

192010

喜田圭伍

## 学習記録 学習メモ

オブジェクトはメソッドとプロパティが一緒になったもの

クラスはオブジェクトの設計図

インスタンスがオブジェクトの実体

裏で動いてるメソッドがコンストラクター

クラス定義内で明示的にそのクラスのインスタンスであると示す

良いプログラミングの条件ISO/IEC

使用性（使いやすさ）

時間効率性

throughput(job/per second)→システム目線

latency(second)→user目線

資源効率性（メモリやCPUをあまり喰わない）

オブジェクト間のやり取りでプログラムを作っているのがオブジェクト指向

カプセル化 窓口を一元化して内部状態を見えないようにする

private

setterとかgetterに値を加えることができる

クラス外からプロパティ値にアクセスするのは避けた方が良い

継承 再利用性を高めるためのあるクラスを拡張したクラスを定義するときを使う

## 学習記録 演習

5-1

```
class Student constructor (argname:String, argid:String, argpass:String, argbirthy:Int){
    var name : String = argname
    var id : String = argid
    var pass : String = argpass
    var birthy : Int = argbirthy
    init{
        if(argid.length != 6 || argbirthy >= 2000){
            println("Id should be 6 digit, birth year is impossible value")
        }
    }
    fun age(): Int{
        return 2018 - birthy
    }
}
```

```

fun main(args: Array<String>) {
    val student = Student("Krigo", "772835", "9383", 2000)
    println(student.name)
    println(student.age())
}

```

5-2

```

class Student constructor (argname:String, argid:String, argpass:String, argbirthy:Int){
    var name : String = argname
    var id : String = argid
    var birthy : Int = argbirthy
    get(){
        return field
    }

    private var pass : String = argpass

    init{
        if(argid.length != 6 || argbirthy >= 2000){
            println("Id should be 6 digit, birth year is impossible value")
        }
    }
    fun age(): Int{
        return 2018 - birthy
    }

    fun passchange(passcha : String){
        pass = passcha
    }

    fun passconfirm(passcon : String): Boolean{
        if (passcon == pass){
            return true
        }else {
            return false
        }
    }

}

fun main(args: Array<String>) {

```

```
        val student = Student("Krigo", "772835", "9383", 2000)
        println(student.name)
        println(student.age())
        student.passchange("293013")
        println(student.passconfirm("293013"))
    }
```

### 5-3

パスワードが外部から得られる設計は危険なためprivate

パスワードを変更したい場合もあるため

パスワードが合っているか確かめる工程が必須なため

学生オブジェクトがパスワード変更メソッドがあると学生がパスを変更できてしまうのでよくない

教員クラスを作成しパス変更メソッドを定義する設計が好ましい。