



Merisa Beširović



Low level pozivi u Solidity–ju

START



Low vs high level call

low

- Veća kontrola
- Ručno upravljanje gasom
- Bezbednosni izazovi
- Efikasnost

high

- Automatsko upravljanje gasom
- Jednostavnost koda
- Integracija sa Solidity apstrakcijama
- Zaštita od reentrancy napada
- Viša apstrakcija



High level calls

- Bezbednosni standardi
- Održavanje koda
- Prevencija od grešaka

Zašto low level calls?

- Veća kontrola
- Interoperabilnost
- Optimizacija gasa i koda



Call

slanje Ether pozivom fallback funkcije

- Revert-ovi se ne prenose dalje
- Bypass-ovanje provere tipa
- Izostavljanje provere postojanja funkcije

(bool success, bytes memory data) = address.call(bytes memory data);



Delegate call

slično call pozivu

koristi u situacijama gde je potrebno izvršiti kod drugog ugovora sa kontekstom pozivačkog

Razlike:

1. Kontekst pozivačkog ugovora
2. Ponašanje prilikom grešaka
3. Gas potrošnja

(bool success, bytes memory data) = address.delegatecall(bytes memory data);



Alice

inc()

1 ether

Contract A

uint x

```
func inc() {  
    B.inc()  
}
```

`msg.sender` = Alice
`msg.value` = 1 ether
`x` = 0

inc()

Contract B

uint x

```
func inc() {  
    x += 1  
}
```

`msg.sender` = contract
`msg.value` = 0 ether
`x` = 1



Alice

inc()
1 ether

Contract A

uint x

```
func inc() {  
    B.delegatecall  
}
```

msg.sender = Alice
msg.value = 1 ether
x = 1

delegatecall
inc()

Contract B

uint x

```
func inc() {  
    x += 1  
}
```

msg.sender = Alice
msg.value = 1 ether
x = 0



Staticcall

- koristi se za pozivanje funkcije u eksternom ugovoru, ali ne omogućava eksternom ugovoru da promeni stanje
- korisna za čitanje podataka iz eksternog ugovora, pri čemu se održava konstantno stanje

(bool success, bytes memory data) = address.delegatecall(bytes memory data);



Send

- prenos malih iznosa drugim ugovorima ili nalogima
- funkcija pruža samo 2300 gasa
- send funkcija ne vraća nikakve podatke iz eksternog ugovora ili naloga

bool success = address.send(uint256 amount);