

Minne

INF101 forelesning 7. februar 2023

Torstein Strømme

Stikkord: stack, heap, primitiver, objekter, likhet

I dag

- Primitive typer vs refererte typer
- Minne
- Likhet

Primitive typer

- Boolske verdier
 - `boolean` - en logisk verdi som kan være enten true eller false
- Heltall
 - `byte` - et heltall mellom -128 og 127
 - `short` - et heltall mellom -32768 og 32767
 - `int` - et heltall mellom -2147483648 og 2147483647
 - `long` - et heltall mellom -9223372036854775808 og 9223372036854775807
- Flyttall
 - `float` - et flyttall med 32 biters presisjon
 - `double` - et flyttall med 64 biters presisjon
- Skrifttegn
 - `char` - et enkelt tegn, for eksempel en bokstav

Refererte typer

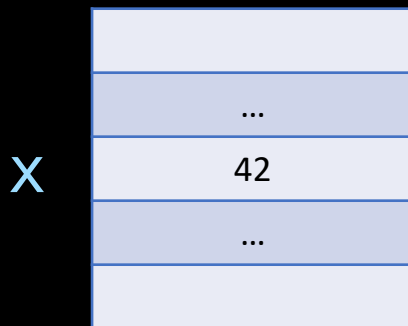
- Alle typer som ikke er primitive
 - `String`
 - `Integer`, `Double`, `Boolean`
 - `int[]`, `double[]`, `String[]`
 - `ArrayList`
 - `LocalDateTime`
 - `IPokemon`, `Pokemon`
 - `CommandLineInterface`, `Command`
 - `DummyShell`, `EchoShell`, `SimpleShell`
 - ...
- Verdier som ikke har en primitiv type kalles *objekter*

Primitive vs refererte typer

Hver variabel har én plass i minnet

- Primitive typer lagres direkte på en variabelen sin plass

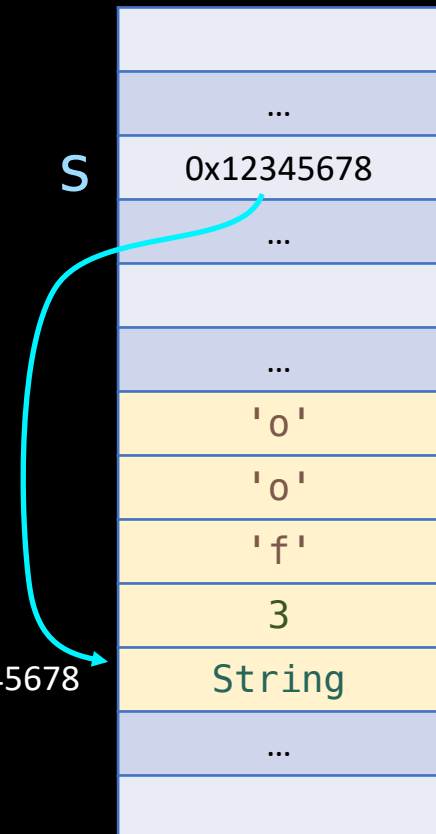
```
int x = 42;
```



- Refererte typer lagres *et annet* sted i minne – på variabelen sin plass lagres en *referanse* til dette stedet.

```
String s = "foo";
```

minneadresse 0x12345678



Likhet

<https://inf101.i.uib.no/notat/typer/#likhet-og-sammenligning>

Minne

- «Random access memory» (RAM)
- Brukes når prosesser kjører
- Alt slettes når strømmen blir borte
- Kommer i ulike størrelser, f. eks. 8GB
- Operativsystemet «faker» minne slik at hver prosess tror at de har alt minnet for seg selv.



Minne

- 64-bit arkitektur → 64 bits per celle

- 8GB RAM → $64 \cdot 10^9$ bits → 10^9 celler

1 Byte = 8 bits
(derfor er 8GB = 64Gb)

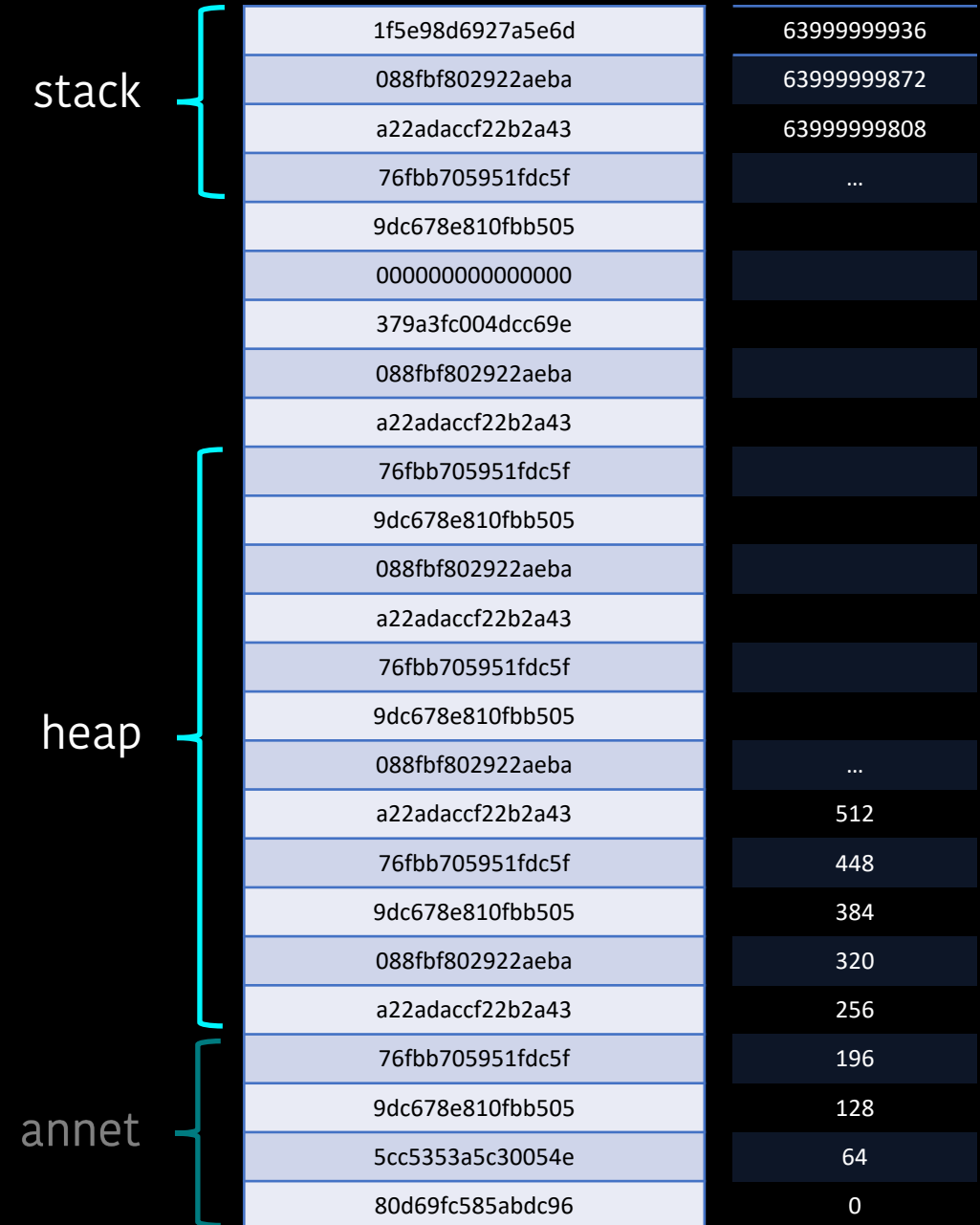
Minneadresse til celle: posisjonen til første bit i cellen

The diagram shows a vertical stack of memory cells. Each cell is represented by a horizontal bar. The left column contains hexadecimal addresses, and the right column contains decimal values. A yellow bracket at the top groups the first three cells. A cyan bracket on the left side groups the entire stack. A pink arrow points from the text 'Minneadresse til celle: posisjonen til første bit i cellen' to the first cell's address. Another pink arrow points from the text '1 Byte = 8 bits (derfor er 8GB = 64Gb)' to the first cell's address. The values in the right column decrease from top to bottom, starting at 63999999936 and ending at 0.

