プログラミング技法 予習課題6

IKEDA Kaito

2021/5/1

make_dinner 関数は，素材 (material）を引数に取って，素材を cut (切って)，cook (煮たり焼いたりして)，serve (盛り付ける）の操作を順番に実行します． 素材が肉と魚では，それぞれの操作 (cut や cook）での作業内容が異なるとします．

以下， make_dinner1 make_dinner2 の 2 通りの実装を試みました．

末尾に示す 問題 1 から 問題 3 に解答しなさい．

1) 個々の操作を呼び出す側で switch する方式

```
def make_dinner1(material)
    case material.type
    when MEAT; make_meat_dinner(material)
    when FISH; make_fish_dinner(material)
    end
end

def make_meat_dinner(material)
    cut_meat(material)
    cook_meat(material)
    serve_meat(material)
end

def make_fish_dinner(material)
    cut_fish(material)
    cook_fish(material)
    serve_fish(material)
end
```

2) 呼ばれた個々の操作で switch する方式

```
def make_dinner2(material)
    cut(material)
    cook(material)
    serve(material)
end

def cut(material)
    case material.type
```

```
    when MEAT; cut_meat(material)
    when FISH; cut_fish(material)
    end
end

def cook(material)
    case material.type
    when MEAT; cook_meat(material)
    when FISH; cook_fish(material)
    end
end

def serve(material)
    case material.type
    when MEAT; serve_meat(material)
    when FISH; serve_fish(material)
    end
end
```

1 どちらの実装のほうがよい実装だと言えますか．拡張性，修正の容易さという観点から，理由とともに説明してください．

私は make_dinner1 の実装の方が良いと考えます．例えば，meat に 新しく bake という機能を持たせようとした際， meat という大きな枠（今回の場合 make_meat_dinner）に bake という機能を追加（オーバーライド）するだけで機能が簡単に追加削除できるからです．

2 make_dinner1, 2 それぞれについて，material に EGG が加わった場合の拡張を施したコードを示してください．

```
def make_dinner1(material)
    case material.type
    when MEAT; make_meat_dinner(material)
    when FISH; make_fish_dinner(material)
    when EGG; make_egg_dinner(material)
    end
end

def make_meat_dinner(material)
    cut_meat(material)
    cook_meat(material)
    serve_meat(material)
end

def make_fish_dinner(material)
    cut_fish(material)
```

```
    cook_fish(material)
    serve_fish(material)
end

def make_egg_dinner(material)
    cut_egg(material)
    cook_egg(material)
    serve_egg(material)
end

def make_dinner2(material)
    cut(material)
    cook(material)
    serve(material)
end

def cut(material)
    case material.type
    when MEAT; cut_meat(material)
    when FISH; cut_fish(material)
    when EGG; cut_egg(material)
    end
end

def cook(material)
    case material.type
    when MEAT; cook_meat(material)
    when FISH; cook_fish(material)
    when EGG; cook_egg(material)
    end
end

def serve(material)
    case material.type
    when MEAT; serve_meat(material)
    when FISH; serve_fish(material)
    when EGG; serve_egg(material)
    end
end
```

3 make_dinner1, 2 それぞれについて，cut を 2 回してから cook, serve するように調理方法を変更したコードを示してください.

```
def make_dinner1(material)
    case material.type
    when MEAT; make_meat_dinner(material)
    when FISH; make_fish_dinner(material)
```

```
    end
end

def make_meat_dinner(material)
    cut_meat(material)
    cut_meat(material)
    cook_meat(material)
    serve_meat(material)
end

def make_fish_dinner(material)
    cut_fish(material)
    cut_fish(material)
    cook_fish(material)
    serve_fish(material)
end

def make_dinner2(material)
    cut(material)
    cut(material)
    cook(material)
    serve(material)
end

def cut(material)
    case material.type
    when MEAT; cut_meat(material)
    when FISH; cut_fish(material)
    end
end

def cook(material)
    case material.type
    when MEAT; cook_meat(material)
    when FISH; cook_fish(material)
    end
end

def serve(material)
    case material.type
    when MEAT; serve_meat(material)
    when FISH; serve_fish(material)
    end
end
```