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

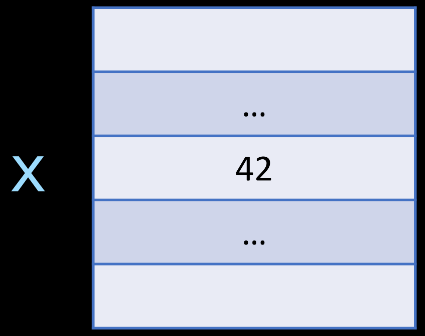
- Alle typer som ikke er primitive
    - `String`
    - `Integer, Double, Boolean`
    - `int[], double[], String[]`
    - `ArrayList`
    - `LocalDateTime`
    - `IPokemon, Pokemon`
    - `CommandLineInterface, Command`
    - `DummyShell, EchoShell, SimpleShell`
    - `...`
- Verdier som ikke har en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



x

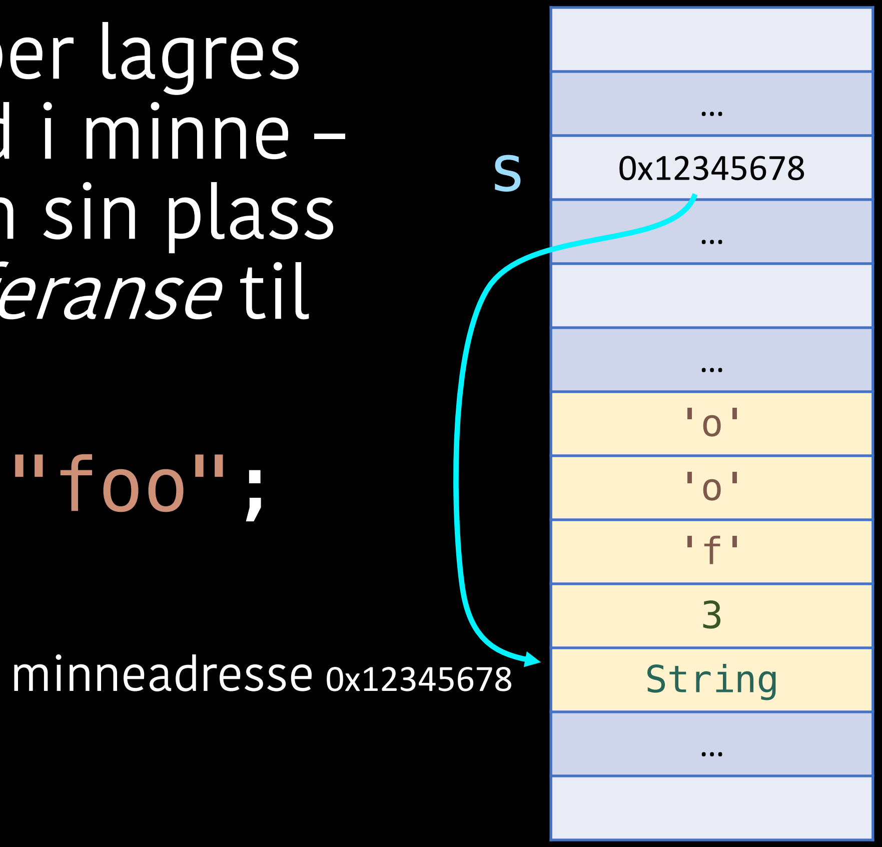
…

42

…

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

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



s

…

0x12345678

…

…

'o'

'o'

'f'

3

String

…

minneadresse 0x12345678

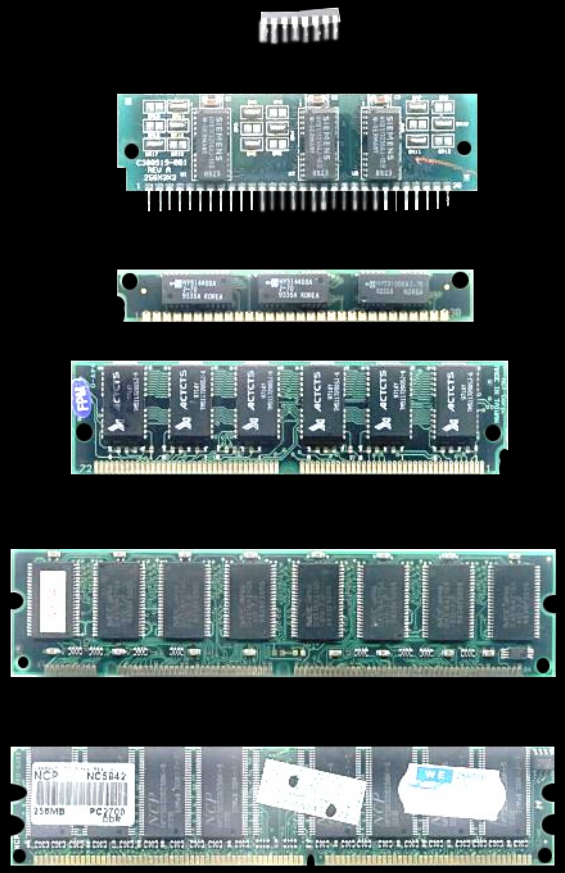
Likhet

https://inf101.ii.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.



Illustrasjon: CC-BY-SA ArachanoxReal via Wikimedia Commons

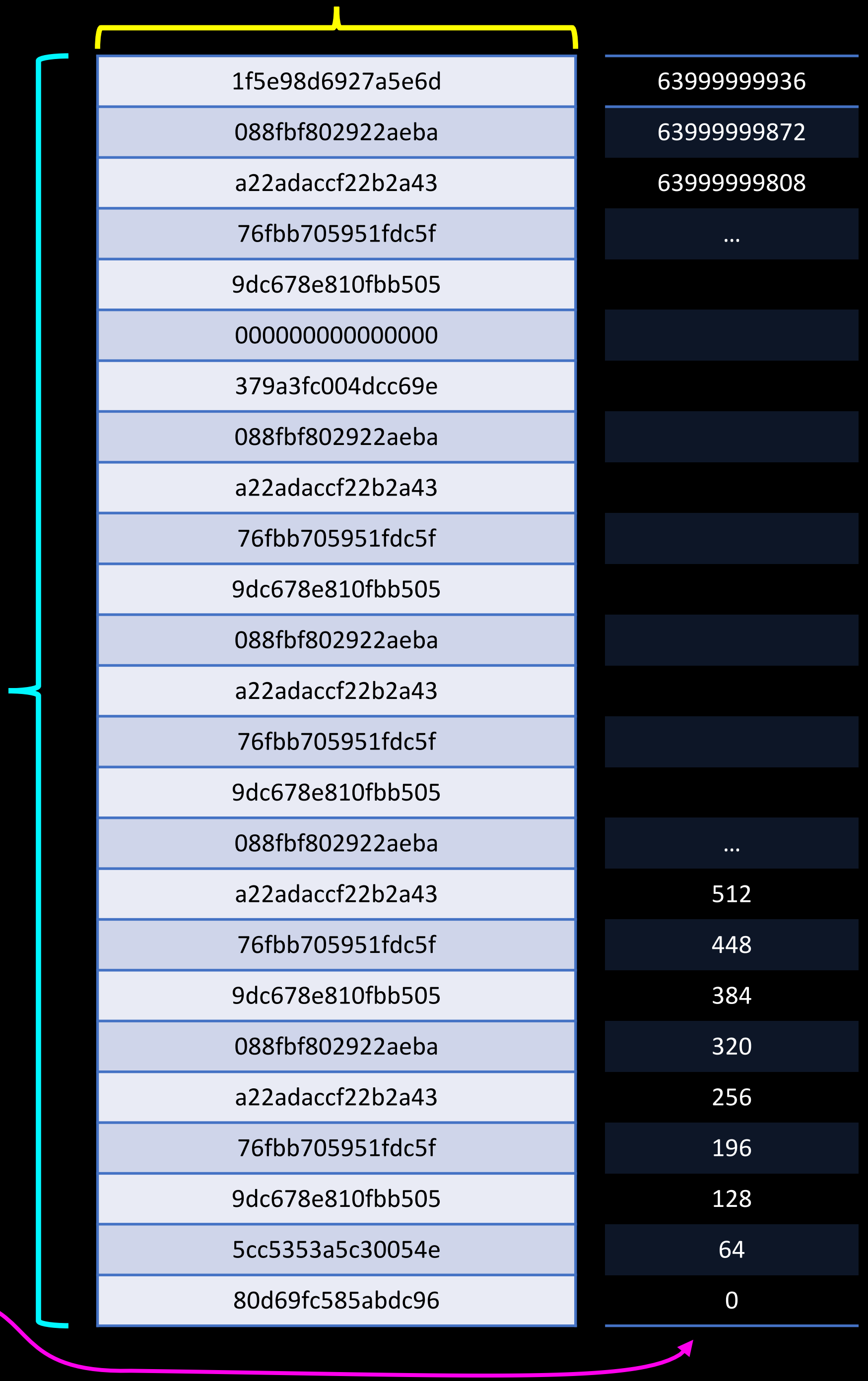
Minne

- 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)

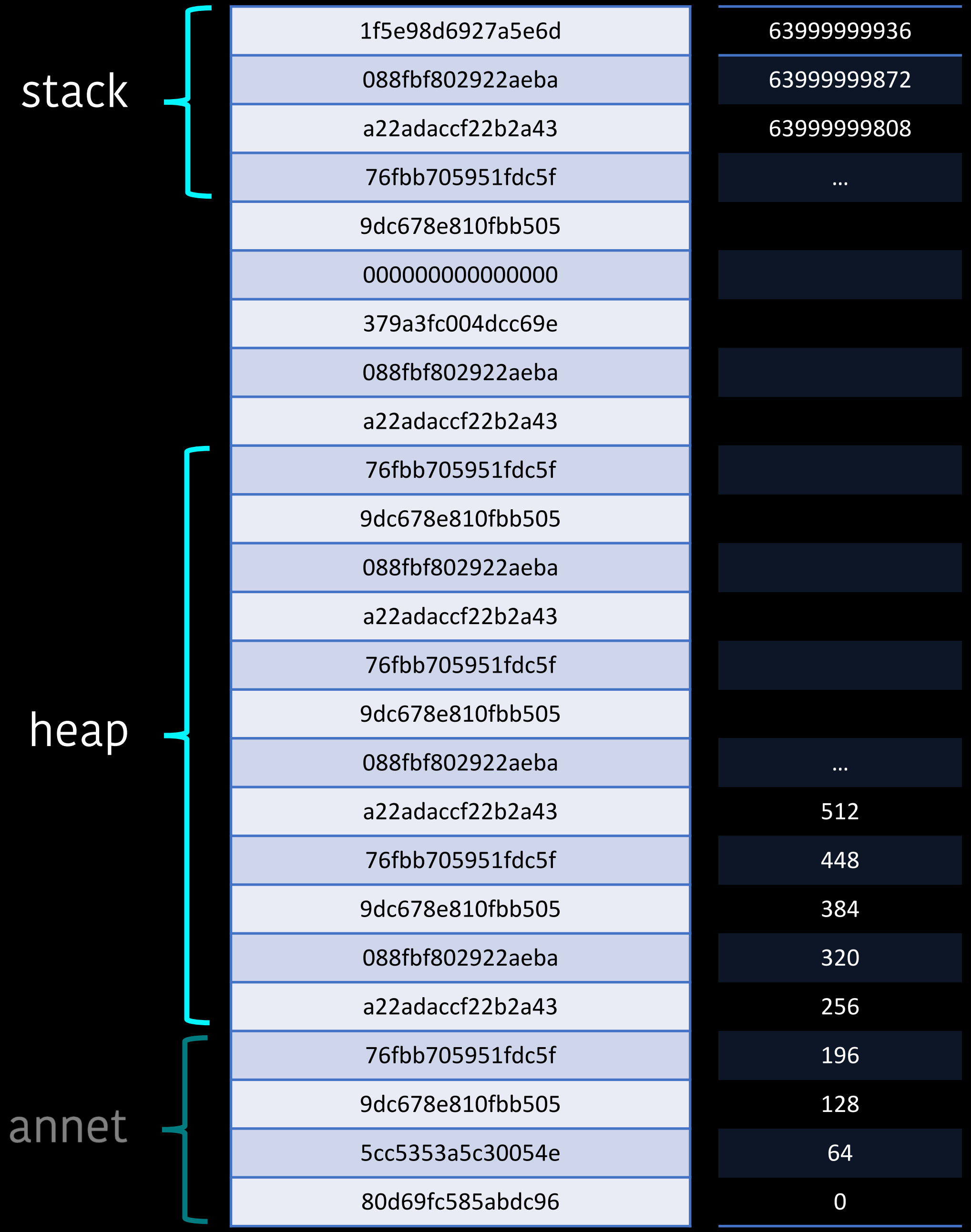
Minneadresse til celle: posisjonen til første bit i cellen



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



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

Objekter i minnet

https://inf101.ii.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

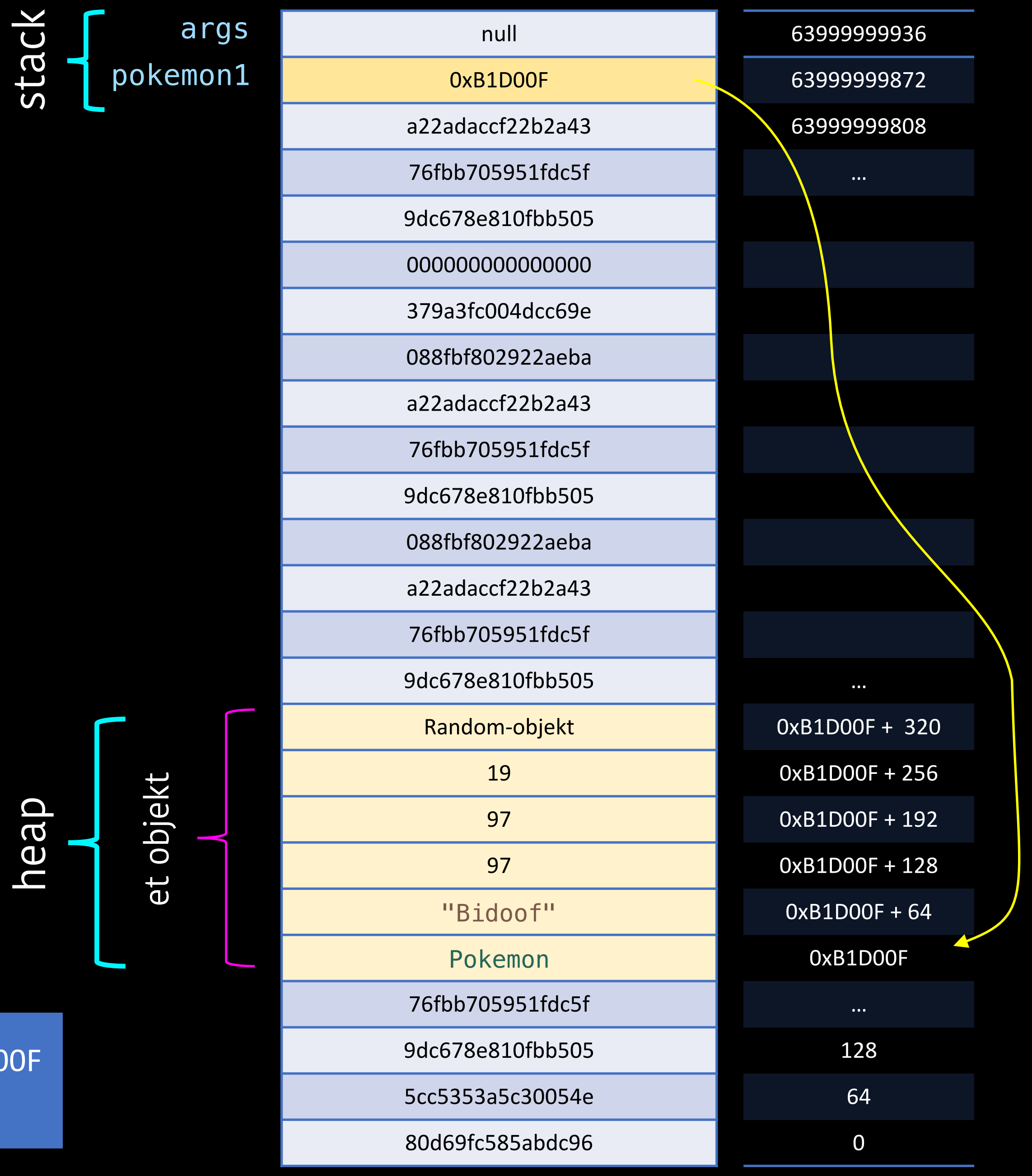
stack

args	null	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

Eksempel: pokemon

```java
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");
}
```

