Object Oriented Programming with Python

Gramsci Hermozo

Session 16

Content

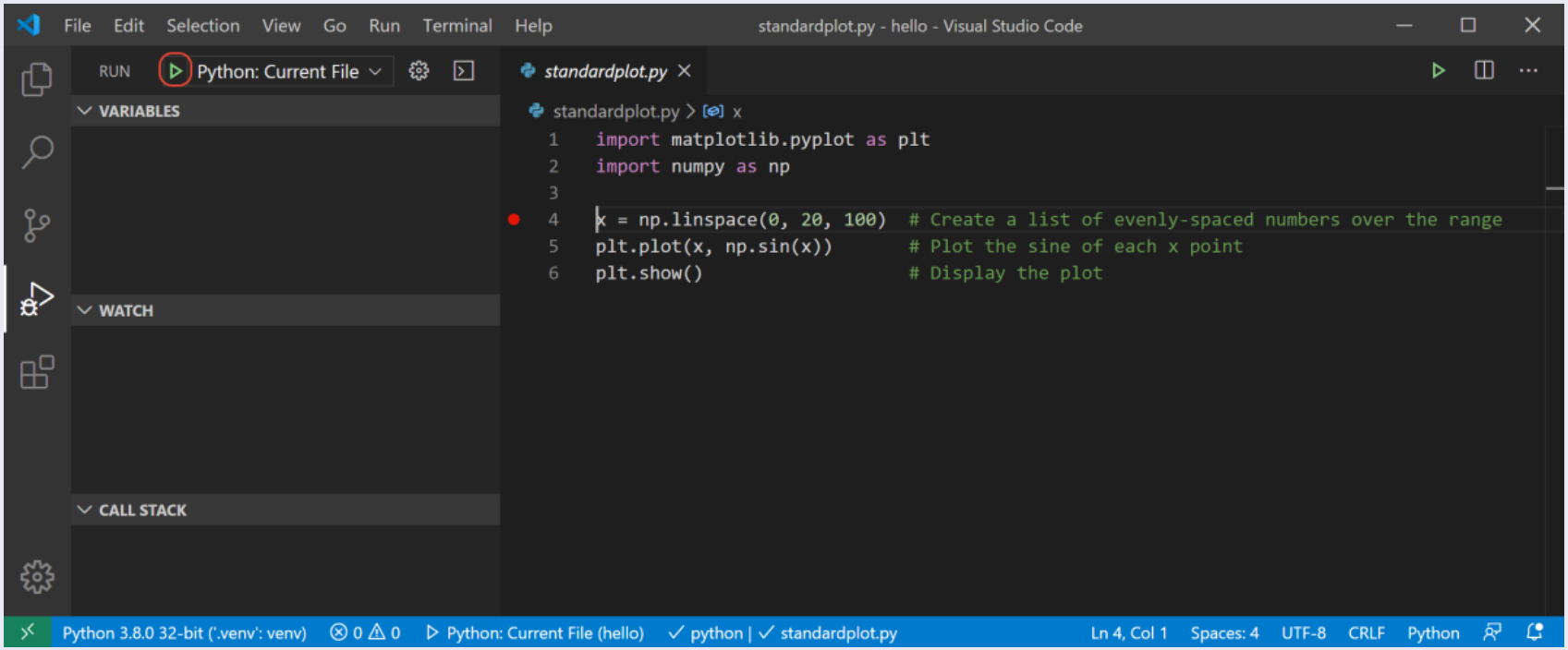
- Debugging code using IDE tool
- Design Patterns (Abstract Factory, Singleton)
- SOLID
- Added unittest where would be needed
- Add console UI
- Practice

Debugging code

Visual Studio Code

If the project allow you to use debugging you can use this option instead of prints or loggings