1f5e98d6927a5e6d	63999999936
088fbf802922aeba	63999999872
a22adaccf22b2a43	63999999808
76fbb705951fdc5f	...
9dc678e810fbb505	
0000000000000000	
379a3fc004dcc69e	
088fbf802922aeba	
a22adaccf22b2a43	
76fbb705951fdc5f	
9dc678e810fbb505	
088fbf802922aeba	
a22adaccf22b2a43	
76fbb705951fdc5f	
9dc678e810fbb505	
088fbf802922aeba	...
a22adaccf22b2a43	512
76fbb705951fdc5f	448
9dc678e810fbb505	384
088fbf802922aeba	320
a22adaccf22b2a43	256
76fbb705951fdc5f	196
9dc678e810fbb505	128
5cc5353a5c30054e	64
80d69fc585abdc96	0

Minne

- Stack:
 - Lokale variabler/parametre
- Heap:
 - Objekter
 - Instansvariabler
- Annet:
 - Kildekode (Java: bytekode)
 - Klassevariabler/konstanter
 - Størrelse fastsettes under kompilering



Objekter i minnet

<https://inf101.i.uib.no/notat/objekter/#objekter-i-minnet>

Større eksempel: pokemon

Eksempel: pokemon

```
public static void main(String[] args) {  
    Ipokemon pokemon1 = new Pokemon("Bidoof");  
    Ipokemon pokemon2 = new Pokemon("Feebas");  
    System.out.println(pokemon1);  
    System.out.println(pokemon2);  
  
    int rounds = 0;  
    while (pokemon1.isAlive() && pokemon2.isAlive()) {  
        pokemon1.attack(pokemon2);  
        if (pokemon2.isAlive()) {  
            pokemon2.attack(pokemon1);  
        }  
        rounds++;  
    }  
    System.out.print(rounds + " rounds were played");  
}
```

main

args: null

stack



args

088fbf802922aeba	63999999936
a22adaccf22b2a43	63999999872
76fbb705951fdc5f	63999999808
9dc678e810fbb505	...
0000000000000000	
379a3fc004dcc69e	
088fbf802922aeba	
a22adaccf22b2a43	
76fbb705951fdc5f	
9dc678e810fbb505	
088fbf802922aeba	
a22adaccf22b2a43	
76fbb705951fdc5f	
9dc678e810fbb505	
088fbf802922aeba	
a22adaccf22b2a43	
76fbb705951fdc5f	
9dc678e810fbb505	
088fbf802922aeba	...
a22adaccf22b2a43	512
76fbb705951fdc5f	448
9dc678e810fbb505	384
088fbf802922aeba	320
a22adaccf22b2a43	256
76fbb705951fdc5f	196
9dc678e810fbb505	128
5cc5353a5c30054e	64
80d69fc585abdc96	0

Eksempel: pokemon

```
public static void main(String[] args) {  
    Ipokemon pokemon1 = new Pokemon("Bidoof");  
    Ipokemon pokemon2 = new Pokemon("Feebas");  
    System.out.println(pokemon1);  
    System.out.println(pokemon2);  
  
    int rounds = 0;  
    while (pokemon1.isAlive() && pokemon2.isAlive()) {  
        pokemon1.attack(pokemon2);  
        if (pokemon2.isAlive()) {  
            pokemon2.attack(pokemon1);  
        }  
        rounds++;  
    }  
    System.out.print(rounds + " rounds were played");  
}
```

main

args: null

pokemon1: 0x00B1D00F

stack { args
pokemon1

heap { et objekt

null	63999999936
0xB1D00F	63999999872
a22adaccf22b2a43	63999999808
76fbb705951fdc5f	...
9dc678e810fbb505	
0000000000000000	
379a3fc004dcc69e	
088fbf802922aeba	
a22adaccf22b2a43	
76fbb705951fdc5f	
9dc678e810fbb505	
088fbf802922aeba	
a22adaccf22b2a43	
76fbb705951fdc5f	
9dc678e810fbb505	
Random-objekt	0xB1D00F + 320
19	0xB1D00F + 256
97	0xB1D00F + 192
97	0xB1D00F + 128
"Bidoof"	0xB1D00F + 64
Pokemon	0xB1D00F
76fbb705951fdc5f	...
9dc678e810fbb505	128
5cc5353a5c30054e	64
80d69fc585abdc96	0

Eksempel: pokemon

```

public static void main(String[] args) {
    Ipokemon pokemon1 = new Pokemon("Bidoof");
    Ipokemon pokemon2 = new Pokemon("Feebas");
    System.out.println(pokemon1);
    System.out.println(pokemon2);

    int rounds = 0;
    while (pokemon1.isAlive() && pokemon2.isAlive()) {
        pokemon1.attack(pokemon2);
        if (pokemon2.isAlive()) {
            pokemon2.attack(pokemon1);
        }
        rounds++;
    }
    System.out.print(rounds + " rounds were played");
}
    
```

main

args: null

pokemon1: 0x00B1D00F

pokemon2: 0x00FEEBA5

stack
 { args
 { pokemon1
 { pokemon2

heap
 { et objekt
 { et objekt

null	63999999936
0xB1D00F	63999999872
0xFEEBA5	63999999808
76fbb705951fdc5f	...
9dc678e810fbb505	
0000000000000000	
379a3fc004dcc69e	
088fbf802922aeba	
Random-objekt	0xFEEBA5 + 320
17	0xFEEBA5 + 256
102	0xFEEBA5 + 192
102	0xFEEBA5 + 128
"Feebas"	0xFEEBA5 + 64
Pokemon	0xFEEBA5
9dc678e810fbb505	...
Random-objekt	0xB1D00F + 320
19	0xB1D00F + 256
97	0xB1D00F + 192
97	0xB1D00F + 128
"Bidoof"	0xB1D00F + 64
Pokemon	0xB1D00F
76fbb705951fdc5f	...
9dc678e810fbb505	128
5cc5353a5c30054e	64
80d69fc585abdc96	0

Eksempel: pokemon

```

public static void main(String[] args) {
    Ipokemon pokemon1 = new Pokemon("Bidoof");
    Ipokemon pokemon2 = new Pokemon("Feebas");
    System.out.println(pokemon1);
    System.out.println(pokemon2);

    int rounds = 0;
    while (pokemon1.isAlive() && pokemon2.isAlive()) {
        pokemon1.attack(pokemon2);
        if (pokemon2.isAlive()) {
            pokemon2.attack(pokemon1);
        }
        rounds++;
    }
    System.out.print(rounds + " rounds were played");
}
    
```

main

args: null
rounds: 0

pokemon1: 0x00B1D00F
pokemon2: 0x00FEEBA5

stack
args
pokemon1
pokemon2
rounds

heap
et objekt
et objekt

args	null	63999999936
pokemon1	0xB1D00F	63999999872
pokemon2	0xFEEBA5	63999999808
rounds	0	...
	9dc678e810fbb505	
	0000000000000000	
	379a3fc004dcc69e	
	088fbf802922aeba	
	Random-objekt	0xFEEBA5 + 320
	17	0xFEEBA5 + 256
	102	0xFEEBA5 + 192
	102	0xFEEBA5 + 128
	"Feebas"	0xFEEBA5 + 64
	Pokemon	0xFEEBA5
	9dc678e810fbb505	...
	Random-objekt	0xB1D00F + 320
	19	0xB1D00F + 256
	97	0xB1D00F + 192
	97	0xB1D00F + 128
	"Bidoof"	0xB1D00F + 64
	Pokemon	0xB1D00F
	76fbb705951fdc5f	...
	9dc678e810fbb505	128
	5cc5353a5c30054e	64
	80d69fc585abdc96	0

Eksempel: pokemon

```

public static void main(String[] args) {
    Ipokemon pokemon1 = new Pokemon("Bidoof");
    Ipokemon pokemon2 = new Pokemon("Feebas");
    System.out.println(pokemon1);
    System.out.println(pokemon2);

    int rounds = 0;
    while (pokemon1.isAlive() && pokemon2.isAlive()) {
        pokemon1.attack(pokemon2);
        if (pokemon2.isAlive()) {
            pokemon2.attack(pokemon1);
        }
        rounds++;
    }
    System.out.print(rounds + " rounds were played");
}
    
```

main

args: null
rounds: 0

pokemon1: 0x00B1D00F
pokemon2: 0x00FEEBA5

stack
args
pokemon1
pokemon2
rounds

heap
et objekt
et objekt

args	null	63999999936
pokemon1	0xB1D00F	63999999872
pokemon2	0xFEEBA5	63999999808
rounds	0	...
	9dc678e810fbb505	
	0000000000000000	
	379a3fc004dcc69e	
	088fbf802922aeba	
	Random-objekt	0xFEEBA5 + 320
	17	0xFEEBA5 + 256
	102	0xFEEBA5 + 192
	102	0xFEEBA5 + 128
	"Feebas"	0xFEEBA5 + 64
	Pokemon	0xFEEBA5
	9dc678e810fbb505	...
	Random-objekt	0xB1D00F + 320
	19	0xB1D00F + 256
	97	0xB1D00F + 192
	97	0xB1D00F + 128
	"Bidoof"	0xB1D00F + 64
	Pokemon	0xB1D00F
	76fbb705951fdc5f	...
	9dc678e810fbb505	128
	5cc5353a5c30054e	64
	80d69fc585abdc96	0