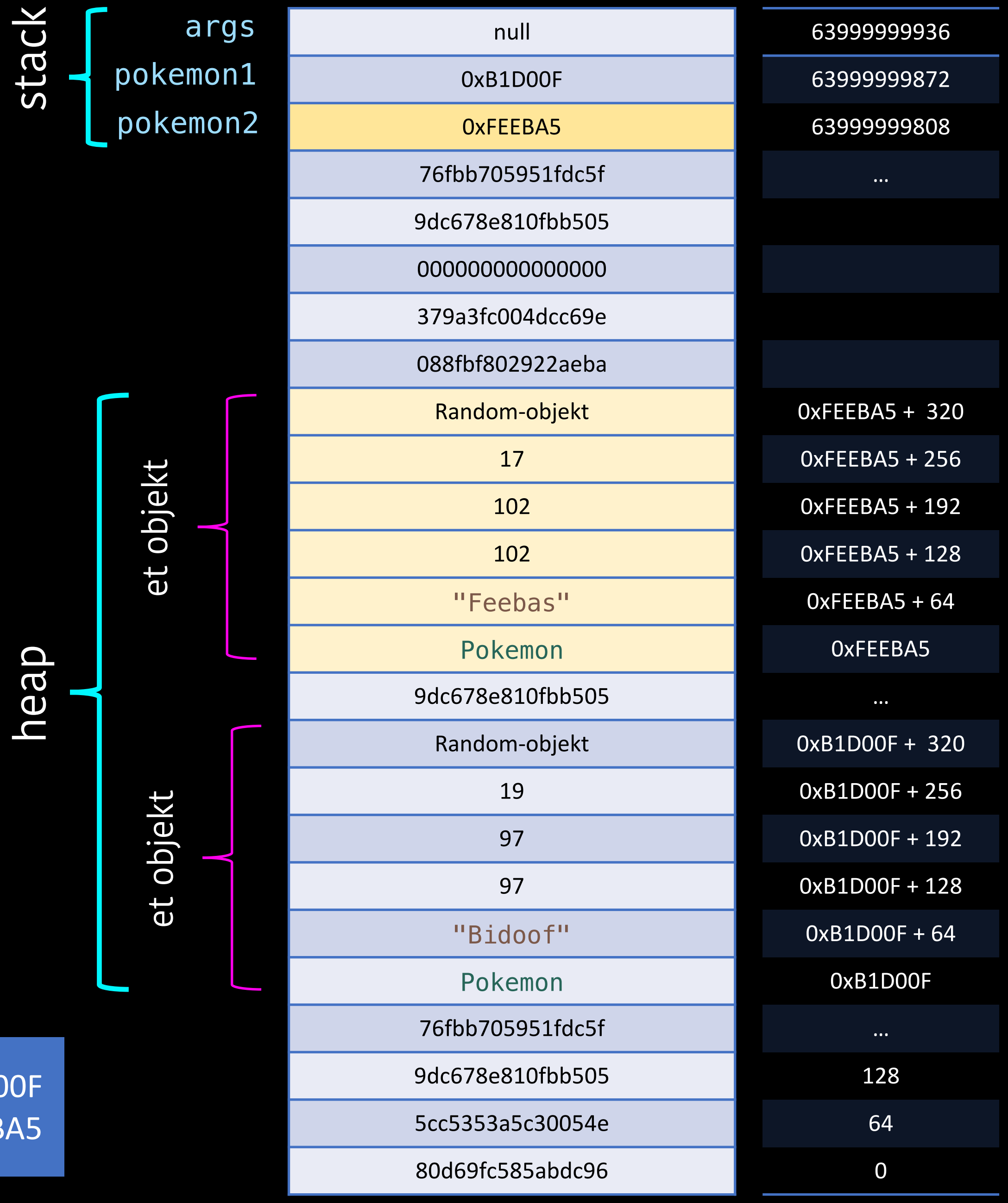
main | args: null | pokemon1: 0x00B1D00F



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");
}
```



args	null	63999999936
pokemon1	0xB1D00F	63999999872
pokemon2	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

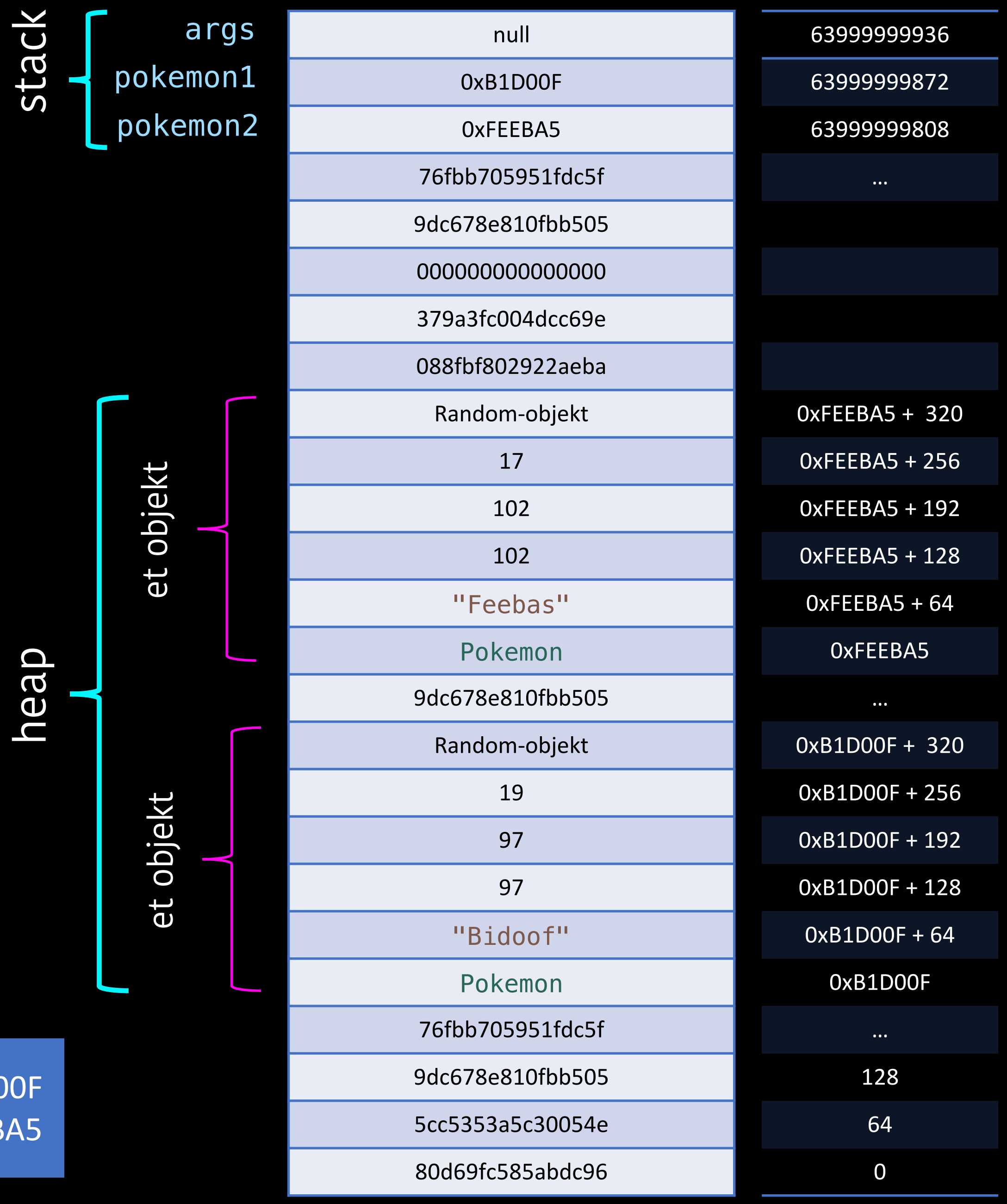
main

args: null

pokemon1: 0x00B1D00F
pokemon2: 0x00FEEBA5

Eksempel: pokemon

```java
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");
}
```



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

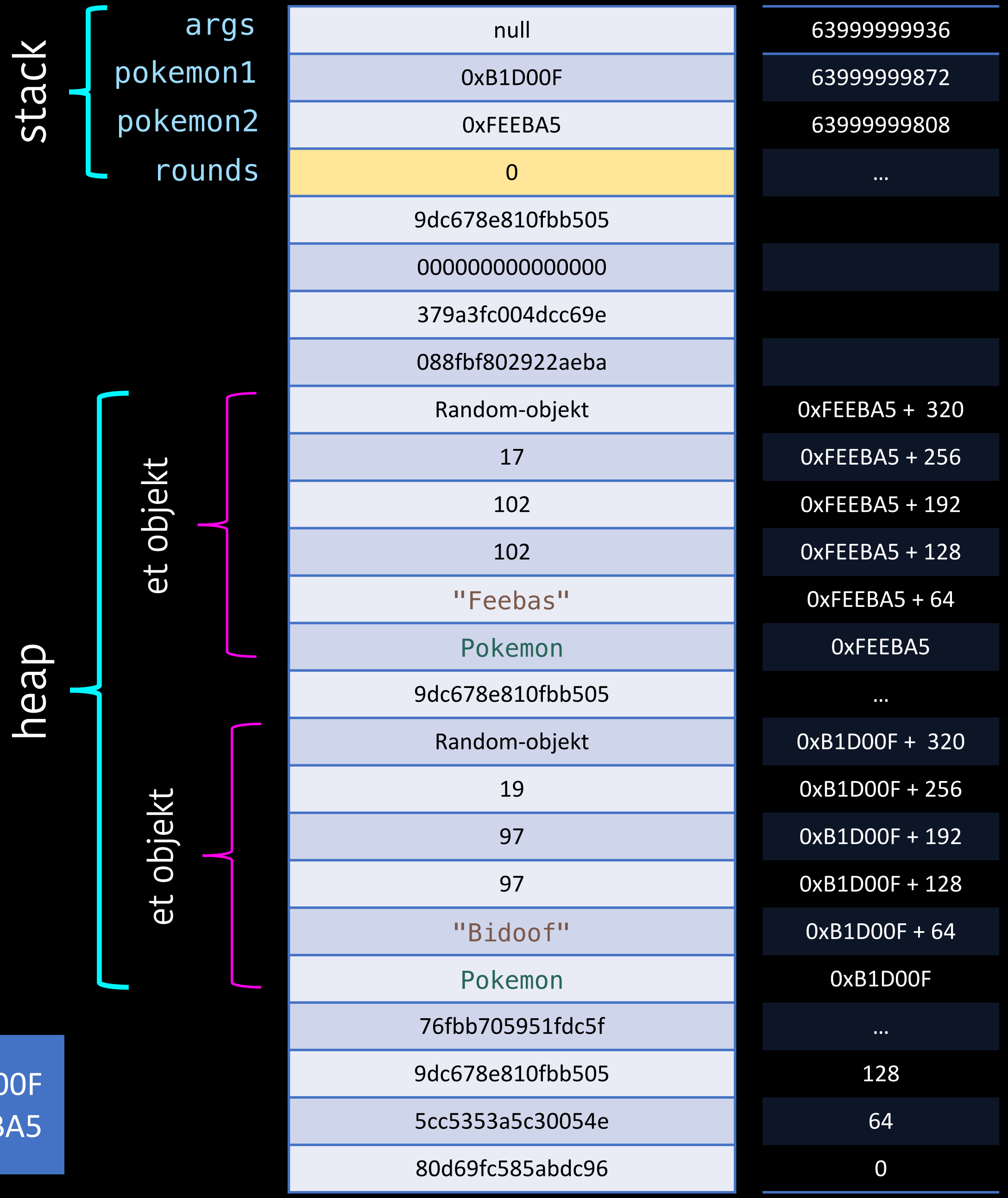
main

args: null

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");
}
```