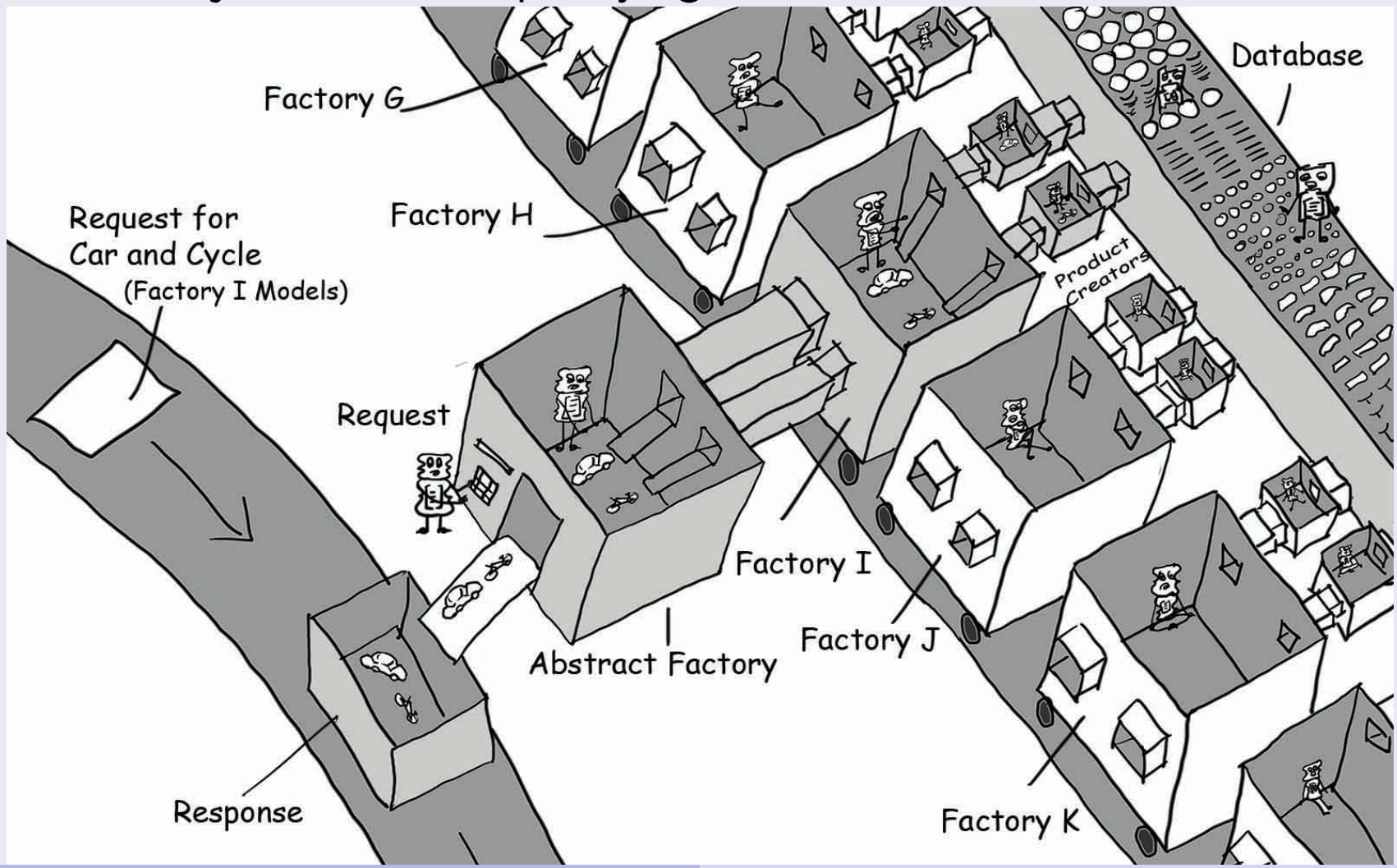
https://code.visualstudio.com/docs/python/Debugging



