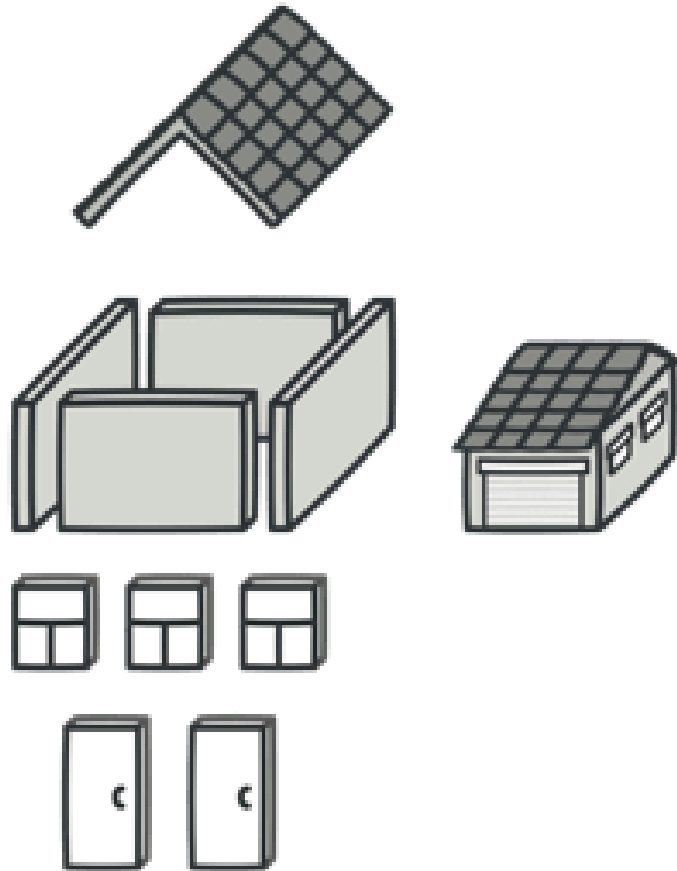
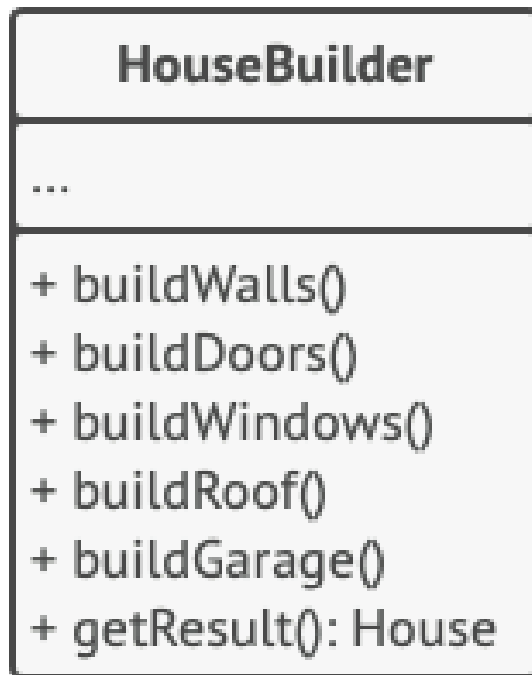
STUDI KASUS NYA



