

ЛЕКЦІЯ №**10****ДАТА ПЕРЕГЛЯДУ:****03/06/2018****ПЕРЕВІРЯВ:****Горбушко К****СТУДЕНТ:****Юрій Жук****ГІЛКА:****lesson10****ОЦІНКА:****7 / 10**

ЗАВДАННЯ 1-2

Додавайте final до сутностей якщо вони не мають наслідуватися
В 10 лекції помилки ще не вчили, проте +
Використовуйте optionalUnwrap замість force Unwrap

```
expectedHiringDate!)! <= employee.birthDay) {
```

Якщо є константи - виносьте їх в окремі змінні, як наприклад

```
    } else if (calendar.date(byAdding: .year, value: -18, to:  
expectedHiringDate!)! <= employee.birthDay) {
```

«-18» - не зрозуміле число

CodeStyle (пусті рядки вкінці класу, пробіли і тп) - все має бути акурантним

Клас Worker у вас передається по посиланню - немає сенсу робити inout параметр - це лише робить просте складнішим (див додаток removedInout.gif). Теж саме правда і для інших випадків

Краще використати private (set) для всіх властивостей об'єктів і окремі методи на модифікацію значень властивостей якщо треба

```
var workers = [Worker]()
```

Використовувати глобальну змінну видимості файла не найкраща практика

```
fileprivate var multiplier = 0
extension Worker {
```

краще використати властивість в класі з модифікатором internal
Назви мають бути такими щоб передавали зміст повністю

```
enum Desc:Error {
    case employeeDidntBorn
    case employeeIsTooYoung
}
```

Написавши Desc замість Description що зекономлено? - нічого, що поганого - погіршується розуміння. Наприклад

```
enum EmployeeBirthDateCheck : Error {
    case invalidOrUnknown
    case underWorkPeriod
}
```

покращує ситуацію адже явно розміємо для чого це і на відміну від Description ми точно знаємо для якої сутності

Локальні змінні краще називати з маленької букви

```
let DataBase = DB()
```