

PROBA PROIEKTUA

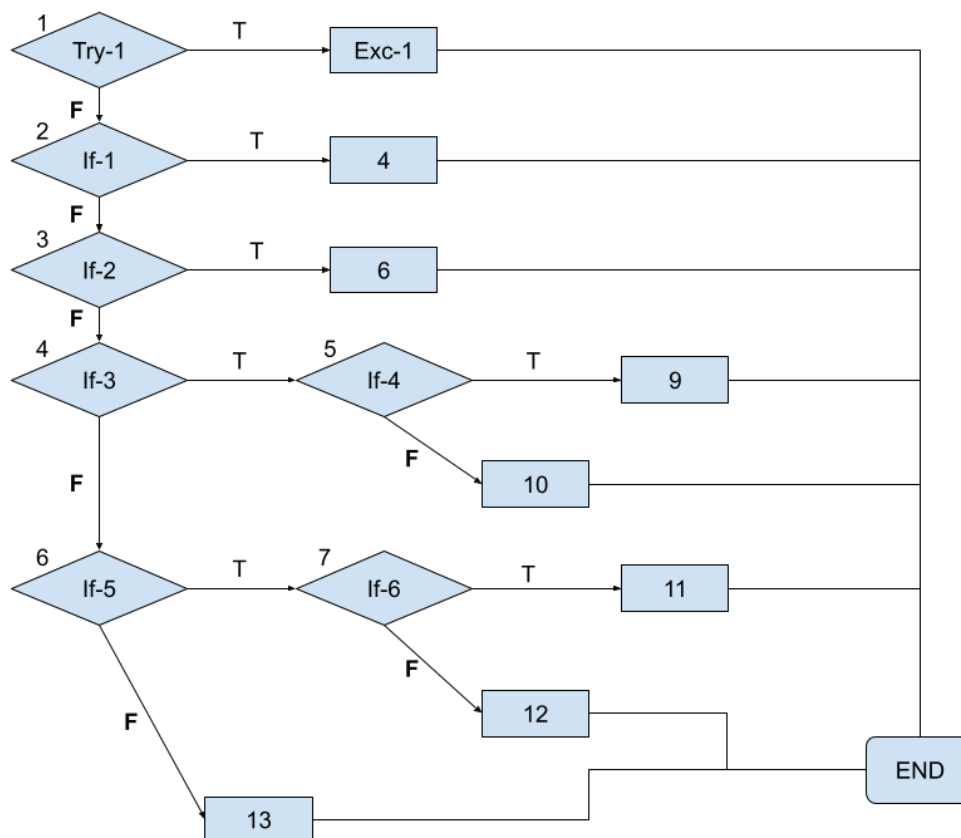
Egilea: Ander Pollacino Merino

Ordu kopurua: 16 ordu

DataAccess-etik aukeratu dudan metodoa *diruaSartu(String userName, double zenbat)* da:

```
public boolean diruaSartu(String userName, double zenbat) {
    try{
        if(zenbat <= 0) {
            return false;
        }
        Bezeroa bezero = db.find(Bezeroa.class, userName);
        db.getTransaction().begin();
        if(!bezero.isLimit()) {
            double actual = bezero.getCash();
            bezero.setCash(actual + zenbat);
            bezero.addTransactionInOut(0, zenbat);
            db.getTransaction().commit();
            System.out.println("amount:"+actual + zenbat);
            return true;
        } else if(!bezero.calculateDate() && bezero.isLimit()) {
            boolean res = true;
            double actual = bezero.getCash();
            if((bezero.getActualLimit() + zenbat) > bezero.getMaxLimit()) {
                res = false;
            } else {
                bezero.setCash(actual + zenbat);
                bezero.addTransactionInOut(0, zenbat);
                bezero.setActualLimit(bezero.getActualLimit() + actual + zenbat);
            }
            db.getTransaction().commit();
            System.out.println("amount:"+actual + zenbat);
            return res;
        } else if(bezero.isLimit()){
            bezero.setActualLimit(0);
            bezero.setLimitDate();
            boolean res = true;
            double actual = bezero.getCash();
            if((bezero.getActualLimit() + zenbat) > bezero.getMaxLimit()) {
                res = false;
            } else {
                bezero.setCash(actual + zenbat);
                bezero.addTransactionInOut(0, zenbat);
                bezero.setActualLimit(bezero.getActualLimit() + actual + zenbat);
            }
            db.getTransaction().commit();
            System.out.println("amount:"+actual + zenbat);
            return res;
        } else {
            return false;
        }
    } catch(Exception e){
        System.out.println("Error Amount");
        return false;
    }
}
```

