test

March 28, 2022

[ ]:
```
* soft
    - changes
    - reasons
```

[ ]:
```
def table(table_for = 5, till = 10):
    for i in range(1, till + 1):
        print(f"{table_for} x {i} = {table_for*i}")
```

[ ]:
```
def f4(para, asjd):
    ...
    ...
```

[ ]:
```
f1,f2, ..fn
```

[ ]:
```
* new feature
```

[ ]:
```
f1 > f2 > f3 > f4 .. > fn
```

[ ]:
```
def f11():
    f4(parna)
```

[ ]:
```
* chnages -> reasons
* change -> dependent functions / classes change
```

[ ]:
```
# high cohesion, loose coupling
```

[ ]:
```
class F1:
    def __init__(self):
        self.int1 = 0
        self.int2 = 10

    def execute(self):
        for i in range(1, till + 1):
            print(f"{table_for} x {i} = {table_for*i}")
```

[ ]:
```
class IPen:
    def write(self):
        pass
class ICanvas:
    def draw(self):
        pass
class IFan:
    def fan():
        pass
class IFly:
    def aero(selff):
        pass
class LekhoFeko(IPen):
    def __init__(self):
        self.color = 'blue'
    def write(self):
        pass
class Paper(ICanvas, IFly):
    def __init__(self):
        pass
    def draw(self):
        self._pen.write()
    def set_pen(self, pen: IPen):
        self._pen = pen
    def set_page_nr(self, nr):
        self._pagr_nr = nr
class FlyingPaper(IFly):
    pass
```

[ ]:
```
p = Paper()

pen = LekhoFeko()

p.draw(pen)

paper > pen

pen > paper

pen.color = "red"
```

[ ]:
```
c1 > c2 > c3 > c4 > c5 > c6 > .. > cn
```

[ ]:
```
c1 < c2 < c3 < c4 < c5 < c6 < .. < cn
```

[ ]:
```
print("hello world")
```

[ ]:
```
# SOLID principles
```

[ ]:
```
class C3(c3):
    def execute(self):
        print("hi world")
```

[ ]:
```
c3 = C3()
c4.set_c3(c3)
c4.write()
```

[ ]:
```
# def denasdj():
#     pass
```

[ ]:
```
if a > 0:
    check()
else:
    check()
=2  #cylcomatic complexity
```

[ ]:
```
# * size coding ( 5 %)
# * re-use ( 90 %)
```

[ ]:
```
# * group coding : principle
# * others reuse
# * code -> code (AI level) # software 2.0
```