

## Projektni zadatak 1

-decembar 2020-

U programskom jeziku Java definisati klasu `Calculator`. Klasa `Calculator` posjeduje jedan privatni atribut tipa `Double` pod nazivom `currentValue`, te getter i seter za dati atribut. U sklopu navedene klase definisati metodu čiji je sljedeći potpis:

```
public void calculate(Double value, char operator)
```

Metoda na osnovu parametra `operator`, a čije su dozvoljene vrijednosti `+`, `-`, `*` i `/`, primjenjuje datu operaciju tako da je prvi operand `currentValue`, a drugi operand parametar metode `value`. Nakon primjenjene operacije rezultat se smješta u vrijednost `currentValue`. U slučaju da se vrši dijeljenje sa nulom kreirati ("baciti") izuzetak `DivisionByZeroException` (izuzetak je potrebno definisati). Ukoliko vrijednost parametra `operator` nije jedna od gore navedenih, potrebno je kreirati izuzetak `NotSupportedOperationException` (izuzetak je potrebno definisati).

Definisati i klasu `CalculatorAdvanced`. Navedena klasa nasljeđuje klasu `Calculator`. U sklopu navedene klase definisati i metodu čiji je sljedeći potpis:

```
public void calculateAdvanced(char action)
```

Metoda na osnovu parametra `action`, a čije su dozvoljene vrijednosti cijelobrojni podatak u osegu od 1 do 100 te karakter `!`, računa stepen odnosno faktoriyel broja (respektivno) za cjelobrojni dio vrijednosti `currentValue`, te rezultat smješta u vrijednost `currentValue`. Prilikom računanja stepena broja ukoliko vrijednost stepena nije u opsegu `[1, 100]`, te cjelobrojna vrijednost u promjenljivoj `currentValue` nije `>=1` kreira se izuzetak `PowerException` (izuzetak je potrebno definisati). Prilikom računanja faktoriyela broja vrijednost `currentValue` mora biti u opsegu `[0, 10]`. U suprotom se kreira izuzetak `NumberNotInAreaException` (izuzetak je potrebno definisati). Ukoliko vrijednost parametra `action` nije podržana potrebno je kreirati izuzetak `NotSupportedOperationException`. Definisati i metodu čiji je sljedeći potpis:

```
public Boolean hasCharacteristic(char value)
```

Metoda na osnovu parametra `value`, a čije su dozvoljene vrijednosti karakteri `A` i `P` vraćaju `true` u slučaju da je cjelobrojni dio vrijednosti `currentValue` Armstrongov broj (ako je navedena vrijednost `A`), odnosno savršen (ako je navedena vrijednost `P`). U slučaju da je cjelobrojni dio vrijednosti `currentValue` `< 1` kreirati izuzetak `NumberNotInAreaException`. Ukoliko vrijednost parametra `value` nije podržana potrebno je kreirati izuzetak `NotSupportedOperationException`. U svakom drugom slučaju metoda kao rezultat vraća `false`.

Napisati jedinične testove za gore navedene metode i konstruktore koristeći Junit5. Svi testovi, gdje god je to moguće, moraju biti parametrizovani (dozvoljeno je kreirati više odvojenih tesnih metoda

u zavisnosti od scenarija pa iste parametrizovati). Obavezno je koristiti Hamcrest mečere. Testovi moraju biti u odvojenom folderu u sklopu projekta. Svi nazivi klasa, atributa te potpisi metoda ne smiju biti modifikovani. Nije dozvoljen poziv već implementiranih Java metoda za neku od gore traženih akcija osim za `+`, `-`, `*` i `/` (recimo `Math.pow` nije dozvoljen već je stepenovanje potrebno samostalno implementirati). Potrebno je za sve metode koje se testiraju ostvariti potpunu pokrivenost testovima (coverage) kao i korigovati nedostatke ukoliko su prijavljeni nakon mutacionog testiranja. Sve klase i metode potrebno je dokumentovati na osnovu primjera sa predavanja te generisati Java dokumentaciju koristeći alat Javadoc. Dokumentaciju je potrebno pisati na engleskom jeziku. Kompletan kod sa testovima kao i dokumentacijom potrebno je dostaviti do 24.12.2020. u 12 časova.

### Napomene:

Armstrongove brojeve predstavljaju svi n-tocifreni prirodni brojevi koji su jednaki zbiru n-tih stepena cifara tog broja. Primjeri su:  $153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$  i  $1634 = 1^4 + 6^4 + 3^4 + 4^4 = 1 + 1296 + 81 + 256 = 1634$

Savršen broj je prirodan broj koji je jednak zbiru svojih pozitivnih djelilaca, uključujući i broj 1, ali ne računajući sam taj broj. Primjeri su:  $6 = 1 + 2 + 3$  i  $28 = 1 + 2 + 4 + 7 + 14$

Implementacija prvog dijela zadatka (klasa sa metodom `calculate`) sa svim realizovanim testovima, dokumentacijom i ostalim što je traženo zadatkom nosi 10 bodova. Implementacija drugog dijela zadatka (druga navedena metoda) nosi 7 bodova, dok treća metoda nosi 8 bodova.