Design Patterns

Abstract Factory

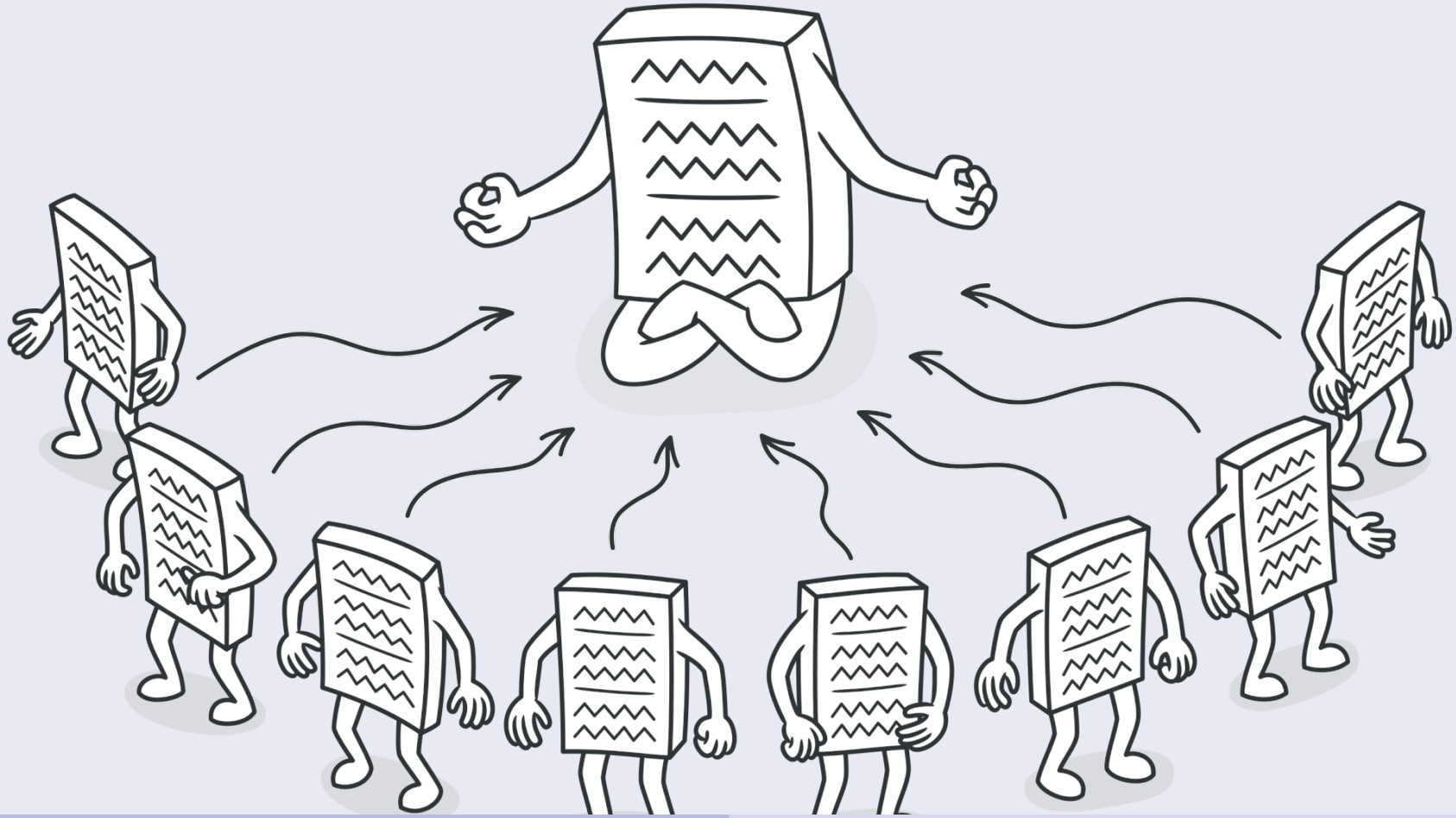
Is a creational design pattern that lets you produce families of related objects without spedifying their concrete classes.



Design Patterns

Singleton

Is a creational design pattern that lets you ensure that a class has only one instance, while providing a global access point to this instance.



SOLID

- S -> Single responsability
- O -> Open and Close (open for extension but closed for modification)
- L -> Liskov Substitution
- I -> Interface Segregation
- D -> Dependency Inversion

Practice

Lets practices

Un grupo de inversionistas nos contacto para poder implementar un sistema con las siguientes caracteristicas.

- Necesitan que un administrador pueda agregar/registrar personas.
- Necesitan poder diferenciar entre personas que trabajan para este grupo de personas y clientes.
- Dentro de la clasificacion personal que trabaja para este grupo tambien manejar un rango como ser: Gerente, cajero, asesor, administrador de sistemas, etc.
- Agregar unit test para la validacion de la creacion de personas dentro el sistema
- Crear el diagrama de clases
- Para el registro en esta primera version usar la consola.