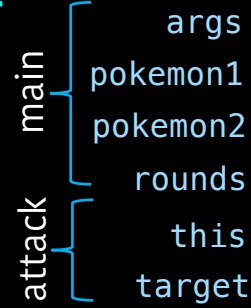
Eksempel: pokemon

```

public static void main(String[] args) {
    // ...
    public void attack(IPokemon target) {
        int damageInflicted = (int) (this.strength
            + this.strength / 2 * random.nextGaussian());

        if (damageInflicted < 0) {
            damageInflicted = 0;
        }
        System.out.println(this.getName() + " attacks "
            + target.getName());
        target.damage(damageInflicted);
        if (!target.isAlive()) {
            System.out.println(target.getName()
                + " defeated by " + this.getName());
        }
    }
}
    
```

stack



heap

et objekt

et objekt

null
0xB1D00F
0xFEEBA5
0
0xB1D00F
0xFEEBA5
379a3fc004dcc69e
088fbf802922aeba
Random-objekt
17
102
102
"Feebas"
Pokemon
9dc678e810fbb505
Random-objekt
19
97
97
"Bidoof"
Pokemon
76fbb705951fdc5f
9dc678e810fbb505
5cc5353a5c30054e
80d69fc585abdc96

63999999936
63999999872
63999999808
...
0xFEEBA5 + 320
0xFEEBA5 + 256
0xFEEBA5 + 192
0xFEEBA5 + 128
0xFEEBA5 + 64
0xFEEBA5
...
0xB1D00F + 320
0xB1D00F + 256
0xB1D00F + 192
0xB1D00F + 128
0xB1D00F + 64
0xB1D00F
...
128
64
0

attack

this: 0x00B1D00F
target: 0x00FEEBA5

main

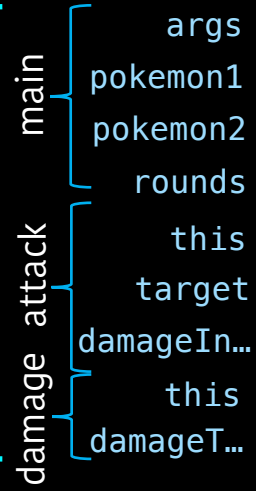
args: null
rounds: 0
pokemon1: 0x00B1D00F
pokemon2: 0x00FEEBA5

Eksempel: pokemon

```

public static void main(String[] args) {
    // ...
    public void attack(IPokemon target) {
        // ...
        public void damage(int damageTaken) {
            if (damageTaken < 0) {
                damageTaken = 0;
            }
            if (damageTaken > this.healthPoints) {
                damageTaken = healthPoints;
            }
            this.healthPoints -= damageTaken;
            System.out.println(this.name + " takes "
                + damageTaken + " damage and is "
                + "left with " + this.healthPoints + "/"
                + this.maxHealthPoints + " HP");
        }
    }
}
    
```

stack

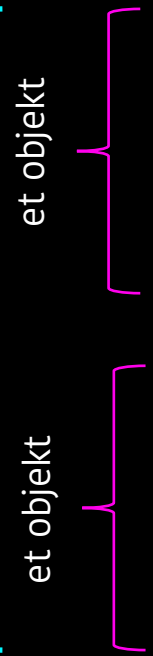


damage this: 0x00FEEBA5
 damageTaken: 20

attack damageInflicted: 20 this: 0x00B1D00F
 target: 0x00FEEBA5

main args: null pokemon1: 0x00B1D00F
 rounds: 0 pokemon2: 0x00FEEBA5

heap



args	null	63999999936
pokemon1	0xB1D00F	63999999872
pokemon2	0xFEEBA5	63999999808
rounds	0	...
this	0xB1D00F	
target	0xFEEBA5	
damageIn...	20	
this	0xFEEBA5	
damageT...	20	
	76fbb705951fdc5f	
	9dc678e810fbb505	
	Random-objekt	0xFEEBA5 + 320
	17	0xFEEBA5 + 256
	102	0xFEEBA5 + 192
	82	0xFEEBA5 + 128
	"Feebas"	0xFEEBA5 + 64
	Pokemon	0xFEEBA5
	9dc678e810fbb505	...
	Random-objekt	0xB1D00F + 320
	19	0xB1D00F + 256
	97	0xB1D00F + 192
	97	0xB1D00F + 128
	"Bidoof"	0xB1D00F + 64
	Pokemon	0xB1D00F
	80d69fc585abdc96	...

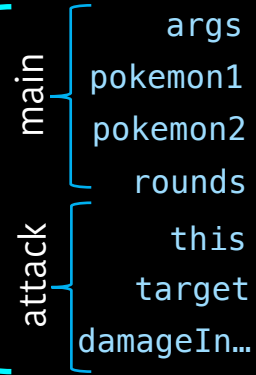
Eksempel: pokemon

```

public static void main(String[] args) {
    // ...
    public void attack(IPokemon target) {
        int damageInflicted = (int) (this.strength
            + this.strength / 2 * random.nextGaussian());

        if (damageInflicted < 0) {
            damageInflicted = 0;
        }
        System.out.println(this.getName() + " attacks "
            + target.getName());
        target.damage(damageInflicted);
        if (!target.isAlive()) {
            System.out.println(target.getName()
                + " defeated by " + this.getName());
        }
    }
}
    
```

stack



attack	damageInflicted: 20	this: 0x00B1D00F target: 0x00FEEBA5
--------	---------------------	----------------------------------------

main	args: null rounds: 0	pokemon1: 0x00B1D00F pokemon2: 0x00FEEBA5
------	-------------------------	----------------------------------------------

heap

et objekt

et objekt

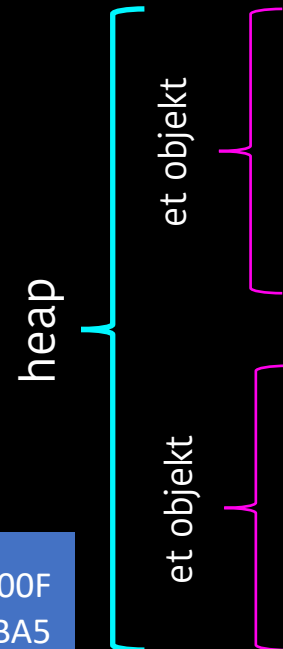
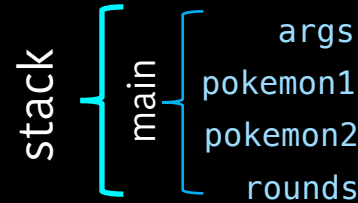
args	null	63999999936
pokemon1	0xB1D00F	63999999872
	0xFEEBA5	63999999808
pokemon2	0	...
rounds	0	...
this	0xB1D00F	...
	0xFEEBA5	...
target	20	...
	0xFEEBA5	...
damageIn...	20	...
	76fbb705951fdc5f	...
	9dc678e810fbb505	...
	Random-objekt	0xFEEBA5 + 320
	17	0xFEEBA5 + 256
	102	0xFEEBA5 + 192
	82	0xFEEBA5 + 128
	"Feebas"	0xFEEBA5 + 64
	Pokemon	0xFEEBA5
	9dc678e810fbb505	...
	Random-objekt	0xB1D00F + 320
	19	0xB1D00F + 256
	97	0xB1D00F + 192
	97	0xB1D00F + 128
	"Bidoof"	0xB1D00F + 64
	Pokemon	0xB1D00F
	80d69fc585abdc96	...

Eksempel: pokemon

```

public static void main(String[] args) {
    Ipokemon pokemon1 = new Pokemon("Bidoof");
    Ipokemon pokemon2 = new Pokemon("Feebas");
    System.out.println(pokemon1);
    System.out.println(pokemon2);

    int rounds = 0;
    while (pokemon1.isAlive() && pokemon2.isAlive()) {
        pokemon1.attack(pokemon2);
        if (pokemon2.isAlive()) {
            pokemon2.attack(pokemon1);
        }
        rounds++;
    }
    System.out.print(rounds + " rounds were played");
}
    
```



0x00000000	null	63999999936
0x00000004	0xB1D00F	63999999872
0x00000008	0xFEEBA5	63999999808
0x0000000C	0	...
0x00000010	0xB1D00F	
0x00000014	0xFEEBA5	
0x00000018	20	
0x0000001C	0xFEEBA5	
0x00000020	20	
0x00000024	76fbb705951fdc5f	
0x00000028	9dc678e810fbb505	
0x0000002C	Random-objekt	0xFEEBA5 + 320
0x00000030	17	0xFEEBA5 + 256
0x00000034	102	0xFEEBA5 + 192
0x00000038	82	0xFEEBA5 + 128
0x0000003C	"Feebas"	0xFEEBA5 + 64
0x00000040	Pokemon	0xFEEBA5
0x00000044	9dc678e810fbb505	...
0x00000048	Random-objekt	0xB1D00F + 320
0x0000004C	19	0xB1D00F + 256
0x00000050	97	0xB1D00F + 192
0x00000054	97	0xB1D00F + 128
0x00000058	"Bidoof"	0xB1D00F + 64
0x0000005C	Pokemon	0xB1D00F
0x00000060	80d69fc585abdc96	...