stack

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

main

args: null
rounds: 0

pokemon1: 0x00B1D00F
pokemon2: 0x00FEEBA5

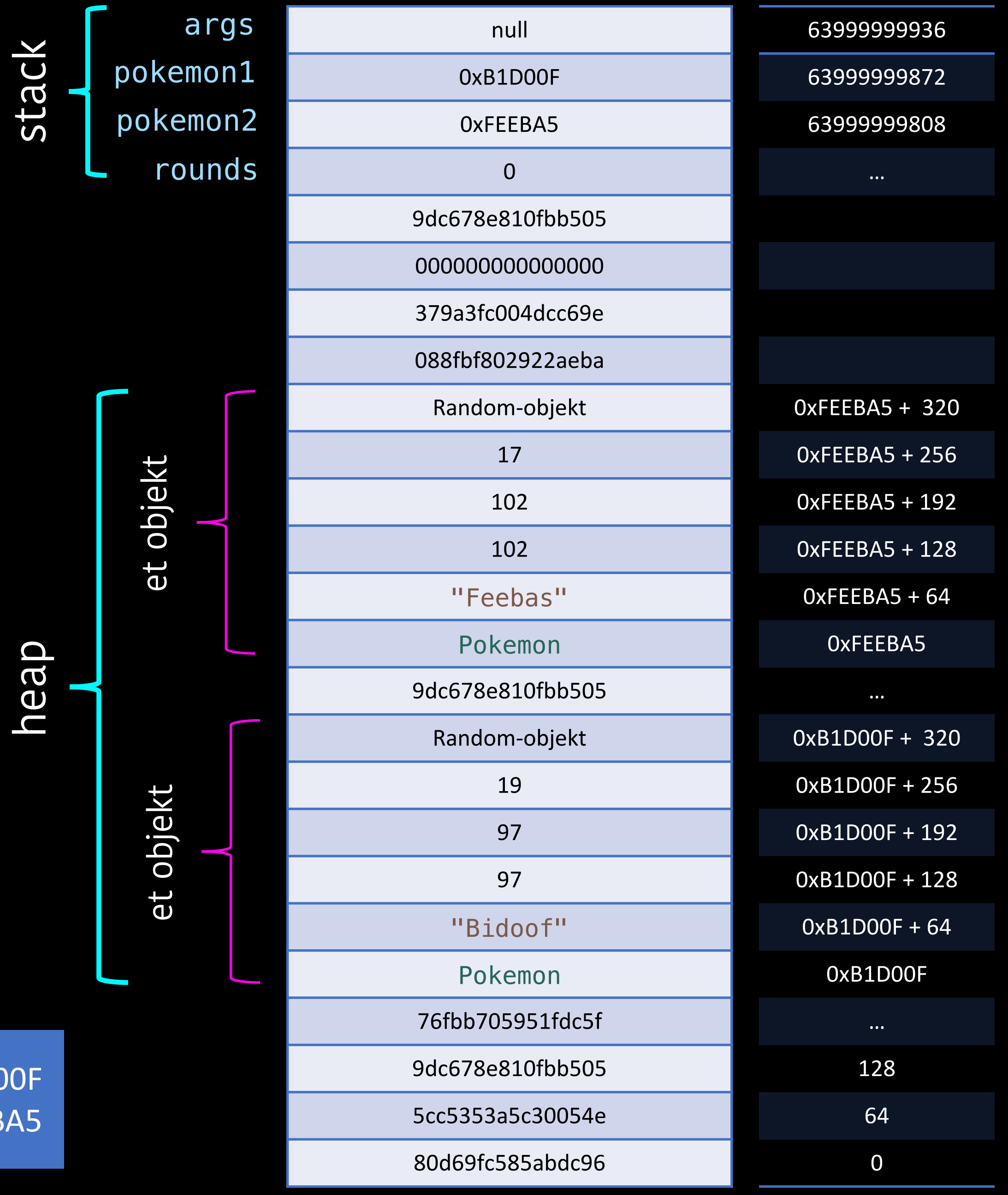
Eksempel: pokemon

```java
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



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

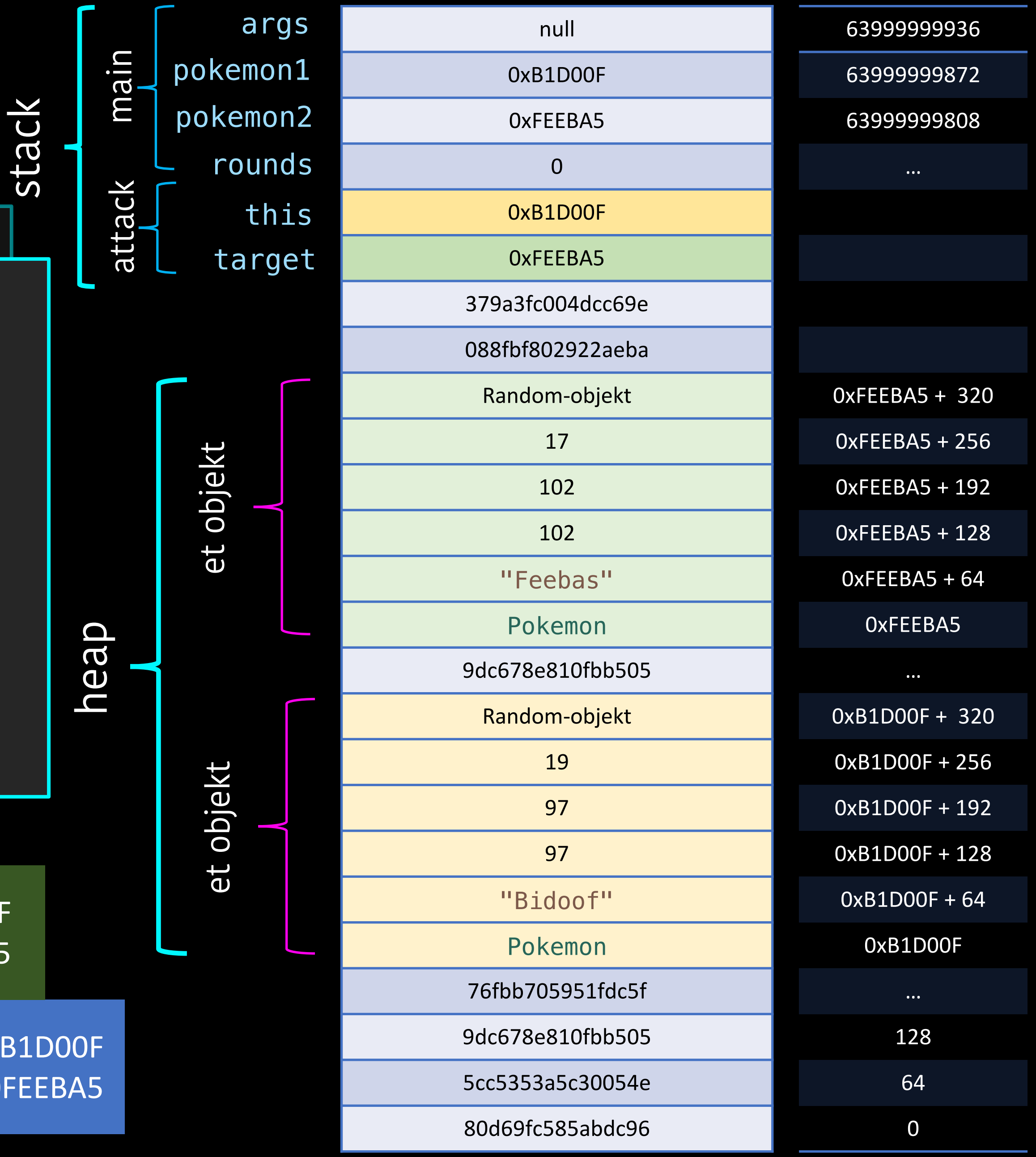
Eksempel: pokemon

```
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());
  }
}
```

attack this: 0x00B1D00F
target: 0x00FEEBA5

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

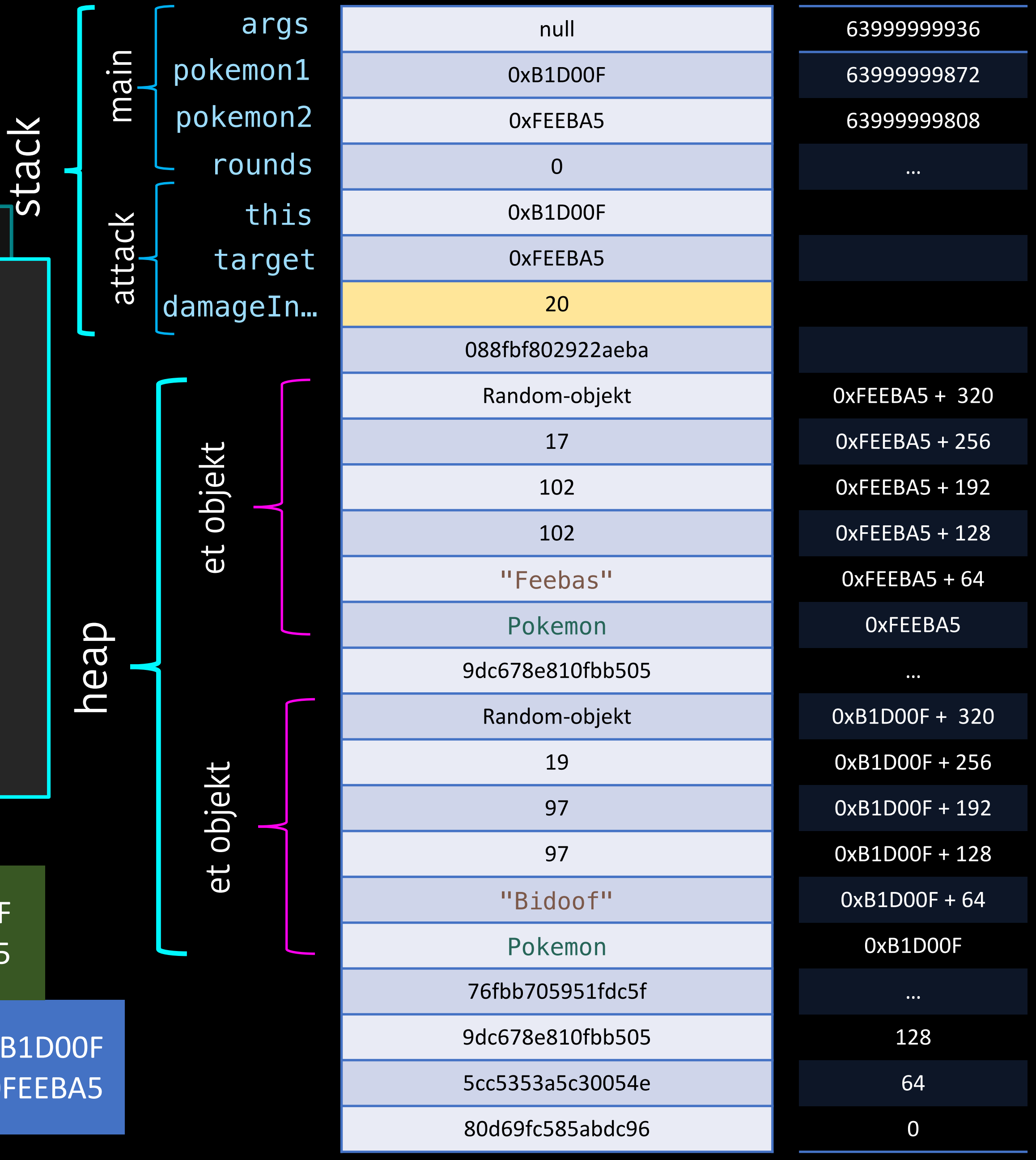


args	null	63999999936
pokemon1	0xB1D00F	63999999872
pokemon2	0xFEEBA5	63999999808
rounds	0	...
this	0xB1D00F	
target	0xFEEBA5	
	379a3fc004dcc69e	
	088fbf802922aeba	
	Random-objekt	0xFEEBA5 + 320
	17	0xFEEBA5 + 256
	102	0xFEEBA5 + 192
	102	0xFEEBA5 + 128
	"Feebas"	0xFEEBA5 + 64
	Pokemon	0xFEEBA5
	9dc678e810fbb505	...
	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());
  }
}
```

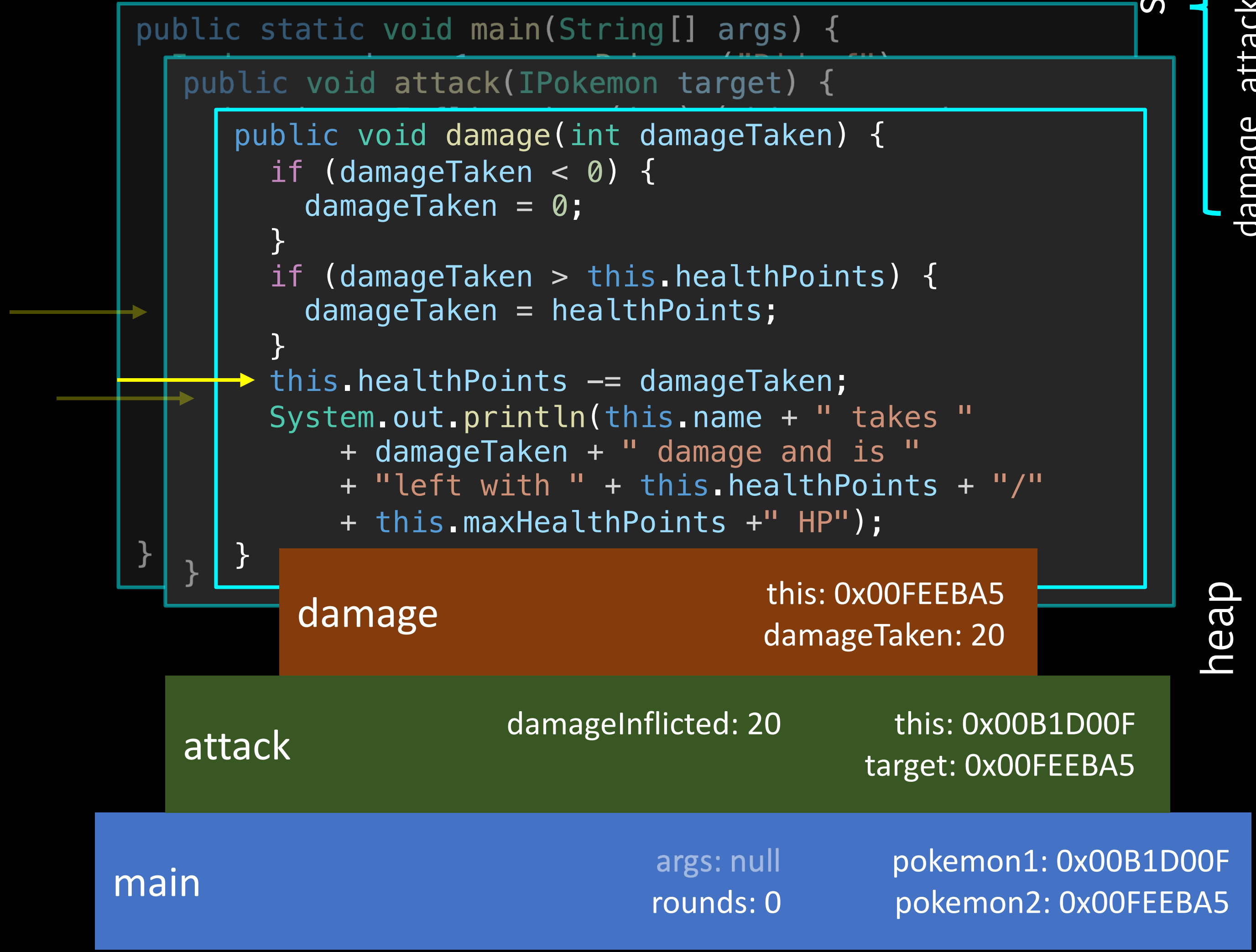


stack	main	args	null	63999999936
		pokemon1	0xB1D00F	63999999872
		pokemon2	0xFEEBA5	63999999808
		rounds	0	...
	attack	this	0xB1D00F	
		target	0xFEEBA5	
		damageIn…	20	
			088fbf802922aeba	
heap	et objekt		Random-objekt	0xFEEBA5 + 320
			17	0xFEEBA5 + 256
			102	0xFEEBA5 + 192
			102	0xFEEBA5 + 128
			"Feebas"	0xFEEBA5 + 64
			Pokemon	0xFEEBA5
			9dc678e810fbb505	...
	et objekt		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

