Szoftver tesztelés

A pytest

(dokumentáció)

Tartalomjegyzék

[1. Bevezető 4](#_Toc98405483)

[2. Az adatról 5](#_Toc98405484)

[3. Tesztelési módszer 6](#_Toc98405485)

[4. Implementálás 7](#_Toc98405486)

[5. Összefoglalás 8](#_Toc98405487)

# Bevezető

--általánosan a tesztelésekről

# Működés

Ahhoz, hogy eljussak a teszteléshez előbb meg kellett ismerkednem a kódbázissal. Rendelkezésemre állt employee\_manager és egy relations\_manager forráskód. A következőkben ezeknek a funkcionalitásait fogom ismertetni.

A relations\_manager tartalmaz egy osztályt (RelationsManager) és ebben van néhány függvény, az init fügvényben található egy employee lista benne 6 darab employee-val (alkalmazottal). Ahhoz, hogy ez megvalósítható legyen, szükség van egy employee osztályra, ami tartalmazza egy alkalmazott adatait, és ezt az osztályt deklarálni kell a relations\_manager forráskód elejére. Ebben a függvényben még található egy struktúra ami reprezentálja az employee listában levő csapatokat. Minden csapatnak van egy vezetője (leader) és ebben az esetben 2-2 csapat tag. Ezen kívül még található a RelationsManager osztályban 4 függvény, amelyek a következők:

* is\_leader(): amely visszatéríti, hogy egy alkalmazott vezetője-e egy csapatnak vagy sem; egy bool típust térít vissza.
* get\_all\_employees(): visszatérít egy listát, ami az alkalmazottakat tartalmazza.
* get\_team\_members(): megnézi, ha egy alkalmazott vezető pozicióban van és ha igen akkor vissza adja a csapat tagjait.
* get\_employee\_name\_and\_birth(): bejárja az employee listát és vissza adja egy vezető employee nevét és születési dátumát.

Az employee\_manager forráskód tartalmaz egy EmployeeManager nevű osztályt és egy fő(main) függvényt. Az osztály megkapja az init függvényben, ami egy konstruktorként szerepel, paraméterként a RelationsManager osztályt. Ebben az osztályba további 2 függvény található:

* calculate\_salary(): kiszámolja egy adott alkalmayott fizetését, ha az alkalmazott egy vezető akkor a fizetéséhez hozzá lesz adva bónusz minden tag után aki a csapatában van. Miután kiszámolta a fizetést visszatéríti ennek az értékét.
* calculate\_salary\_and\_send\_email(): kiszámítja egy alkalmazott fizetését meghívva az előző függvényt és kiír a terminálra egy üzenetet.

A fő függvényben meg le van kérve az alkalmazottak listája és ki van iratva, meg van vizsgálva egy adott alkalmazott csapattagjait, majd ki van szmolva a fizetése úgy, hogy meg van hívva a calculate\_salary\_and\_send\_email() függvény.

Tehát a forrás kódok alkalmazottakról tárolnak el információkat és különböző műveleteket segítségével meg lehet vizsgálni ezeknek az információknak a helyességét.

# Metodológia

--- megfogalmazom ,hogy mi akarom hogy a kimenet legyen

# Eszközök

--- beszélni a módszerekről és kiemelni hogy én melyiket választottam, bemutatni azt(telepítés, működés)

# Tesztelés

---hogyan alkalmaztam a tesztelést, kódrészletek

# Következtetések

---mit tudtam elérni, mit vonok le az egészből(mi jó, mi nem)