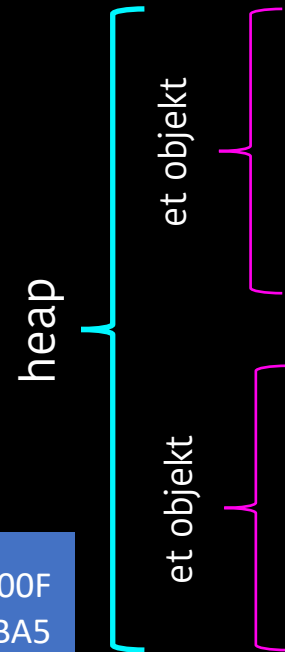
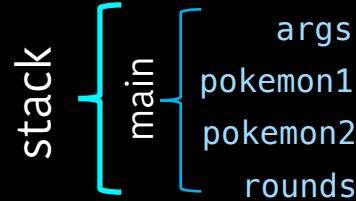
main	args: null rounds: 0	pokemon1: 0x00B1D00F pokemon2: 0x00FEEBA5
------	-------------------------	----------------------------------------------

Eksempel: pokemon

```

public static void main(String[] args) {
    Ipokemon pokemon1 = new Pokemon("Bidoof");
    Ipokemon pokemon2 = new Pokemon("Feebas");
    System.out.println(pokemon1);
    System.out.println(pokemon2);

    int rounds = 0;
    while (pokemon1.isAlive() && pokemon2.isAlive()) {
        pokemon1.attack(pokemon2);
        if (pokemon2.isAlive()) {
            pokemon2.attack(pokemon1);
        }
        rounds++;
    }
    System.out.print(rounds + " rounds were played");
}
    
```



0	63999999936
0xB1D00F	63999999872
0xFEEBA5	63999999808
0	...
0xB1D00F	
0xFEEBA5	
20	
0xFEEBA5	
20	
76fbb705951fdc5f	
9dc678e810fbb505	
Random-objekt	0xFEEBA5 + 320
17	0xFEEBA5 + 256
102	0xFEEBA5 + 192
82	0xFEEBA5 + 128
"Feebas"	0xFEEBA5 + 64
Pokemon	0xFEEBA5
9dc678e810fbb505	...
Random-objekt	0xB1D00F + 320
19	0xB1D00F + 256
97	0xB1D00F + 192
97	0xB1D00F + 128
"Bidoof"	0xB1D00F + 64
Pokemon	0xB1D00F
80d69fc585abdc96	...

main	args: null rounds: 0	pokemon1: 0x00B1D00F pokemon2: 0x00FEEBA5
------	-------------------------	----------------------------------------------

Over tid

Eksempel: pokemon

```
public static void main(String[] args) {
    Ipokemon pokemon1 = new Pokemon("Pikachu");
    Ipokemon pokemon2 = new Pokemon("Oddish");
    System.out.println(pokemon1);
    System.out.println(pokemon2);

    while (pokemon1.isAlive() && pokemon2.isAlive()) {
        pokemon1.attack(pokemon2);
        if (pokemon2.isAlive()) {
            pokemon2.attack(pokemon1);
        }
    }
}
```

...
Random-objekt
17
102
102
"Feebas"
Pokemon
...
Random-objekt
19
97
97
"Bidoof"
Pokemon
...

damage this: 0x00FEEBA5
damageTaken: 20

damage this: 0x00B1D00F
damageTaken: 18

damage this: 0x00FEEBA5
damageTaken: 21

attack this: 0x00B1D00F
target: 0x00FEEBA5

attack this: 0x00FEEBA5
target: 0x00B1D00F

attack this: 0x00B1D00F
target: 0x00FEEBA5



main

args: null
rounds: 1

pokemon1: 0x00B1D00F
pokemon2: 0x00FEEBA5

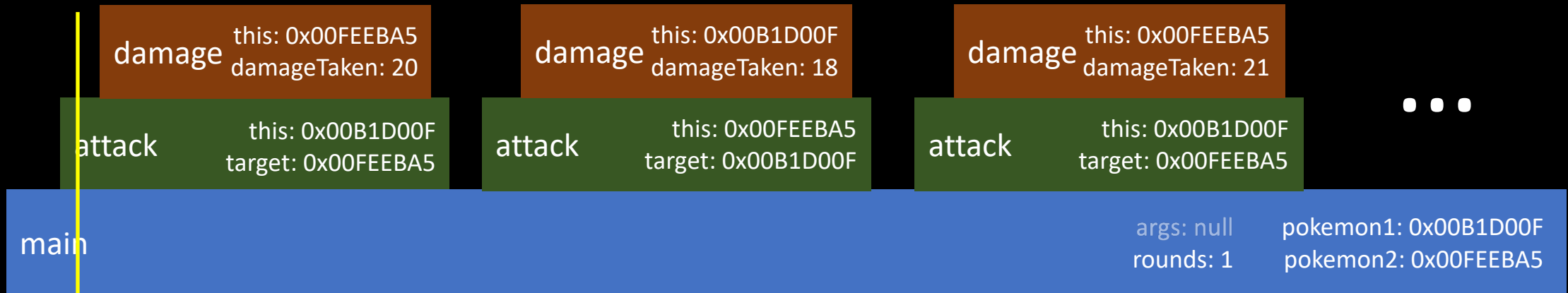
Eksempel: pokemon

```

public static void main(String[] args) {
    // ...
    public void attack(IPokemon target) {
        int damageInflicted = (int) (this.strength + this.strength / 2 * random.nextGaussian());
        if (damageInflicted < 0) {
            damageInflicted = 0;
        }

        System.out.println(this.getName() + " attacks " + target.getName());
        target.damage(damageInflicted);
        if (!target.isAlive()) {
            System.out.println(target.getName() + " is defeated by " + this.getName());
        }
    }
}
    
```

...
Random-objekt
17
102
102
"Feebas"
Pokemon
...
Random-objekt
19
97
97
"Bidoof"
Pokemon
...

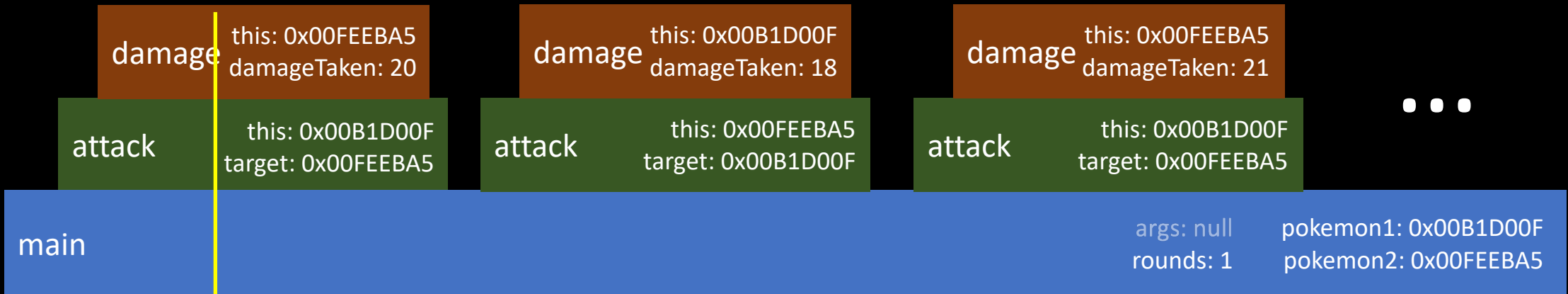


Eksempel: pokemon

```

public static void main(String[] args) {
    // ...
    public void attack(IPokemon target) {
        // ...
        public void damage(int damageTaken) {
            if (damageTaken < 0) {
                damageTaken = 0;
            }
            if (damageTaken > this.healthPoints) {
                damageTaken = healthPoints;
            }
            this.healthPoints -= damageTaken;
            System.out.println(this.name + " takes " + damageTaken + " damage and is "
                + "left with " + this.healthPoints + "/" + this.maxHealthPoints + " HP");
        }
    }
}
    
```

...
Random-objekt
17
102
102
"Feebas"
Pokemon
...
Random-objekt
19
97
97
"Bidoof"
Pokemon
...