attack — damageInflicted: 20 — 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");
}
```

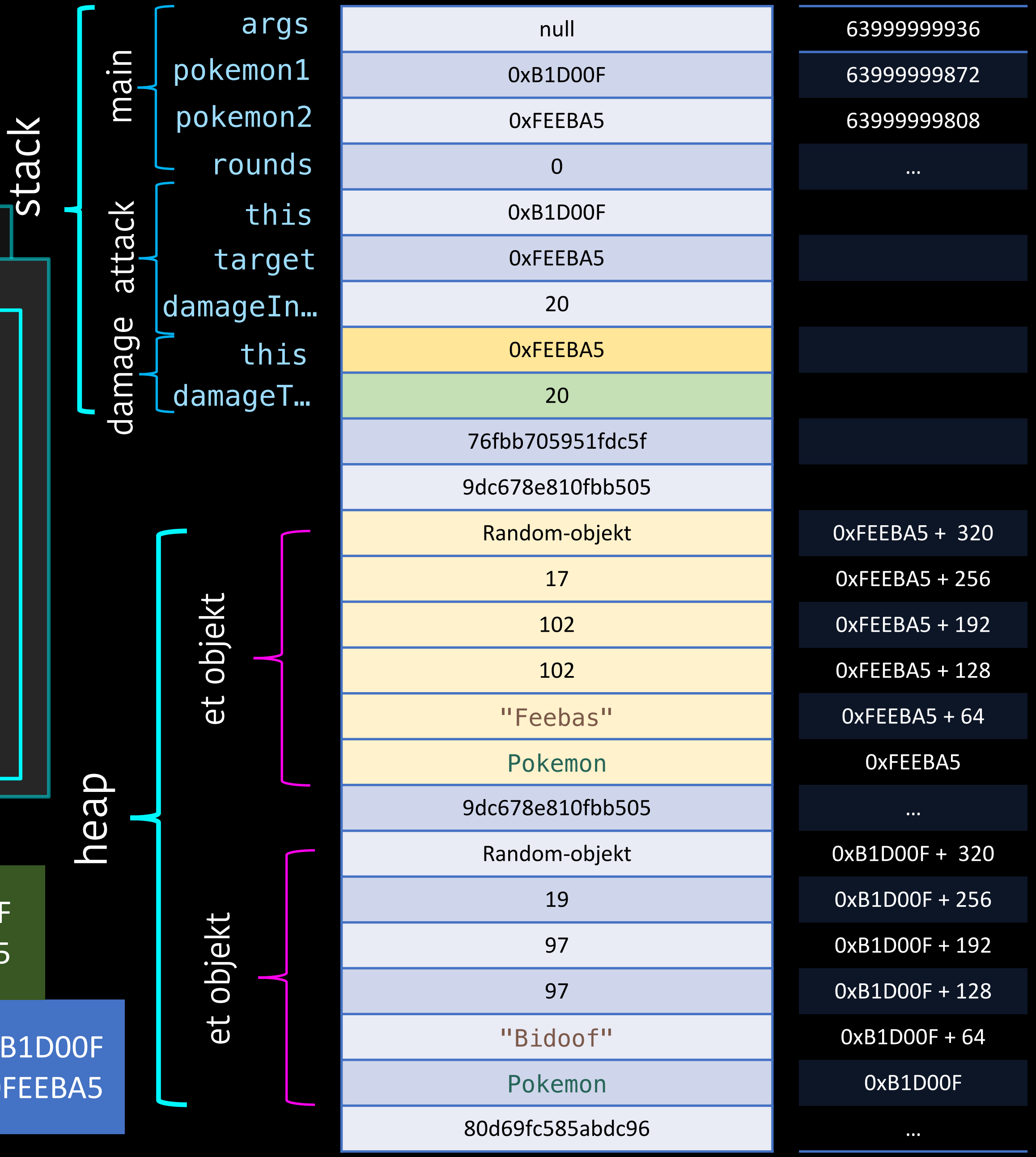
```
}
```

```
}
```

damage — this: 0x00FEEBA5, damageTaken: 20

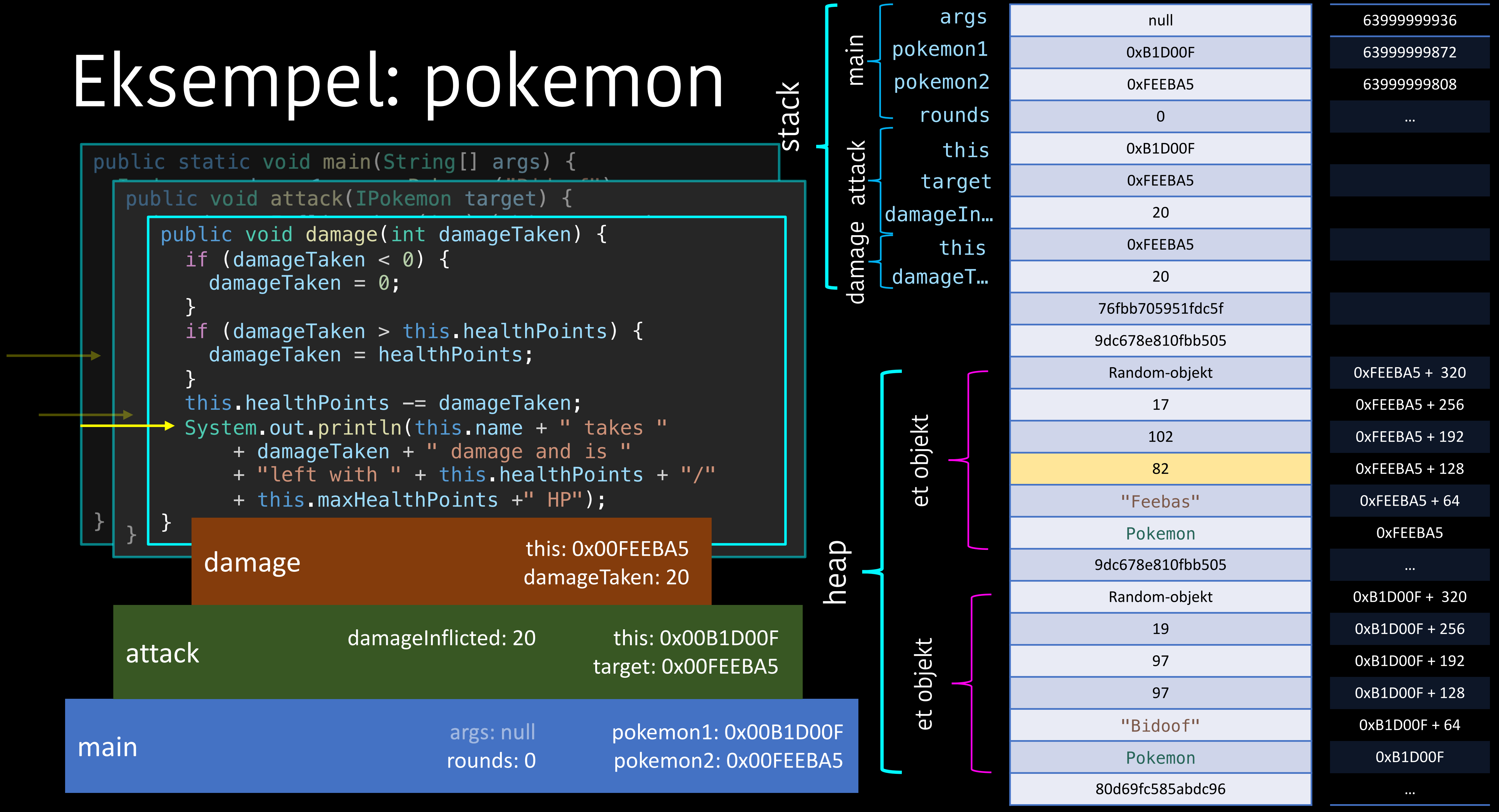
attack — damageInflicted: 20, this: 0x00B1D00F, target: 0x00FEEBA5

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



stack	main	args	null	63999999936
		pokemon1	0xB1D00F	63999999872
		pokemon2	0xFEEBA5	63999999808
		rounds	0	…
	attack	this	0xB1D00F	
		target	0xFEEBA5	
		damageIn…	20	
	damage	this	0xFEEBA5	
		damageT…	20	
			76fbb705951fdc5f	
			9dc678e810fbb505	
heap	et objekt		Random-objekt	0xFEEBA5 + 320
			17	0xFEEBA5 + 256
			102	0xFEEBA5 + 192
			102	0xFEEBA5 + 128
			"Feebas"	0xFEEBA5 + 64
			Pokemon	0xFEEBA5
			9dc678e810fbb505	…
	et objekt		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) {
```

```
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");
}
```

```
}
```

```
}
```

damage — this: 0x00FEEBA5, damageTaken: 20

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

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

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

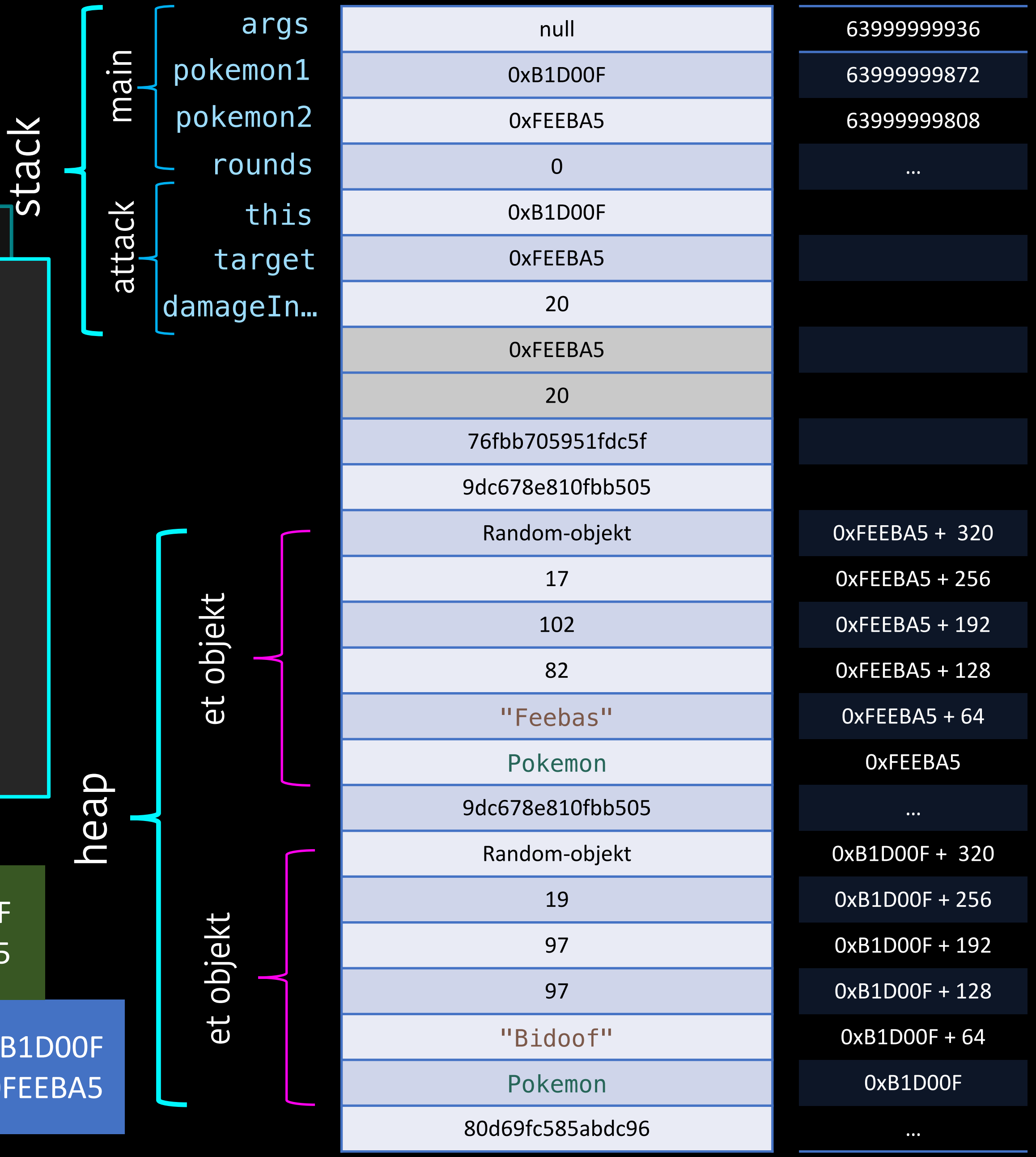
Eksempel: pokemon

```
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());
  }
}
```

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

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



args	null	63999999936
pokemon1	0xB1D00F	63999999872
pokemon2	0xFEEBA5	63999999808
rounds	0	...
this	0xB1D00F	
target	0xFEEBA5	
damageIn…	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	...