Kutxa txuria:



Konplexutasun ziklomatikoa:
erabaki nodo + 1 -> 7+1 = 8

Proba kasuak:

#	Bidea	Sarrerako baldintza	Sarrerako datua	DB egoera	Irteera
1	1		(null, 10.0)		False
2	1,2	zenbat<=0	("ander", 0.0)		False
3	1,2,3	!isLimit	("ander", 6)	("ander", 10, "2348375K", 0.2, 02/09/2021, 0.0, 0.0, false)	True
4	1,2,3,4,5(T)	!calculatedDate && isLimit	("ander", 20)	("ander", 10, "2348375K", 0.2, 02/09/2021, 50.0, 40.0, true)	False
5	1,2,3,4,5(F)	!calculatedDate && isLimit	("ander", 10)	("ander", 10, "2348375K", 0.2, 02/09/2021, 50.0, 20.0, true)	True
6	1,2,3,4,6,7(T)	isLimit	("ander", 55)	("ander", 10, "2348375K", 0.2, 02/09/2021, 50.0, 40.0, true)	False
7	1,2,3,4,6,7(F)	isLimit	("ander", 10)	("ander", 10, "2348375K", 0.2, 02/09/2021, 50.0, 20.0, true)	True

8	1,2,3,4,6(F)		("ander", 10)		False
---	--------------	--	---------------	--	-------

Implementazioa: DiruaSartuDAW.java

Aurkitutako akatsak: Proba kasu guztiak espero zuten emaitza eman dute.

Kutxa beltza:

	Sarrera baldintza	BK egokia	BK ez egokia
Sarrerako datuak	String null	userName!=null (1)	userName==null (7)
	userName DB-n	userName ∈ DB (2)	userName ∉ DB (8)
	zenbat positiboa	zenbat > 0 (3)	zenbat <= 0 (9)
Klaseko semantika	kalkulateDate	bezero.calculateDate() (4)	!bezero.calculateDate() (10)
	isLimit	bezero.isLimit() (5)	!bezero.isLimit() (11)
		bezero.getActualLimit+zenbat<b ezero.getMaxLimit (6)	bezero.getActualLimit+zenbat> =bezero.getMaxLimit (12)

Proba kasuak eta muga balioak:

#	Estalitakoa	Sarrera	DB egoera	Emaitza
1	1,2,3,4,5,6	("ander", 55)	("ander",10,"2348375K",0.2,02/09/2021, 50.0, 30.0,true)	True
2	7	(" ", 20)		False
3	8	("ander", 10)	("Jon",10,"2348375K",0.2,02/09/2021, 50.0, 20.0,true)	False
4	1,2,4,5	("ander", 9)	("ander",10,"2348375K",0.2,03/10/2021, 50.0, 40.0,true)	True
5	1,2,4,5	("ander", 10)	("ander",10,"2348375K",0.2,03/10/2021, 50.0, 40.0,true)	True
6	1,2,3,5	("ander", 11)	("ander",10,"2348375K",0.2,03/10/2021, 50.0, 40.0,true)	False

Implementazioa: DiruaSartuDAB.java

Aurkitutako akatsak: Proba kasu eta muga balioa guztiak espero zuten emaitza eman dute.

Ezin izan dut Mockito-ko probak egin, nahiz eta pom.xml fitxategia ondo konfiguratuta izan eta klasekideei laguntza eskatu ez dut lortu errore hauek kentzea, eta ezinezkoa egiten zuen proba kasuak implementatzea:

GitHub-eko esteka: <https://github.com/anderpm/Bets>

Sonarcloud-eko esteka: <https://sonarcloud.io/dashboard?id=anderpm>