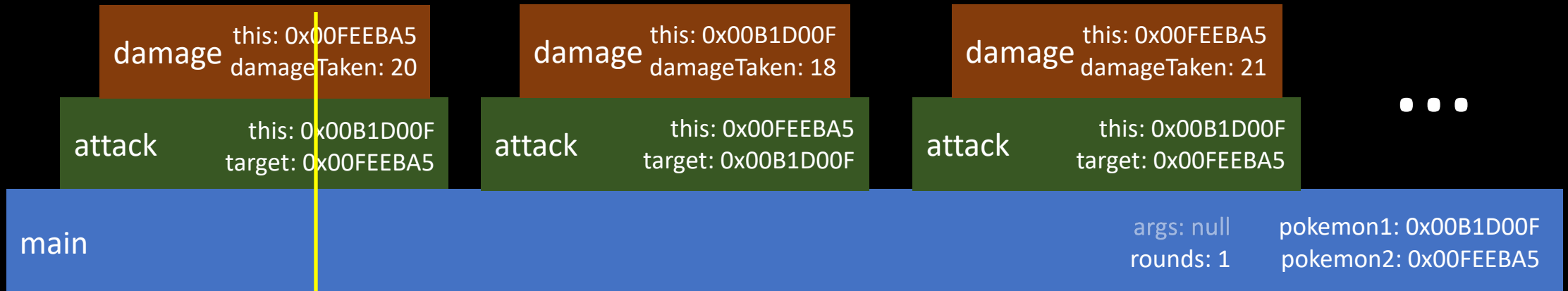
Eksempel: pokemon

```
public static void main(String[] args) {
    Pokemon p1 = new Pokemon("Bidoof", 100, 100);
    Pokemon p2 = new Pokemon("Feebas", 100, 100);
    p1.attack(p2);
    p2.attack(p1);
    p1.attack(p2);
}

public void attack(IPokemon target) {
    damage((int) Math.random() * target.maxHealthPoints);
}

public void damage(int damageTaken) {
    if (damageTaken < 0) {
        damageTaken = 0;
    }
    if (damageTaken > this.healthPoints) {
        damageTaken = healthPoints;
    }
    this.healthPoints -= damageTaken;
    System.out.println(this.name + " takes " + damageTaken + " damage and is "
        + "left with " + this.healthPoints + "/" + this.maxHealthPoints + " HP");
}
```

...
Random-objekt
17
102
82
"Feebas"
Pokemon
...
Random-objekt
19
97
97
"Bidoof"
Pokemon
...



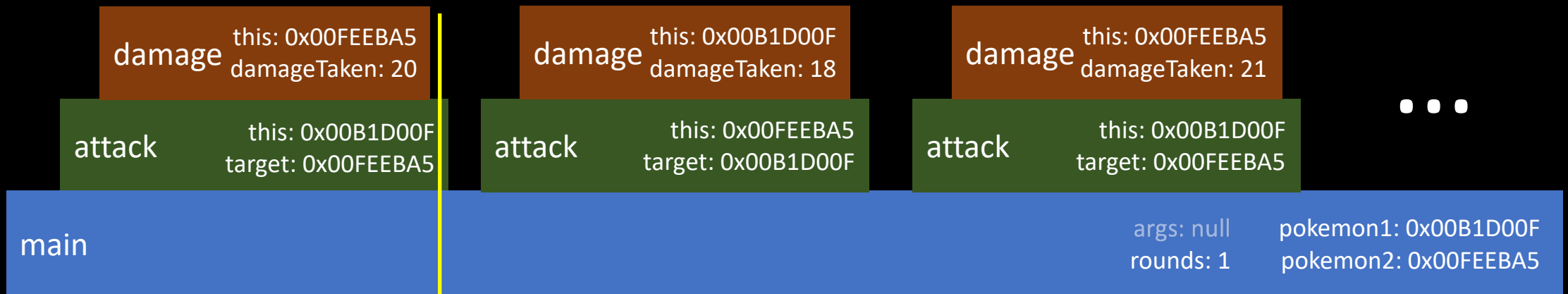
Eksempel: pokemon

```

public static void main(String[] args) {
    // ...
    public void attack(IPokemon target) {
        int damageInflicted = (int) (this.strength + this.strength / 2 * random.nextGaussian());
        if (damageInflicted < 0) {
            damageInflicted = 0;
        }

        System.out.println(this.getName() + " attacks " + target.getName());
        target.damage(damageInflicted);
        if (!target.isAlive()) {
            System.out.println(target.getName() + " is defeated by " + this.getName());
        }
    }
}
    
```

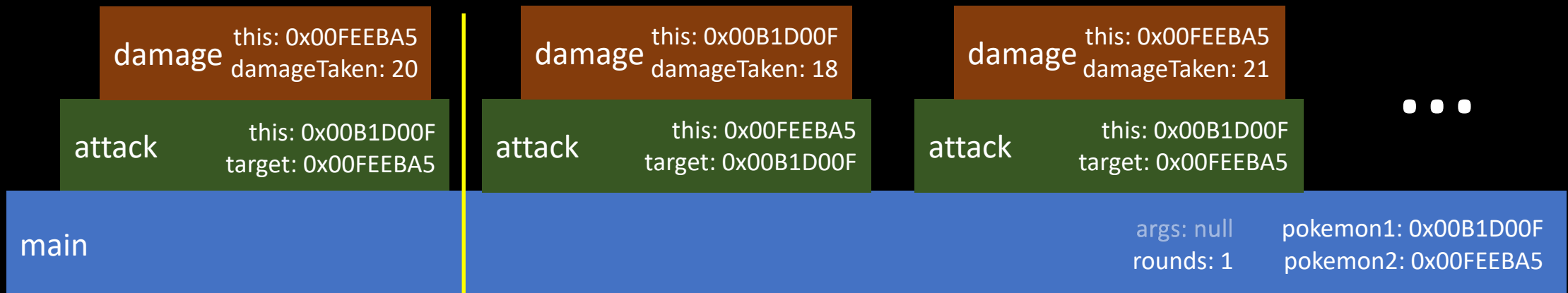
...
Random-objekt
17
102
82
"Feebas"
Pokemon
...
Random-objekt
19
97
97
"Bidoof"
Pokemon
...



Eksempel: pokemon

```
public static void main(String[] args) {  
    Ipokemon pokemon1 = new Pokemon("Pikachu");  
    Ipokemon pokemon2 = new Pokemon("Oddish");  
    System.out.println(pokemon1);  
    System.out.println(pokemon2);  
  
    while (pokemon1.isAlive() && pokemon2.isAlive()) {  
        pokemon1.attack(pokemon2);  
        if (pokemon2.isAlive()) {  
            pokemon2.attack(pokemon1);  
        }  
    }  
}
```

...
Random-objekt
17
102
82
"Feebas"
Pokemon
...
Random-objekt
19
97
97
"Bidoof"
Pokemon
...

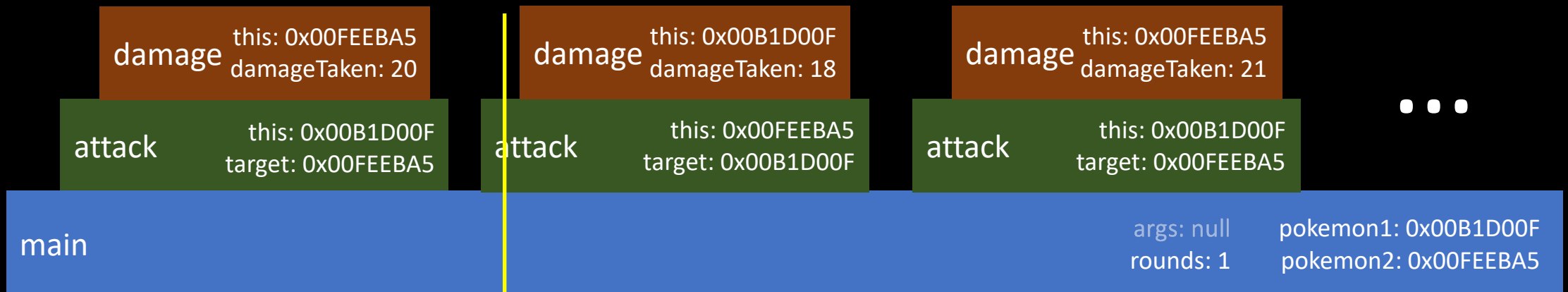


Eksempel: pokemon

```
public static void main(String[] args) {
    // ...
    public void attack(IPokemon target) {
        int damageInflicted = (int) (this.strength + this.strength / 2 * random.nextGaussian());
        if (damageInflicted < 0) {
            damageInflicted = 0;
        }

        System.out.println(this.getName() + " attacks " + target.getName());
        target.damage(damageInflicted);
        if (!target.isAlive()) {
            System.out.println(target.getName() + " is defeated by " + this.getName());
        }
    }
}
```

...
Random-objekt
17
102
82
"Feebas"
Pokemon
...
Random-objekt
19
97
97
"Bidoof"
Pokemon
...