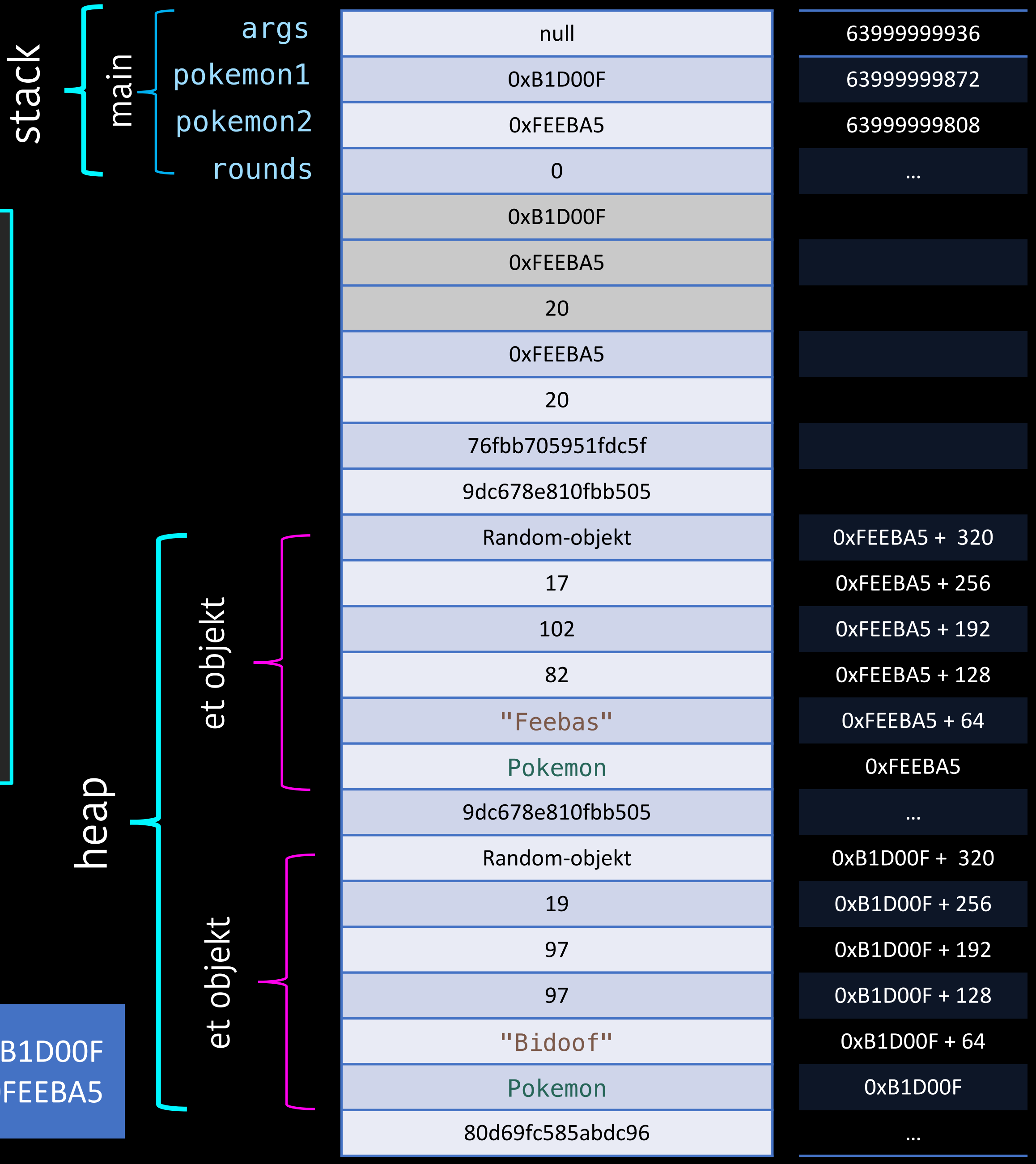
Eksempel: pokemon

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

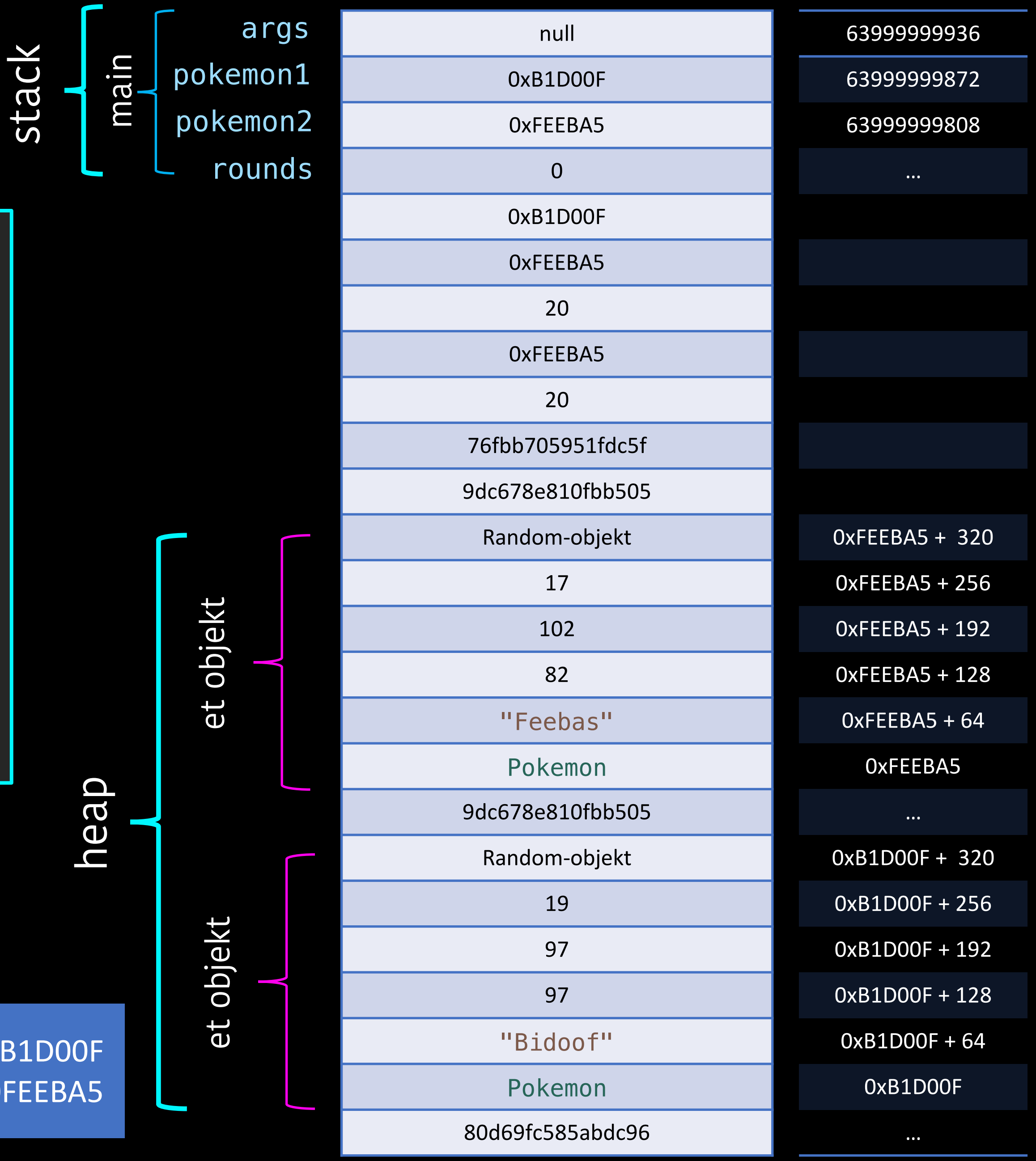
Eksempel: pokemon

```java
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

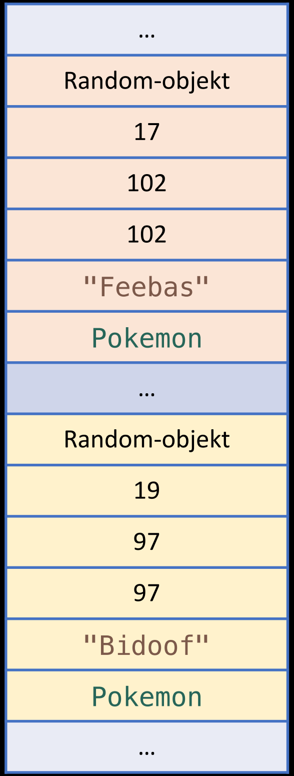


args	null	63999999936
pokemon1	0xB1D00F	63999999872
pokemon2	0xFEEBA5	63999999808
rounds	0	...
	0xB1D00F	
	0xFEEBA5	
	20	
	0xFEEBA5	
	20	
	76fbb705951fdc5f	
	9dc678e810fbb505	
	Random-objekt	0xFEEBA5 + 320
	17	0xFEEBA5 + 256
	102	0xFEEBA5 + 192
	82	0xFEEBA5 + 128
	"Feebas"	0xFEEBA5 + 64
	Pokemon	0xFEEBA5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

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

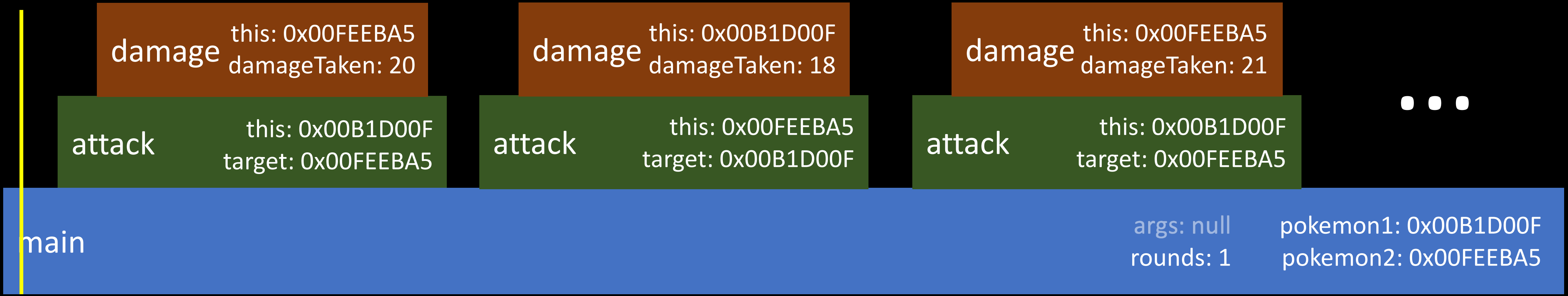
97

97

"Bidoof"

Pokemon

…



damage this: 0x00FEEBA5 damageTaken: 20

attack this: 0x00B1D00F target: 0x00FEEBA5

damage this: 0x00B1D00F damageTaken: 18

attack this: 0x00FEEBA5 target: 0x00B1D00F

damage this: 0x00FEEBA5 damageTaken: 21

attack this: 0x00B1D00F target: 0x00FEEBA5

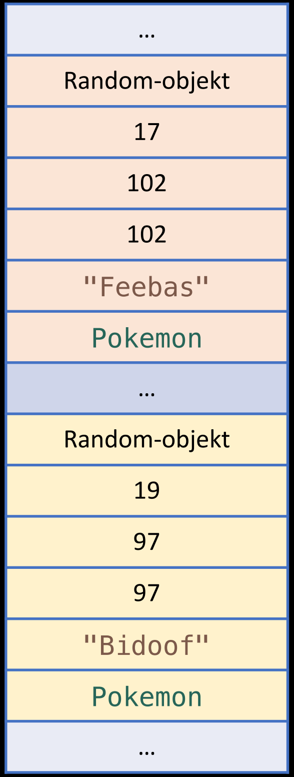
…

main args: null rounds: 1 pokemon1: 0x00B1D00F pokemon2: 0x00FEEBA5

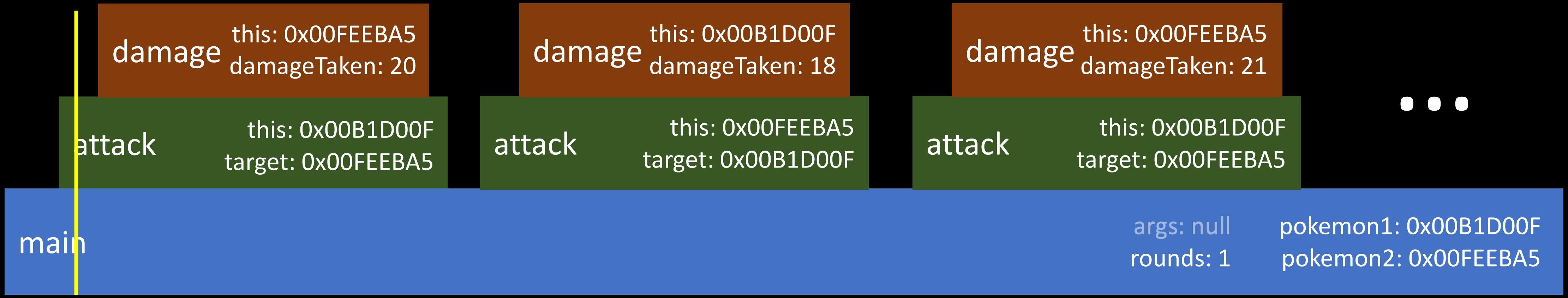
Eksempel: pokemon

