# Bibliotekos aprašymas

Biblioteka sudaro 7 klases, main ir 2 interfaces:

Interfaces:

1. Utilization – duoda funkciją kuri apskaičiuoja ant kiek daug yra užpildyta Play Site procentais;
2. Watch – turi funkcijas duodančias gebėjimą apskaičiuoti kiek laiko vaikas yra play site.

Klasės:

Main – tuščia, nereikalinga bibliotekai;

1. PlaySite: abstrakcinė klasė, turi 90% funcijų, kurios yra reikalingos skirtingoms play sites. Implementuoja Utilization.
2. DoubleSwings, Carousel, Slide, BallPit – klasės extends PlaySite. Skirtos išskirstyti playsites į konkrečius objektus.
3. Kid – klasė, kuri turi vaiko duomenys. Implementuoja Watch.
4. Kid Manager – klasė, kuri turi static funkcijas ir ArrayList. Skirta sužinoti kiek viso yra vaikų visame playground ir sužinoti, kiek laiko vaikas yra play site.

Biblioteka duota funkcijas, kurias galima nauduoti:

1. Galima sukirti playsitus: DoubleSwings, Carousel, Slide, BallPit ir jiems priskirti vaikų limitą;
2. Sukurti vaiką ir priskirti jam duomenys;
3. Priskirti vaiką į paskirtą play site;
4. Pašalinti vaiką nuo playsite ir nuo eilės;
5. Sužinoti kiek vaikas skyrė laiko playsite;
6. Gauti playsito utilizaciją;
7. Gauti kiek visitorių(vaikų) yra playgrounde.

# Kai kurių funkcijų paaiškinimas

* Kontaineriai: playsites turi Array, kuris yra sukuriamas priskiriant jam limitą. Kiek limitas bus, tiek ir jo didis. Šis Array laiko vaikus playsite.
* ArrayDeque: kontroliuoja kur priskirti vaiką į Playsite Array. Jis Laiko pozicijas ir kada vaikas yra priskistas, arraydeque naikina poziciją. Kada vaikas yra pašalimas, pozicija nuo vaiko yra atlaisvinama.
* ArrayList: kontroliuoja eilę.

add(kid):

Kada prie pvz: double swings pridedama vaiką, iškart yra tikrinama, ar arraydeque turi poziciją. Jeigu taip, reiškiasi, kad yra laisvos vietos array playsite ir priskiriama. Funkcija gražina string su pranešimu kad vaikas yra priskirtas į playsite. Jeigu ne, yra bandoma priskirti į eilę. Yra galimybė kad vaikas nesutiks. Jeigu sutiko, priskiriama į eilę, rūšiuojama jeigu vaikas yra VIP ir gražina pranešimą kad vaikas sutiko pabūti eilėje. Jeigu nesutiko, tai tik yra gražinamas pranešimas, kad vaikas nesutiko.

sortVIPQueue():

Jeigu VIP vaikas yra priskirtas į eilę, ši funkcija bus pradėta. Iškart VIP yra priskirtas į eilės galą, pvz.: buvo VNVNNNNN, tai dabar yra VNVNNNNNV. Eilė yra pradėdama nuo nulio. For loop veikia iš galo, kad rasti paskutinįjį VIPa, šitame pavyzdyje jis yra 2 pozicijoje. Tada yra skaičiuojama pozicijų skirtumą tarp dabartinio(8) ir paskutiniojo(2). Čia bus 6. Kaip yra aprašyta užduotyje, turi būti max 3 ne VIPo vaikų tarp VIPų, tai reiškia skirtumas turi būti didesnis nei 4 kad priskirstyti VIPą į reikiamą vietą. Po rūšiavimo eilė yra VNVNNNVNNV ir reikia išnaikinti paskutinįjį VIPa, nes tai yra tas pats vaikas. Galutinis rezultatas: VNVNNNVNN. Jeigu skirtumas būtu 4 ir mažesnis, tai VIP liktų paskutinioje vietoje. Pvz: VNNV liks VNNV.

# Aplikacijos naudojimas

Iš kairinės pusės yra mygtukai, kuriuos galima spausti. Šituoje aplikacijoje negalima padaryti daugiau negu po vieną DoubleSwings, Carousel, Slide, BallPit objektą, nes mano tikslas buvo patestuoti turimus objektus.

Centre yra TextFields, kurios priima tekstus. Jie reikalingi kad galėčiau pašalinti vaiką, rašydamas jo poziciją eilėje, playsite ir t.t, arba gauti jo informaciją. Mano tikslas buvo patestuoti funkcijas ir pademonstruoti, tai čia TextFields veiks tik tada, kada rašysite skaičius ir tie skaičiai nebus didesni negu eilės, playsito containerio dydžiai, nes kitaip tai tik exceptionai kils.

Dešinėje pusėje bus parašytas tekstas su informacijomis, kurias gausite spaudžius mygtukus arba rašydami į textFields.