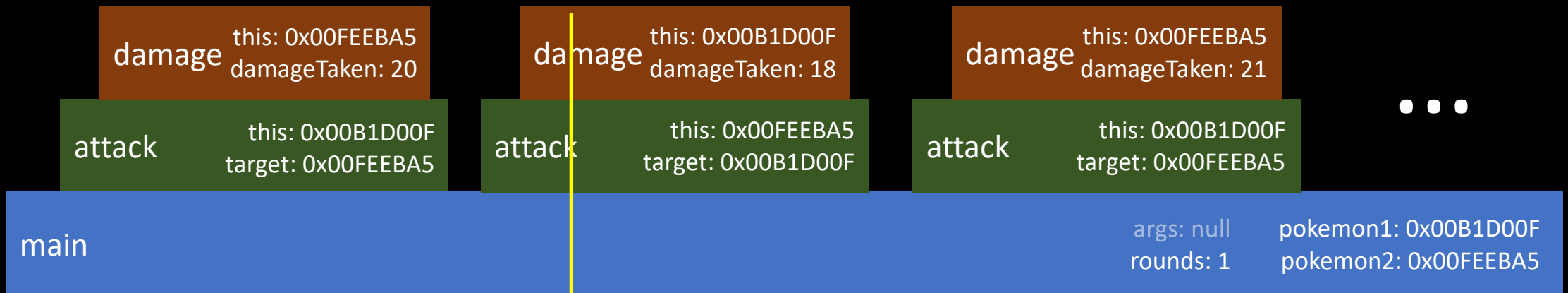
Eksempel: pokemon

```

public static void main(String[] args) {
    // ...
    public void attack(IPokemon target) {
        // ...
        public void damage(int damageTaken) {
            if (damageTaken < 0) {
                damageTaken = 0;
            }
            if (damageTaken > this.healthPoints) {
                damageTaken = healthPoints;
            }
            this.healthPoints -= damageTaken;
            System.out.println(this.name + " takes " + damageTaken + " damage and is "
                + "left with " + this.healthPoints + "/" + this.maxHealthPoints + " HP");
        }
    }
}

```

...
Random-objekt
17
102
82
"Feebas"
Pokemon
...
Random-objekt
19
97
97
"Bidoof"
Pokemon
...



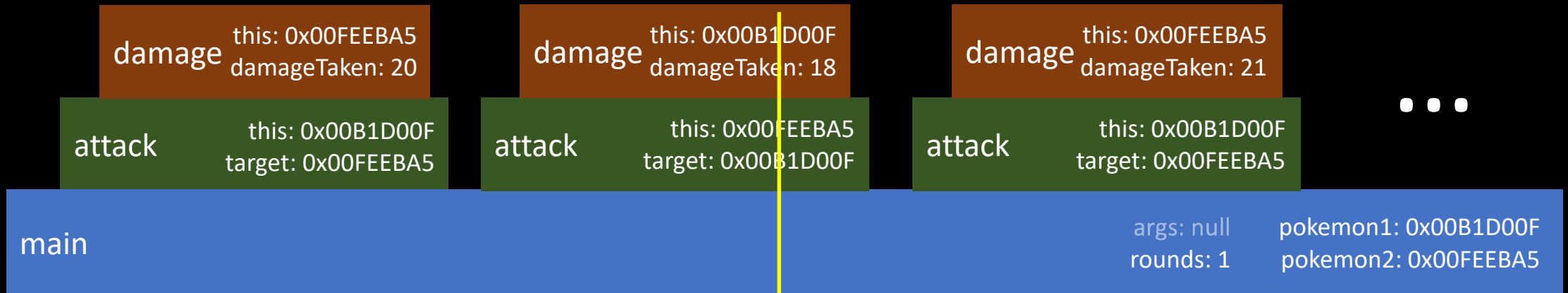
Eksempel: pokemon

```

public static void main(String[] args) {
    // ...
    public void attack(IPokemon target) {
        // ...
        public void damage(int damageTaken) {
            if (damageTaken < 0) {
                damageTaken = 0;
            }
            if (damageTaken > this.healthPoints) {
                damageTaken = healthPoints;
            }
            this.healthPoints -= damageTaken;
            System.out.println(this.name + " takes " + damageTaken + " damage and is "
                + "left with " + this.healthPoints + "/" + this.maxHealthPoints + " HP");
        }
    }
}

```

...
Random-objekt
17
102
82
"Feebas"
Pokemon
...
Random-objekt
19
97
79
"Bidoof"
Pokemon
...



Eksempel: pokemon

```

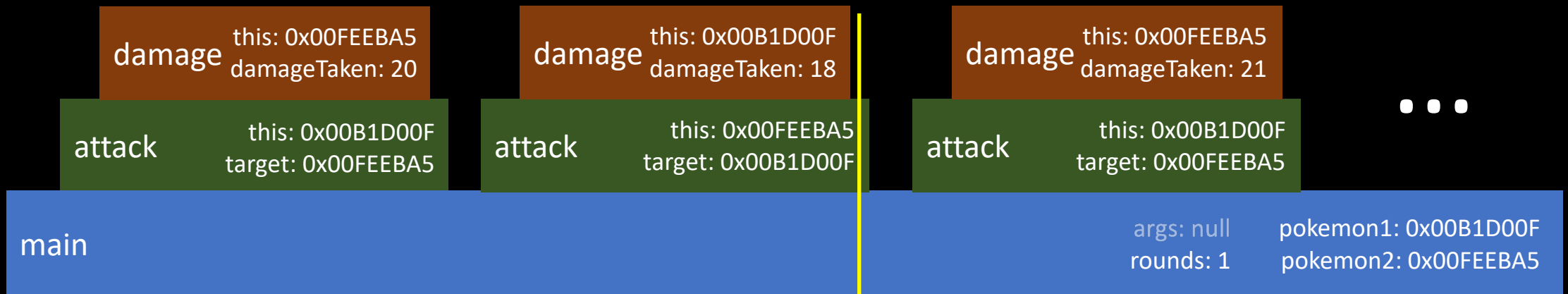
public static void main(String[] args) {
    // ...
    pokemon1.attack(pokemon2);
}

public void attack(IPokemon target) {
    int damageInflicted = (int) (this.strength + this.strength / 2 * random.nextGaussian());
    if (damageInflicted < 0) {
        damageInflicted = 0;
    }

    System.out.println(this.getName() + " attacks " + target.getName());
    target.damage(damageInflicted);
    if (!target.isAlive()) {
        System.out.println(target.getName() + " is defeated by " + this.getName());
    }
}
}

```

...
Random-objekt
17
102
82
"Feebas"
Pokemon
...
Random-objekt
19
97
79
"Bidoof"
Pokemon
...

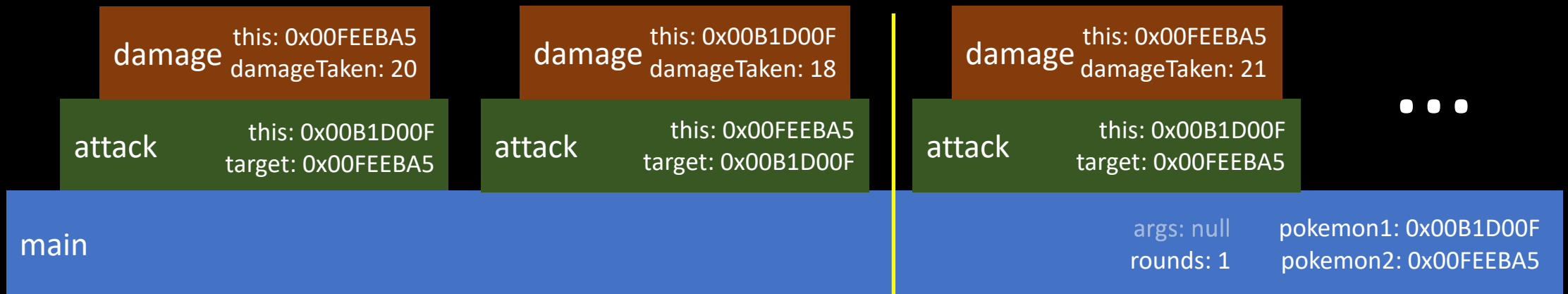


Eksempel: pokemon

```
public static void main(String[] args) {
    Ipokemon pokemon1 = new Pokemon("Pikachu");
    Ipokemon pokemon2 = new Pokemon("Oddish");
    System.out.println(pokemon1);
    System.out.println(pokemon2);

    while (pokemon1.isAlive() && pokemon2.isAlive()) {
        pokemon1.attack(pokemon2);
        if (pokemon2.isAlive()) {
            pokemon2.attack(pokemon1);
        }
    }
}
```

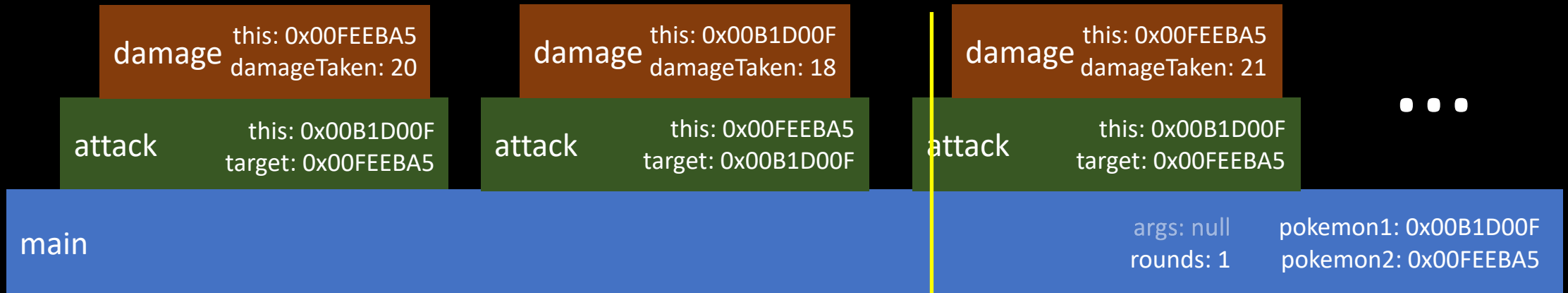
...
Random-objekt
17
102
82
"Feebas"
Pokemon
...
Random-objekt
19
97
79
"Bidoof"
Pokemon
...