```
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
…



damage this: 0x00FEEBA5 damageTaken: 20

attack this: 0x00B1D00F target: 0x00FEEBA5

damage this: 0x00B1D00F damageTaken: 18

attack this: 0x00FEEBA5 target: 0x00B1D00F

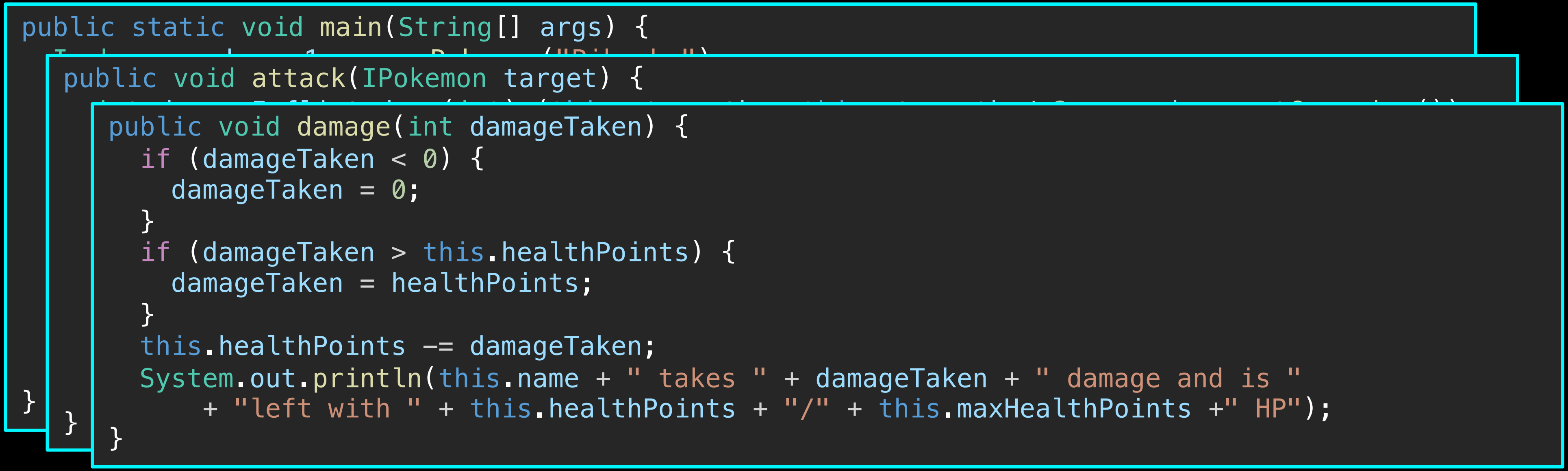
damage this: 0x00FEEBA5 damageTaken: 21

attack this: 0x00B1D00F target: 0x00FEEBA5

…

main args: null rounds: 1 pokemon1: 0x00B1D00F pokemon2: 0x00FEEBA5

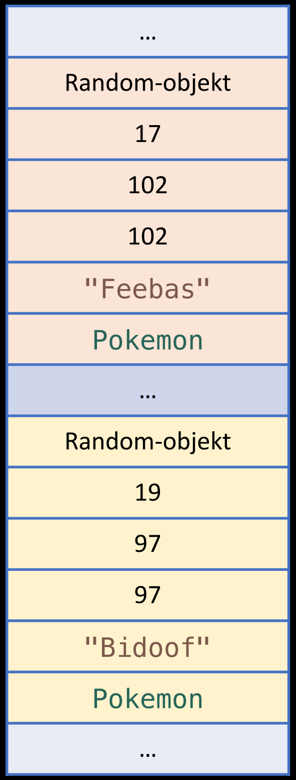
Eksempel: pokemon



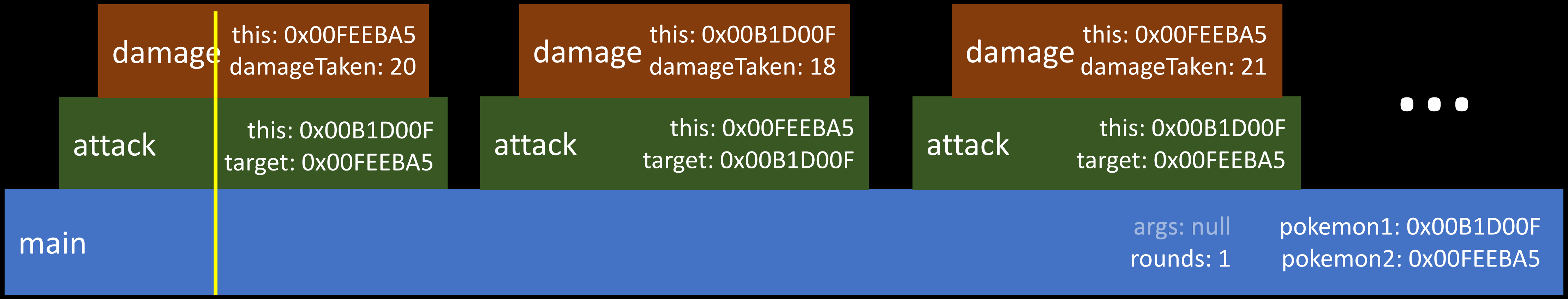
```
public static void main(String[] args) {
}
```

```
public void attack(IPokemon target) {
}
```

```
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
…



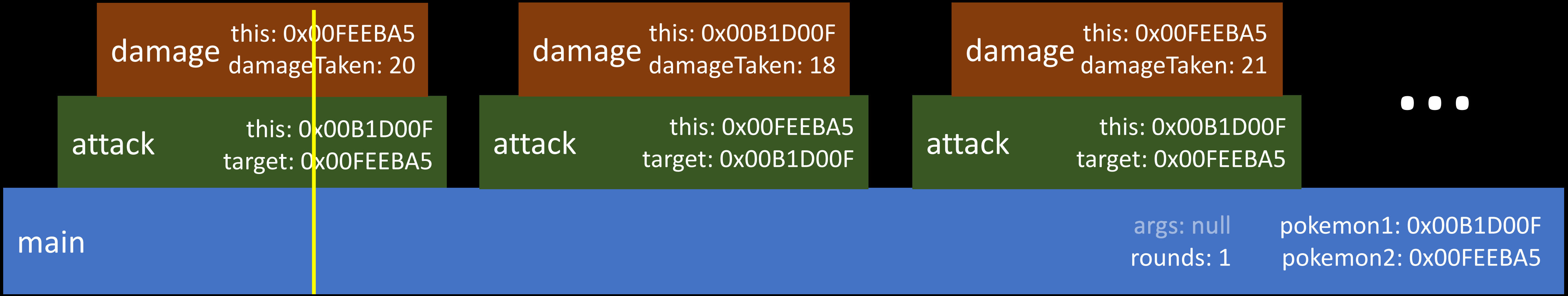
Eksempel: pokemon

```
public static void main(String[] args) {
}
```

```
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
…



damage this: 0x00FEEBA5 damageTaken: 20
attack this: 0x00B1D00F target: 0x00FEEBA5

damage this: 0x00B1D00F damageTaken: 18
attack this: 0x00FEEBA5 target: 0x00B1D00F

damage this: 0x00FEEBA5 damageTaken: 21
attack this: 0x00B1D00F target: 0x00FEEBA5

…

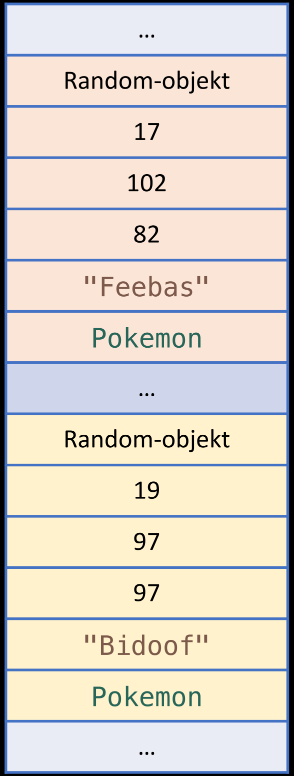
main
args: null
rounds: 1
pokemon1: 0x00B1D00F
pokemon2: 0x00FEEBA5

Eksempel: pokemon

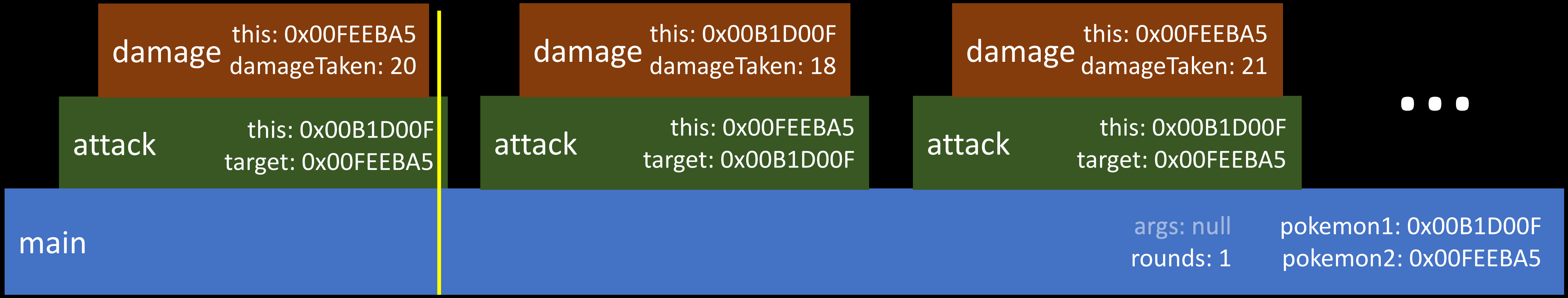
```
public static void main(String[] args) {
```

```
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
82
"Feebas"
Pokemon
…
Random-objekt
19
97
97
"Bidoof"
Pokemon
…



damage
this: 0x00FEEBA5
damageTaken: 20

attack
this: 0x00B1D00F
target: 0x00FEEBA5

damage
this: 0x00B1D00F
damageTaken: 18

attack
this: 0x00FEEBA5
target: 0x00B1D00F

damage
this: 0x00FEEBA5
damageTaken: 21

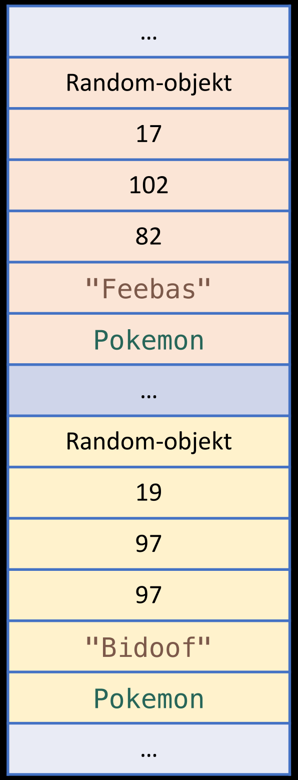
attack
this: 0x00B1D00F
target: 0x00FEEBA5

…

main
args: null
rounds: 1
pokemon1: 0x00B1D00F
pokemon2: 0x00FEEBA5

Eksempel: pokemon

```java
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

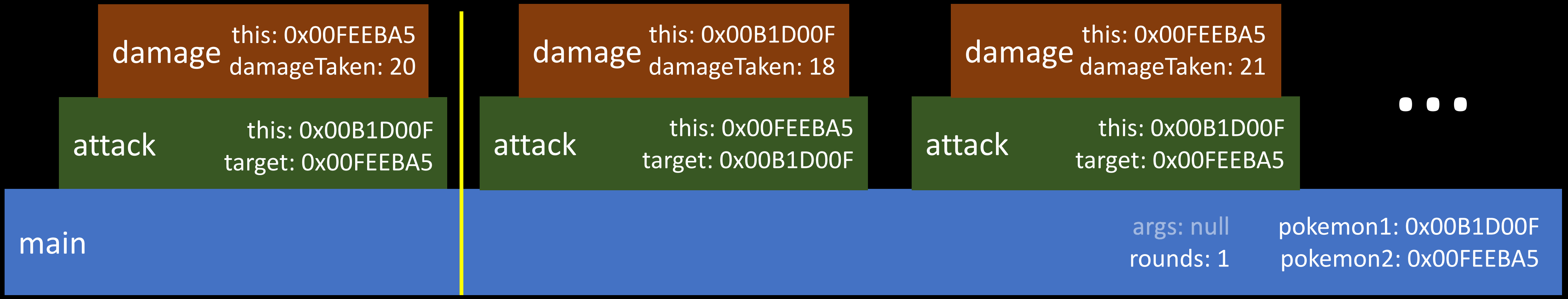
97

97

"Bidoof"

Pokemon

…



damage this: 0x00FEEBA5 damageTaken: 20

attack this: 0x00B1D00F target: 0x00FEEBA5

damage this: 0x00B1D00F damageTaken: 18

attack this: 0x00FEEBA5 target: 0x00B1D00F

damage this: 0x00FEEBA5 damageTaken: 21

attack this: 0x00B1D00F target: 0x00FEEBA5

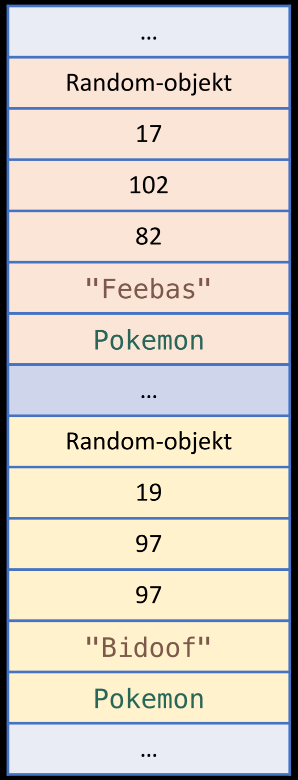
…

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

82

"Feebas"

Pokemon

…

Random-objekt

19

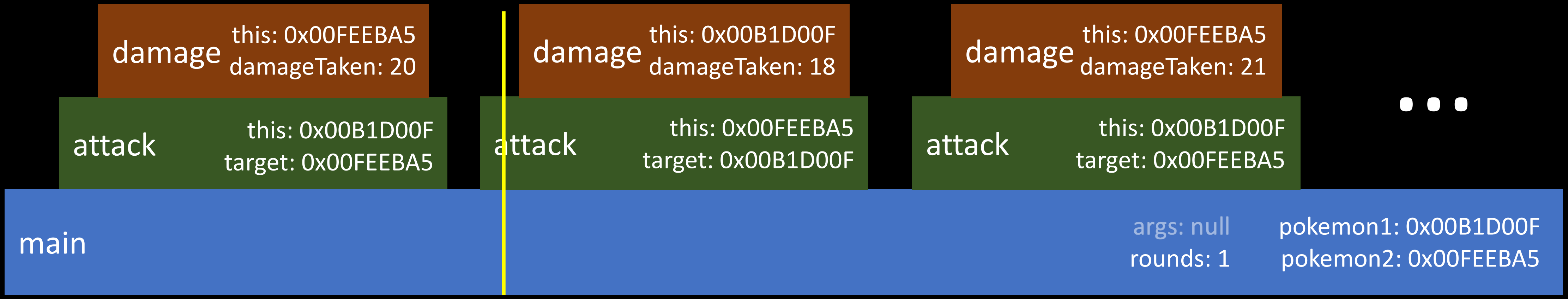
97

97

"Bidoof"

Pokemon

…



damage this: 0x00FEEBA5 damageTaken: 20

attack this: 0x00B1D00F target: 0x00FEEBA5

damage this: 0x00B1D00F damageTaken: 18

attack this: 0x00FEEBA5 target: 0x00B1D00F

damage this: 0x00FEEBA5 damageTaken: 21

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) {

}
```

```
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

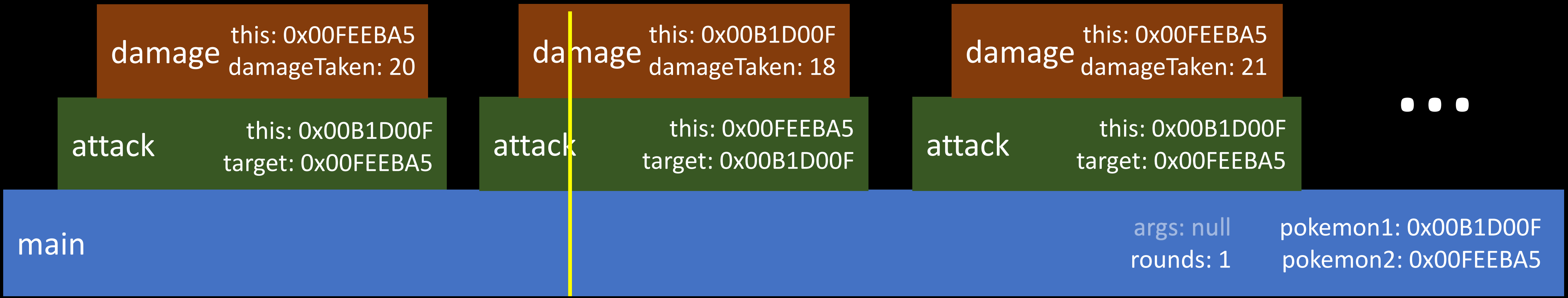
97

97

"Bidoof"