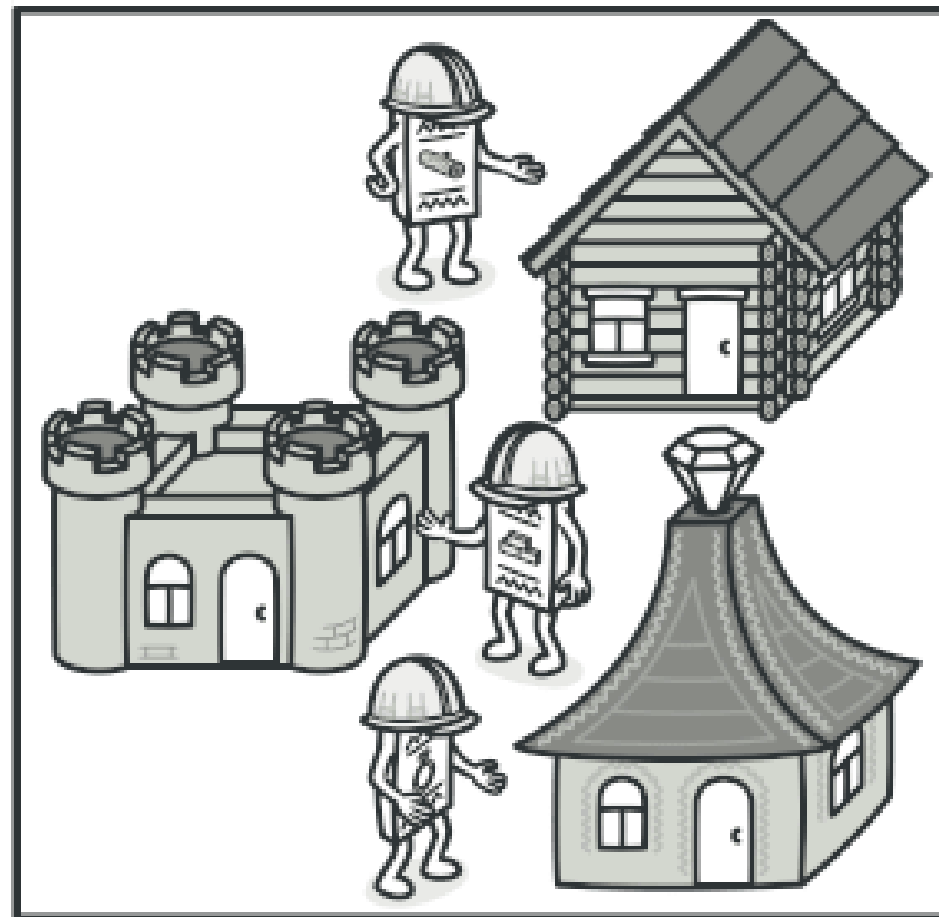
STUDI KASUS



SOLUSI



SOLUSI



ALGORITMA

Pastiin kamu udah mendefinisikan semua kebutuhan yang kamu butuhkan untuk membuat sebuah representasi objek.

Deklarasikan fungsi-fungsi dari tahap pembuatan di interface kamu, misalnya setRoof, setDoor (Hampir mirip dengan konsep OOP enkapsulasi).

Buat sebuah Class yang akan menjadi Objek pengenalan pembuatan produk itu sendiri.

Kode utama kamu akan memanggil objek di Nomor 3, dan memberitahunya untuk bekerja membuatkan produkmu, bisa dengan step by step, atau satu fungsi dasar untuk membentuk semuanya sekaligus

Kembalian dari produk akan kamu dapatkan dengan memanggil getResult().

PROJECT.PY

```
class Project:
    def __init__(self):
        self.name = None
        self.description = None
        self.framework = None
        self.lang = None
        self.pattern = None

    def getProject(self):
        str = "New Project wants to be started! \n"

        str += f"Project Name: {self.name}\n"
        str += f"Project Description: {self.description}\n"
        str += f"Project Framework: {self.framework}\n"
        str += f"Project Programming Language: {self.lang}\n"
        str += f"Project Development Pattern: {self.pattern}\n"

        return str
```

PROJECTBUILDER.PY,

```
from abc import ABC, abstractstaticmethod

class ProjectBuilder(ABC):

    @abstractstaticmethod
    def setName(self, name):
        pass

    @abstractstaticmethod
    def setDescription(self, description):
        pass

    @abstractstaticmethod
    def setFramework(self, framework):
        pass

    @abstractstaticmethod
    def setLanguage(self, language):
        pass

    @abstractstaticmethod
    def setPattern(self, pattern):
        pass
```

ECOMMERCEPROJECTBUILDER.PY,

```
from ProjectBuilder import ProjectBuilder
from Project import Project

class ECommerceProjectBulder(ProjectBuilder):
    def __init__(self):
        self.project = Project()

    def setName(self):
        self.project.name = "E-Commerce Web App"
        return self

    def setDescription(self):
        self.project.description = "E-commerce web app with cool features!"
        return self

    def setFramework(self, framework):
        self.project.framework = framework
        return self

    def setLanguage(self, language):
        self.project.lang = language
        return self

    def setPattern(self, pattern):
        self.project.pattern = pattern
        return self

    def buildProject(self):
        self.setName()
        self.setDescription()

    def getProject(self):
        return self.project.getProject()
```

BLOGPROJECTBUILDER.PY.

```
from ProjectBuilder import ProjectBuilder
from Project import Project

class BlogProjectBuilder(ProjectBuilder):
    def __init__(self):
        self.project = Project()

    def setName(self):
        self.project.name = "Blog Web App"
        return self

    def setDescription(self):
        self.project.description = "Blog web app with cool SEO features!"
        return self

    def setFramework(self, framework):
        self.project.framework = framework
        return self

    def setLanguage(self, language):
        self.project.lang = language
        return self

    def setPattern(self, pattern):
        self.project.pattern = pattern
        return self

    def buildProject(self):
        self.setName()
        self.setDescription()

    def getProject(self):
        return self.project.getProject()
```

MAIN.PY

```
from BlogProjectBuilder import BlogProjectBulder
from ECommerceProjectBuilder import ECommerceProjectBulder

builder = ECommerceProjectBulder()
builder.setLanguage("PHP").setFramework("Laravel").setPattern("MVC")
builder.buildProject()
print(builder.getProject())

builder = BlogProjectBulder()
builder.setLanguage("Python").setFramework("Django").setPattern("MVT")
builder.buildProject()
print(builder.getProject())
```

HASIL



```
kiddy@elementary-os:~/code/python/learn$ python3 main.py
```

```
New Project wants to be started!
```

```
Project Name: E-Commerce Web App
```

```
Project Description: E-commerce web app with cool features!
```

```
Project Framework: Laravel
```

```
Project Programming Language: PHP
```

```
Project Development Pattern: MVC
```

```
New Project wants to be started!
```

```
Project Name: Blog Web App
```

```
Project Description: Blog web app with cool SEO features!
```

```
Project Framework: Django
```

```
Project Programming Language: Python
```

```
Project Development Pattern: MVT
```