Eksempel: pokemon

```
public static void main(String[] args) {  
    // ...  
    public void attack(IPokemon target) {  
        int damageInflicted = (int) (this.strength + this.strength / 2 * random.nextGaussian());  
        if (damageInflicted < 0) {  
            damageInflicted = 0;  
        }  
  
        System.out.println(this.getName() + " attacks " + target.getName());  
        target.damage(damageInflicted);  
        if (!target.isAlive()) {  
            System.out.println(target.getName() + " is defeated by " + this.getName());  
        }  
    }  
}
```

...
Random-objekt
17
102
82
"Feebas"
Pokemon
...
Random-objekt
19
97
79
"Bidoof"
Pokemon
...



Eksempel: pokemon

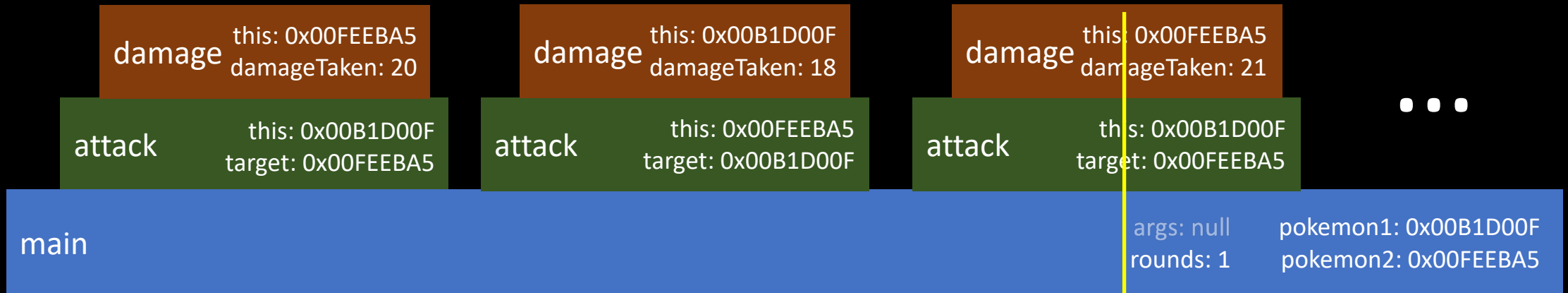
```

public static void main(String[] args) {
    Pokemon pokemon1 = new Pokemon("Bidoof", 19, 97);
    Pokemon pokemon2 = new Pokemon("Feebas", 17, 102);
    pokemon1.attack(pokemon2);
    pokemon2.attack(pokemon1);
}

public void attack(IPokemon target) {
    damage(target, 20);
}

public void damage(int damageTaken) {
    if (damageTaken < 0) {
        damageTaken = 0;
    }
    if (damageTaken > this.healthPoints) {
        damageTaken = healthPoints;
    }
    this.healthPoints -= damageTaken;
    System.out.println(this.name + " takes " + damageTaken + " damage and is "
        + "left with " + this.healthPoints + "/" + this.maxHealthPoints + " HP");
}
    
```

...
Random-objekt
17
102
82
"Feebas"
Pokemon
...
Random-objekt
19
97
79
"Bidoof"
Pokemon
...



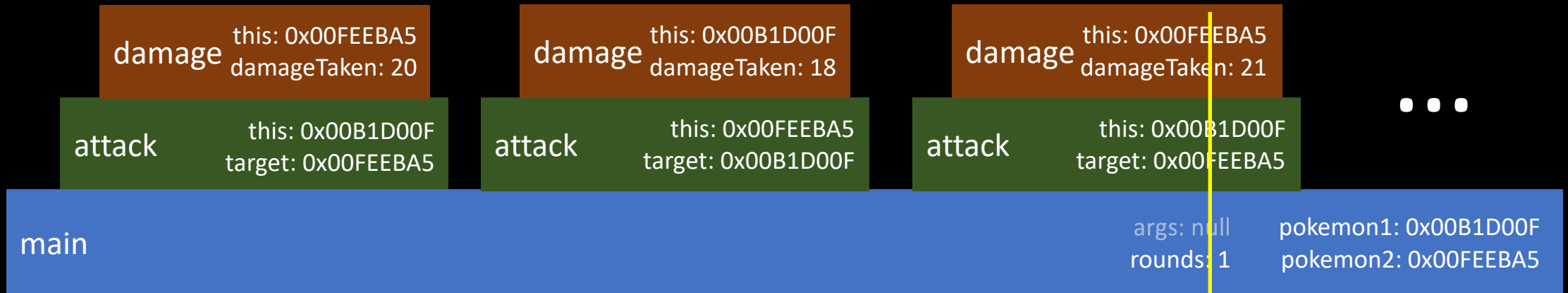
Eksempel: pokemon

```

public static void main(String[] args) {
    // ...
    public void attack(IPokemon target) {
        // ...
        public void damage(int damageTaken) {
            if (damageTaken < 0) {
                damageTaken = 0;
            }
            if (damageTaken > this.healthPoints) {
                damageTaken = healthPoints;
            }
            this.healthPoints -= damageTaken;
            System.out.println(this.name + " takes " + damageTaken + " damage and is "
                + "left with " + this.healthPoints + "/" + this.maxHealthPoints + " HP");
        }
    }
}

```

...
Random-objekt
17
102
61
"Feebas"
Pokemon
...
Random-objekt
19
97
79
"Bidoof"
Pokemon
...



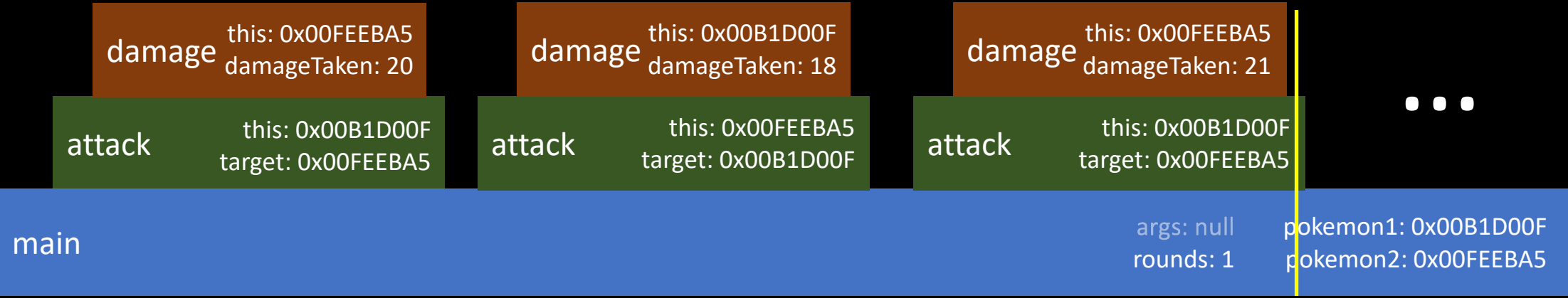
Eksempel: pokemon

```

public static void main(String[] args) {
    // ...
    public void attack(IPokemon target) {
        int damageInflicted = (int) (this.strength + this.strength / 2 * random.nextGaussian());
        if (damageInflicted < 0) {
            damageInflicted = 0;
        }

        System.out.println(this.getName() + " attacks " + target.getName());
        target.damage(damageInflicted);
        if (!target.isAlive()) {
            System.out.println(target.getName() + " is defeated by " + this.getName());
        }
    }
}
    
```

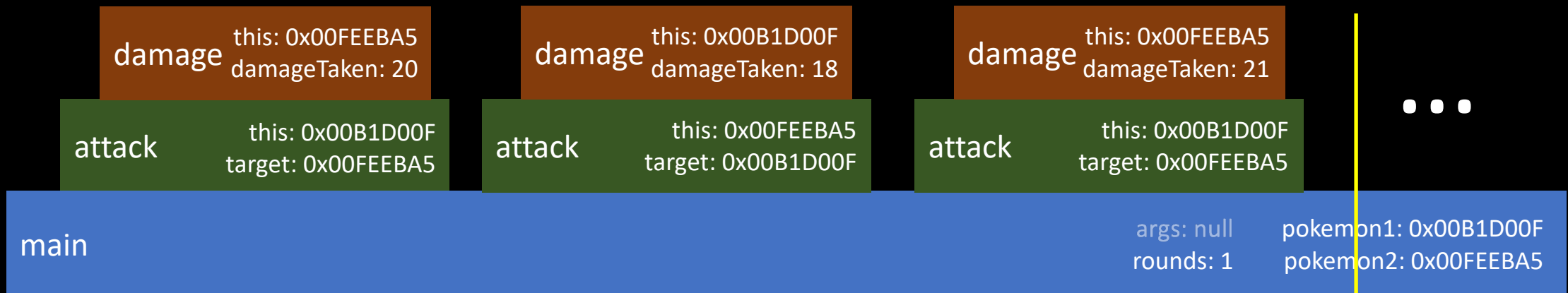
...
Random-objekt
17
102
61
"Feebas"
Pokemon
...
Random-objekt
19
97
79
"Bidoof"
Pokemon
...