Pokemon

…



damage

this: 0x00FEEBA5

damageTaken: 20

attack

this: 0x00B1D00F

target: 0x00FEEBA5

damage

this: 0x00B1D00F

damageTaken: 18

attack

this: 0x00FEEBA5

target: 0x00B1D00F

damage

this: 0x00FEEBA5

damageTaken: 21

attack

this: 0x00B1D00F

target: 0x00FEEBA5

…

main

args: null

rounds: 1

pokemon1: 0x00B1D00F

pokemon2: 0x00FEEBA5

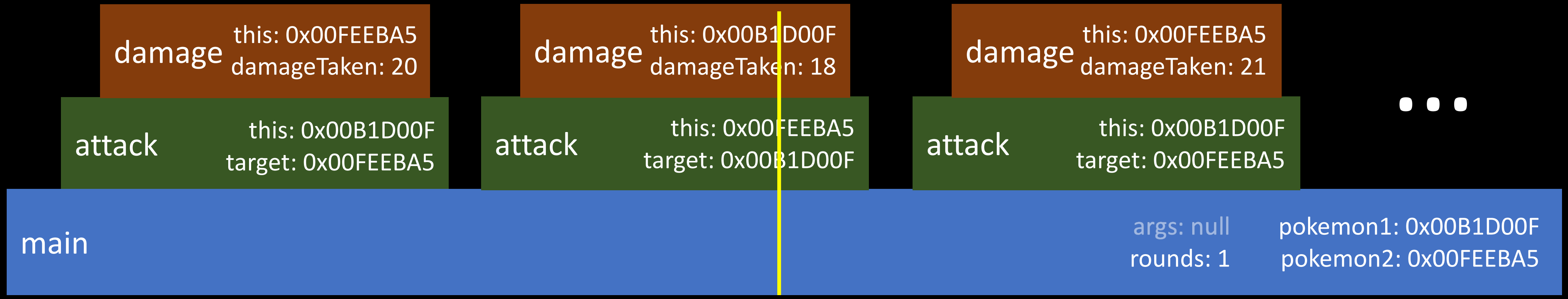
Eksempel: pokemon

```java
public static void main(String[] args) {
}
```

```java
public void attack(IPokemon target) {
}
```

```java
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
…



damage this: 0x00FEEBA5 damageTaken: 20

attack this: 0x00B1D00F target: 0x00FEEBA5

damage this: 0x00B1D00F damageTaken: 18

attack this: 0x00FEEBA5 target: 0x00B1D00F

damage this: 0x00FEEBA5 damageTaken: 21

attack this: 0x00B1D00F target: 0x00FEEBA5

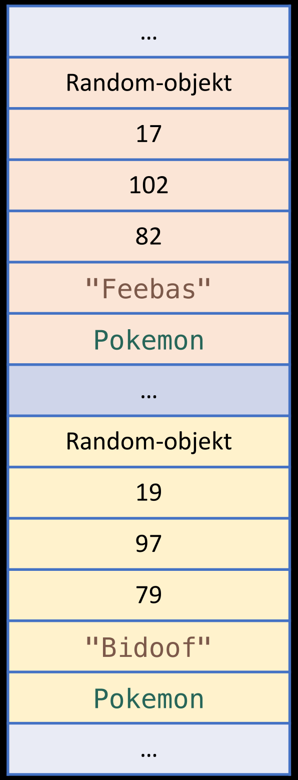
…

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

82

"Feebas"

Pokemon

…

Random-objekt

19

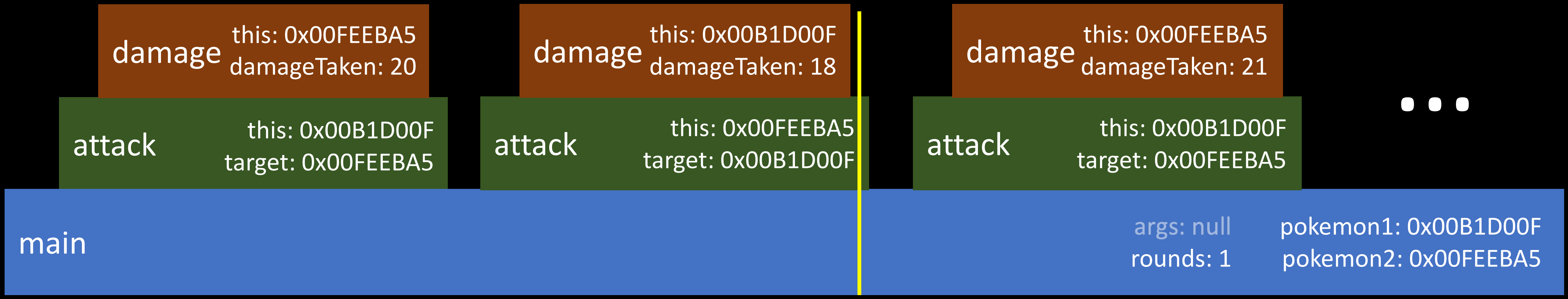
97

79

"Bidoof"

Pokemon

…



damage this: 0x00FEEBA5 damageTaken: 20

attack this: 0x00B1D00F target: 0x00FEEBA5

damage this: 0x00B1D00F damageTaken: 18

attack this: 0x00FEEBA5 target: 0x00B1D00F

damage this: 0x00FEEBA5 damageTaken: 21

attack this: 0x00B1D00F target: 0x00FEEBA5

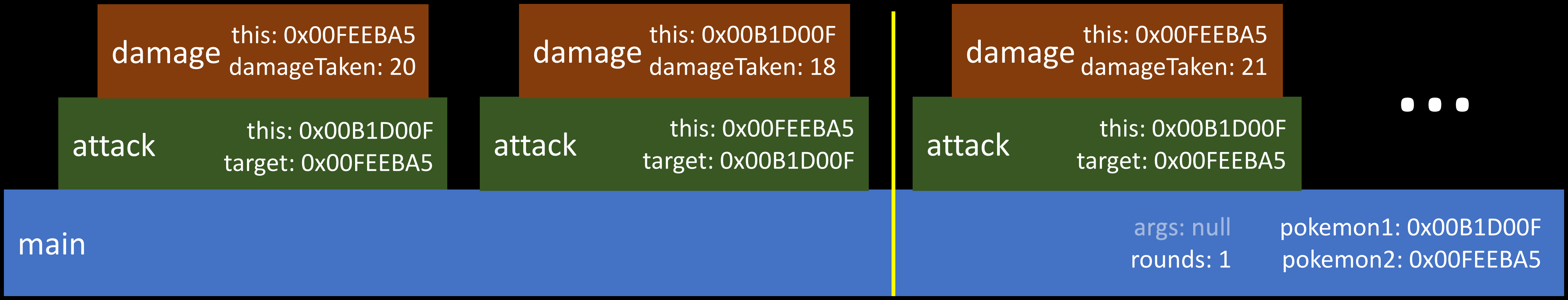
…

main args: null rounds: 1 pokemon1: 0x00B1D00F pokemon2: 0x00FEEBA5

Eksempel: pokemon

```java
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
…



damage this: 0x00FEEBA5 damageTaken: 20
attack this: 0x00B1D00F target: 0x00FEEBA5

damage this: 0x00B1D00F damageTaken: 18
attack this: 0x00FEEBA5 target: 0x00B1D00F

damage this: 0x00FEEBA5 damageTaken: 21
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

82

"Feebas"

Pokemon

…

Random-objekt

19

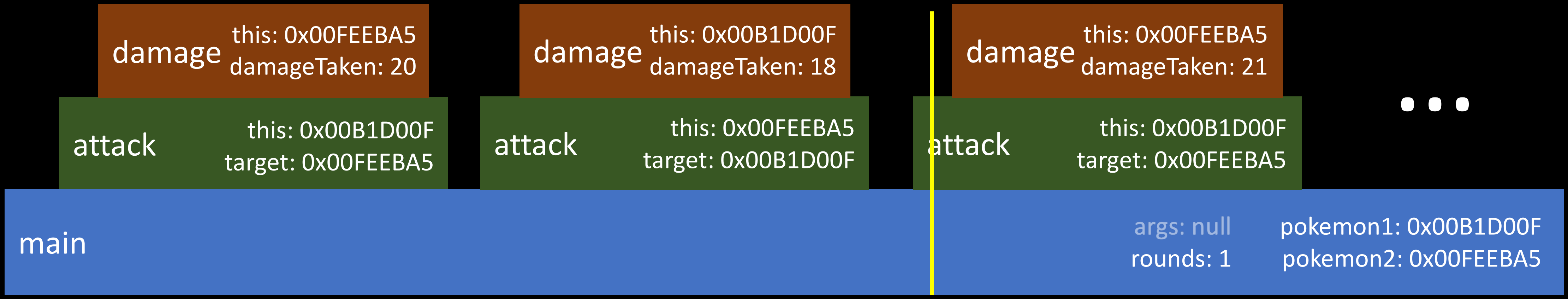
97

79

"Bidoof"

Pokemon

…



damage this: 0x00FEEBA5 damageTaken: 20

attack this: 0x00B1D00F target: 0x00FEEBA5

damage this: 0x00B1D00F damageTaken: 18

attack this: 0x00FEEBA5 target: 0x00B1D00F

damage this: 0x00FEEBA5 damageTaken: 21

attack this: 0x00B1D00F target: 0x00FEEBA5

…

main args: null rounds: 1 pokemon1: 0x00B1D00F pokemon2: 0x00FEEBA5

Eksempel: pokemon

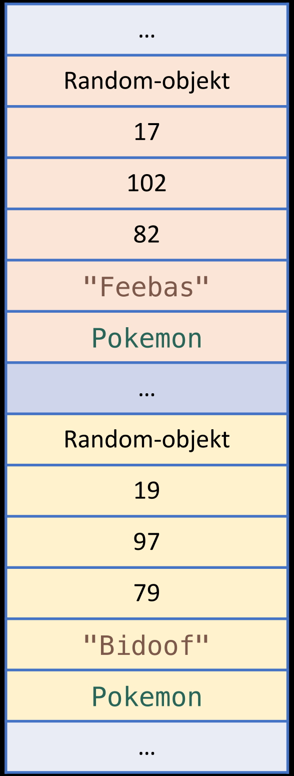
```
public static void main(String[] args) {
```

```
public void attack(IPokemon target) {
```

```
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

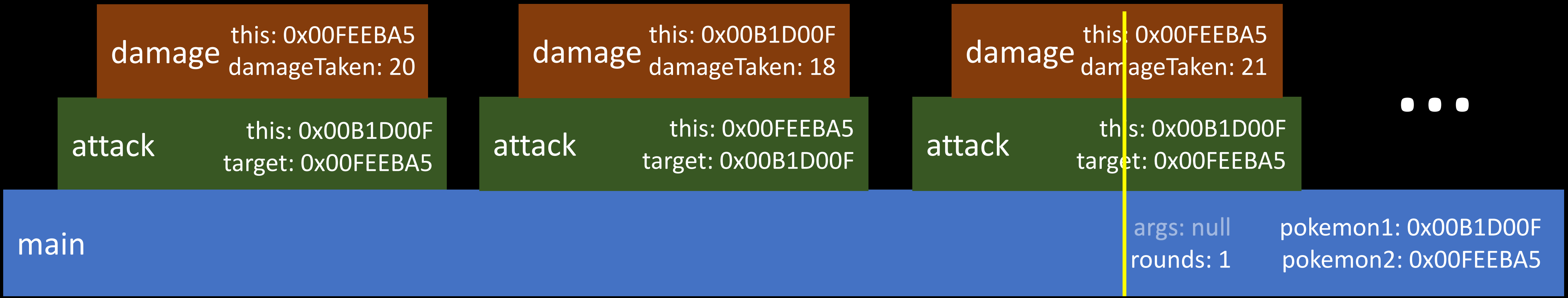
97

79

"Bidoof"

Pokemon

…



damage this: 0x00FEEBA5 damageTaken: 20

attack this: 0x00B1D00F target: 0x00FEEBA5

damage this: 0x00B1D00F damageTaken: 18

attack this: 0x00FEEBA5 target: 0x00B1D00F

damage this: 0x00FEEBA5 damageTaken: 21

attack this: 0x00B1D00F target: 0x00FEEBA5

...

main args: null rounds: 1 pokemon1: 0x00B1D00F pokemon2: 0x00FEEBA5

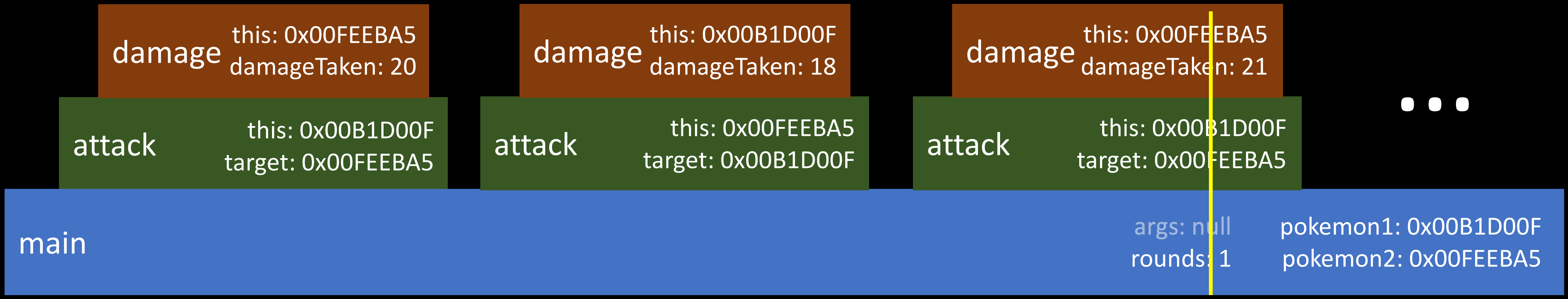
Eksempel: pokemon

```
public static void main(String[] args) {
}
```

```
public void attack(IPokemon target) {
}
```

```
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");
}
```

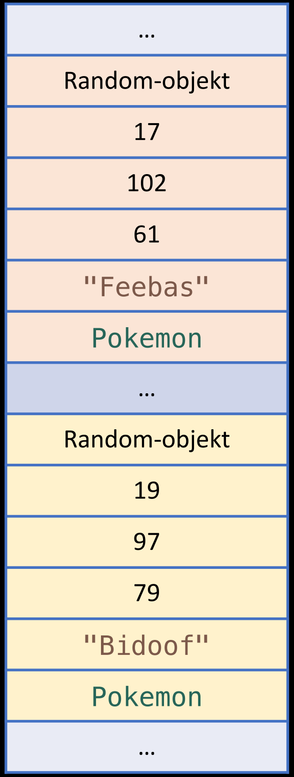
…
Random-objekt
17
102
61
"Feebas"
Pokemon
…
Random-objekt
19
97
79
"Bidoof"
Pokemon
…



Eksempel: pokemon

```java
public static void main(String[] args) {

}
```

```java
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

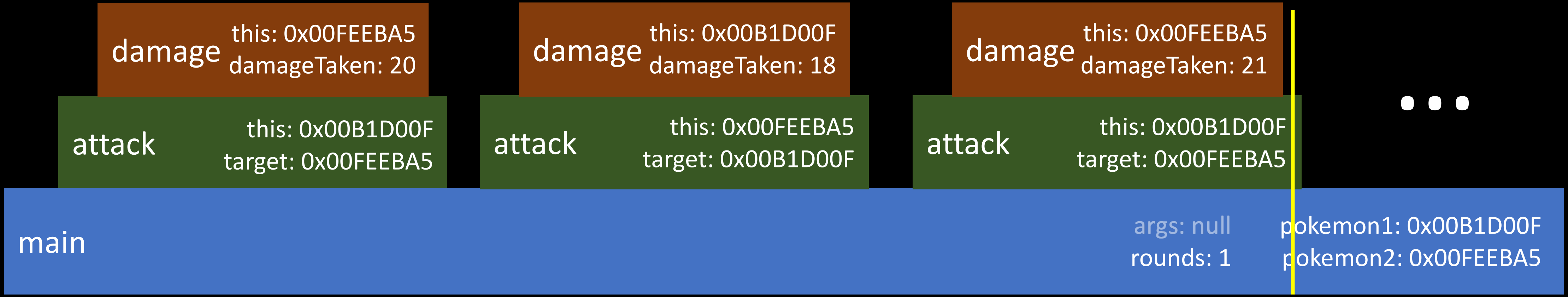
97

79

"Bidoof"

Pokemon

…



damage this: 0x00FEEBA5 damageTaken: 20

attack this: 0x00B1D00F target: 0x00FEEBA5

damage this: 0x00B1D00F damageTaken: 18

attack this: 0x00FEEBA5 target: 0x00B1D00F

damage this: 0x00FEEBA5 damageTaken: 21

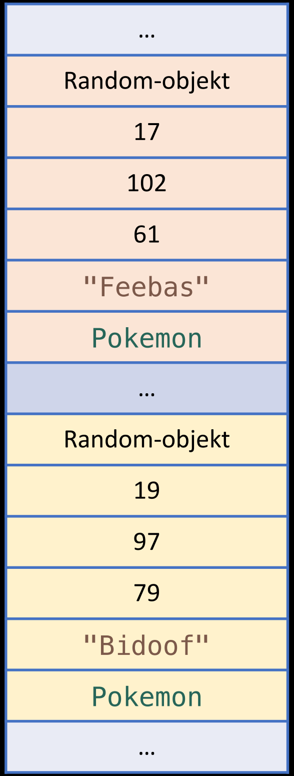
attack this: 0x00B1D00F target: 0x00FEEBA5