Eksempel: pokemon

```
public static void main(String[] args) {  
    Ipokemon pokemon1 = new Pokemon("Pikachu");  
    Ipokemon pokemon2 = new Pokemon("Oddish");  
    System.out.println(pokemon1);  
    System.out.println(pokemon2);  
  
    while (pokemon1.isAlive() && pokemon2.isAlive()) {  
        pokemon1.attack(pokemon2);  
        if (pokemon2.isAlive()) {  
            pokemon2.attack(pokemon1);  
        }  
    }  
}
```

...
Random-objekt
17
102
61
"Feebas"
Pokemon
...
Random-objekt
19
97
79
"Bidoof"
Pokemon
...



Debugger

```
12  public static void main(String[] args) {
13      IPokemon pokemon1 = new Pokemon(name: "Bidoof");
14      IPokemon pokemon2 = new Pokemon(name: "Feebas");
15      System.out.println(pokemon1);
16      System.out.println(pokemon2);
17      int rounds = 0;
18
19      while (pokemon1.isAlive() && pokemon2.isAlive()) {
20          pokemon1.attack(pokemon2);
21          if (pokemon2.isAlive()) {
22              pokemon2.attack(pokemon1);
23          }
24          rounds++;
25      }
26      System.out.print(rounds + " rounds were played");
27  }
28 }
```

Steg 1 – sett et breakpoint:
klikk i marginen ved linjenummeret hvor
du ønsker å begynne debuggen

```
11  |
12  |
13  | Run | Debug
14  | public static void main(String[] args) {
15  |     IPokemon pokemon1 = new Pokemon(name: "Bidoof");
16  |     IPokemon pokemon2 = new Pokemon(name: "Feebas");
17  |     System.out.println(pokemon1);
18  |     System.out.println(pokemon2);
19  |     int rounds = 0;
```

Steg 2 – start debugger:
kjør programmet i debug-modus

```
src > main > java > INF101 > lab2 > J
12  | public static void main(String[] args) { args = String[0]@8
13  |     IPokemon pokemon1 = new Pokemon(name: "Bidoof"); pokemon1
14  |     IPokemon pokemon2 = new Pokemon(name: "Feebas");
15  |     System.out.println(pokemon1);
16  |     System.out.println(pokemon2);
17  |     int rounds = 0;
18  | }
```

Steg 3 – gå steg for steg:
klikk på pilen som hopper over en
prikk for å gå til neste linje

Debugger: navigasjon

Gå inn i metodekallet

Gjør ferdig kun denne metoden

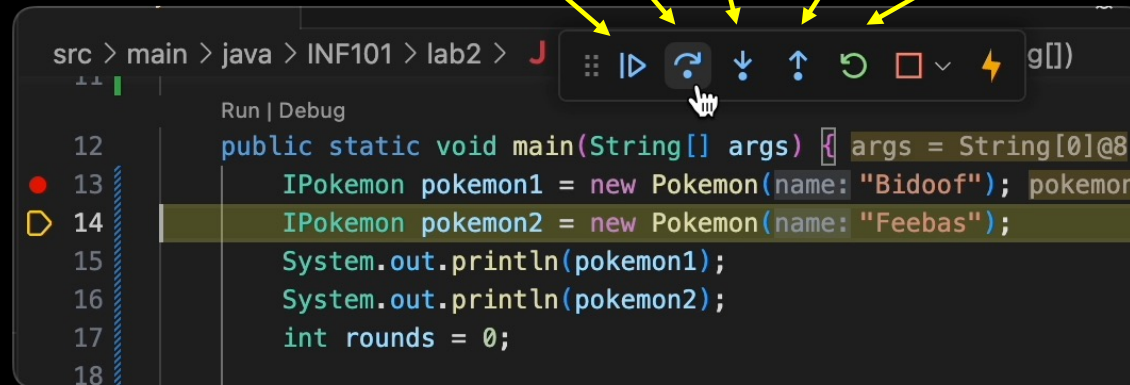
Gå til neste linje

Fortsett til neste breakpoint

Start på nytt

Breakpoint

Neste linje som vil utføres



```
src > main > java > INF101 > lab2 > J
Run | Debug
12 public static void main(String[] args) { args = String[0]@8
13     IPokemon pokemon1 = new Pokemon(name: "Bidoof"); pokemon1
14     IPokemon pokemon2 = new Pokemon(name: "Feebas");
15     System.out.println(pokemon1);
16     System.out.println(pokemon2);
17     int rounds = 0;
18 }
```

Debugger: inspeksjon

Se verdier av variabler

Se på variabler (og endre på variabler)

Se verdien av uttrykk du selv angir

Se hvorfor du er her (og velg hvilken del av stacken du ser på)

Slå av og på breakpoint

The screenshot shows an IDE with a Java debugger. The left sidebar contains several panels: **VARIABLES** (Local: damageTaken: 15, this: Pokemon@30), **WATCH** (damageTaken < 0: false, damageTaken > this.healthPoints: false, this.healthPoints - damageTaken: 66), **CALL STACK** (Thread [main] - PAUSED ON STEP: Pokemon.damage(int) at line 70:1, Pokemon.attack(IPokemon) at line 55:1, Main.main(String[]) at line 20:1), and **BREAKPOINTS** (Uncaught Exceptions, Caught Exceptions). The main editor shows the `damage(int)` method in `Pokemon.java` with line 70 highlighted: `this.healthPoints -= damageTaken; healthPoints = 81, damageTaken = 15`. The bottom right shows the **TERMINAL** with the following output: `torsteins@mbp21 lab2 % /usr/bin/env /Users/torsteins/.vscode/extensions/redhat.java-1.14.0-darwin-arm64/jre/17.0.5-macosx-aarch64/bin/java -agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:60645 -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/torsteins/Library/Mobile Documents/com~apple~CloudDocs/Documents/UiB/INF101/23v/lab2/target/classes INF101.lab2.Main Bidoo HP: (97/97) STR: 7 Feebas HP: (81/81) STR: 18 Bidoo attacks Feebas`

Likhet i egne klasser

```
public class Person {  
    String name;  
    String id;  
  
    public Person(String name, String id) {  
        this.name = name;  
        this.id = id;  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Person p1 = new Person("Ola", "111111 12345");  
        Person p2 = new Person("Ola", "111111 12345");  
        System.out.println(p1.equals(p2));  
    }  
}
```

← false

Standard equals -metode

```
public boolean equals(Object obj) {  
    return (this == obj);  
}
```

- Arvet fra *Object*-klassen

Vår egen equals-metode: første forsøk

```
public boolean equals(Person other) {  
    return this.name.equals(other.name) && this.id.equals(other.id);  
}
```

- Fungerer tilsynelatende
- Men hva hvis klient-koden bruker et grensesnitt som type?

Vår egen equals-metode: bedre forsøk

Viktig!

Casting

```
@Override
public boolean equals(Object other) {
    if (other instanceof Person) {
        Person otherPerson = (Person) other;
        return this.name.equals(otherPerson.name) && this.id.equals(otherPerson.id);
    }
    return false;
}
```


Best: autogenerert

```
@Override  
public int hashCode() {  
    return Objects.hash(name, id);  
}
```

```
@Override  
public boolean equals(Object o) {  
    if (o == this)  
        return true;  
    if (!(o instanceof Person)) {  
        return false;  
    }  
    Person person = (Person) o;  
    return Objects.equals(name, person.name) && Objects.equals(id, person.id);  
}
```

```
ic class Person implements IPerson
```

```
ring name;  
ring id;
```

```
ublic Person  
this.name =  
this.id = i
```

```
verride  
ublic void g  
System.out.
```

- Go to Definition F12
- Go to Type Definition
- Go to Implementations ⌘ F12
- Go to References ⇧ F12
- Go to Super Implementation
- Go to Test
- Peek >
- Find All References ⌘ ⇧ F12
- Find All Implementations
- Show Call Hierarchy ⌘ ⇧ H
- Show Type Hierarchy
- Rename Symbol F2
- Change All Occurrences ⌘ F2
- Format Document ⌘ ⇧ F
- Format Document With...
- Refactor... ^ ⇧ R
- Source Action...
- Cut ⌘ X
- Copy ⌘ C
- Paste ⌘ V
- Add to Watch
- Run Java
- Debug Java
- Generate Everything
- Generator GUI
- Java Code Generators >
- Command Palette... ⇧ ⌘ P

- Generate Empty Constructor
- Generate Constructor Using Only Final Fields
- Generate Constructor Using All Fields
- Generate Logger Debug of any selected text
- Generate Getters And Setters
- Generate Only Fluent Setters
- Generate hashCode And equals**
- Generate toString