…

main args: null rounds: 1 pokemon1: 0x00B1D00F pokemon2: 0x00FEEBA5

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

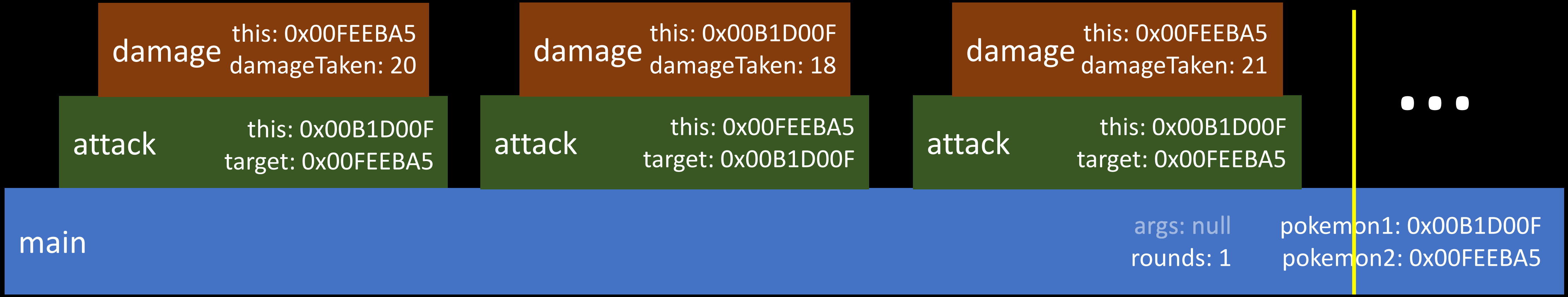
97

79

"Bidoof"

Pokemon

…



damage this: 0x00FEEBA5 damageTaken: 20

attack this: 0x00B1D00F target: 0x00FEEBA5

damage this: 0x00B1D00F damageTaken: 18

attack this: 0x00FEEBA5 target: 0x00B1D00F

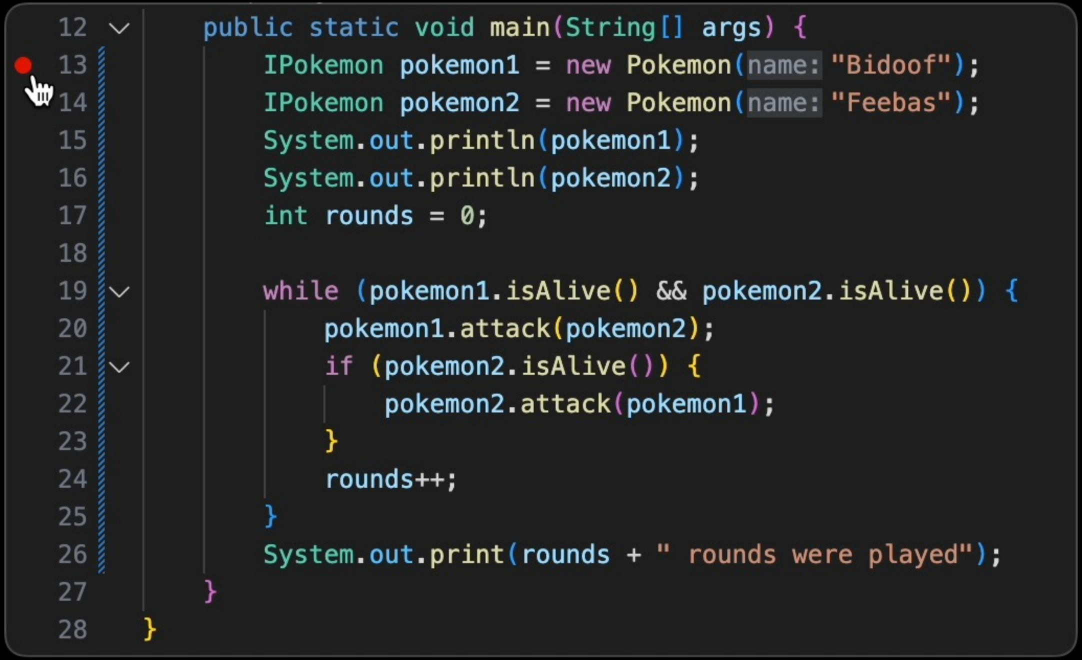
damage this: 0x00FEEBA5 damageTaken: 21

attack this: 0x00B1D00F target: 0x00FEEBA5

…

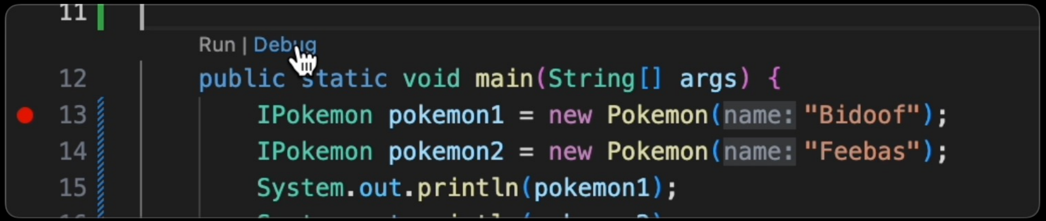
main args: null rounds: 1 pokemon1: 0x00B1D00F pokemon2: 0x00FEEBA5

Debugger



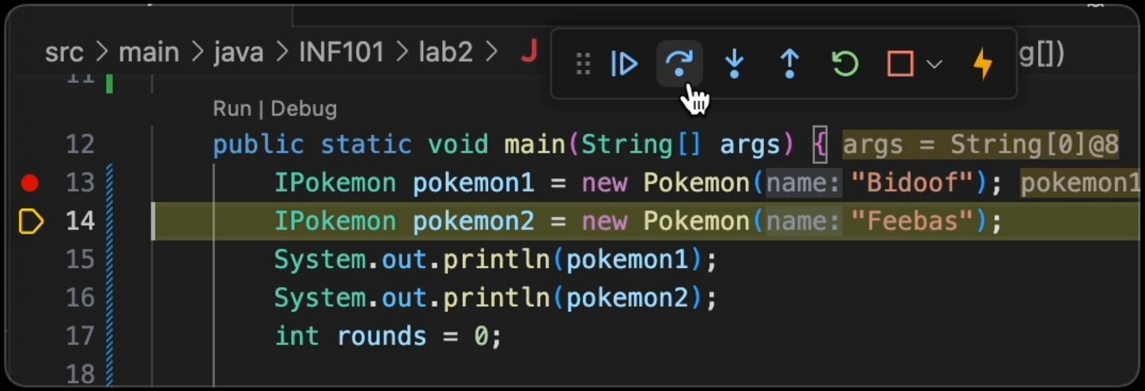
```
public static void main(String[] args) {
    IPokemon pokemon1 = new Pokemon(name: "Bidoof");
    IPokemon pokemon2 = new Pokemon(name: "Feebas");
    System.out.println(pokemon1);
    System.out.println(pokemon2);
    int rounds = 0;

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

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



```
Run | Debug
public static void main(String[] args) {
    IPokemon pokemon1 = new Pokemon(name: "Bidoof");
    IPokemon pokemon2 = new Pokemon(name: "Feebas");
    System.out.println(pokemon1);
```

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



```
Run | Debug
public static void main(String[] args) { args = String[0]@8
    IPokemon pokemon1 = new Pokemon(name: "Bidoof"); pokemon1
    IPokemon pokemon2 = new Pokemon(name: "Feebas");
    System.out.println(pokemon1);
    System.out.println(pokemon2);
    int rounds = 0;
```

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

Debugger: navigasjon

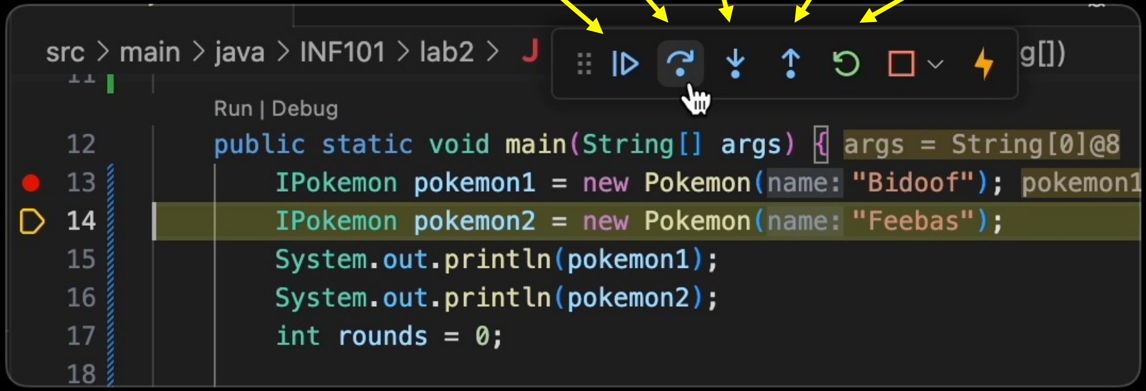
Gå inn i metodekallet

Gjør ferdig kun denne metoden

Gå til neste linje

Start på nytt

Fortsett til neste breakpoint



src > main > java > INF101 > lab2 > J g[])

Run | Debug

```
public static void main(String[] args) { args = String[0]@8
    IPokemon pokemon1 = new Pokemon(name: "Bidoof"); pokemon1
    IPokemon pokemon2 = new Pokemon(name: "Feebas");
    System.out.println(pokemon1);
    System.out.println(pokemon2);
    int rounds = 0;
```

Breakpoint

Neste linje som vil utføres

Debugger: inspeksjon

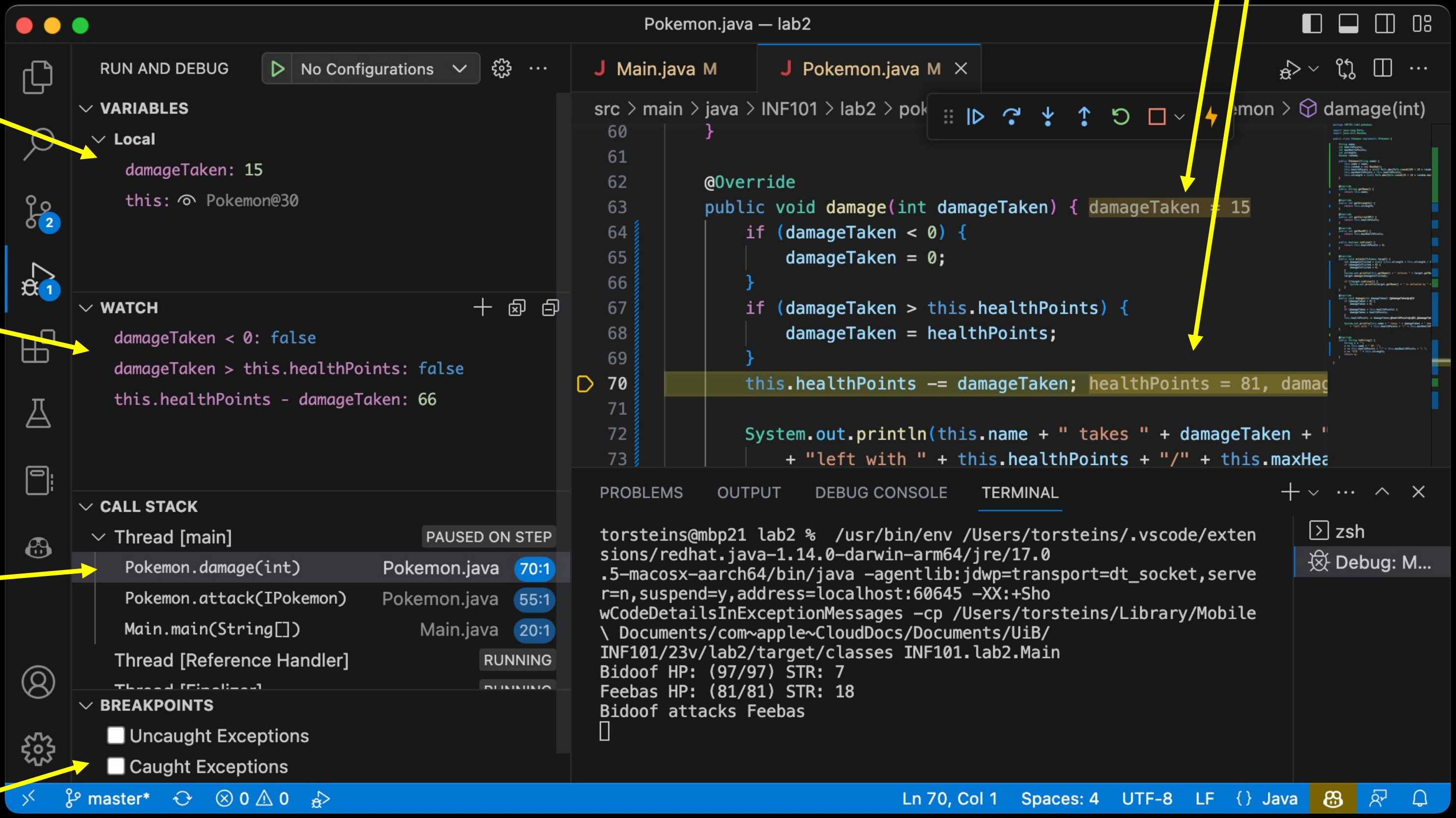
Se verdier av variabler

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



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)	Pokemon.java	70:1
Pokemon.attack(IPokemon)	Pokemon.java	55:1
Main.main(String[])	Main.java	20:1
Thread [Reference Handler]		RUNNING

BREAKPOINTS

□ Uncaught Exceptions

□ Caught Exceptions

```java
    }

    @Override
    public void damage(int damageTaken) { damageTaken = 15
        if (damageTaken < 0) {
            damageTaken = 0;
        }
        if (damageTaken > this.healthPoints) {
            damageTaken = healthPoints;
        }
        this.healthPoints -= damageTaken; healthPoints = 81, damag

        System.out.println(this.name + " takes " + damageTaken + "
            + "left with " + this.healthPoints + "/" + this.maxHea
```

```
torsteins@mbp21 lab2 %  /usr/bin/env /Users/torsteins/.vscode/exten
sions/redhat.java-1.14.0-darwin-arm64/jre/17.0
.5-macosx-aarch64/bin/java -agentlib:jdwp=transport=dt_socket,serve
r=n,suspend=y,address=localhost:60645 -XX:+Sho
wCodeDetailsInExceptionMessages -cp /Users/torsteins/Library/Mobile
\ Documents/com~apple~CloudDocs/Documents/UiB/
INF101/23v/lab2/target/classes INF101.lab2.Main
Bidoof HP: (97/97) STR: 7
Feebas HP: (81/81) STR: 18
Bidoof attacks Feebas
```

Likhet i egne klasser

```java
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

```java
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

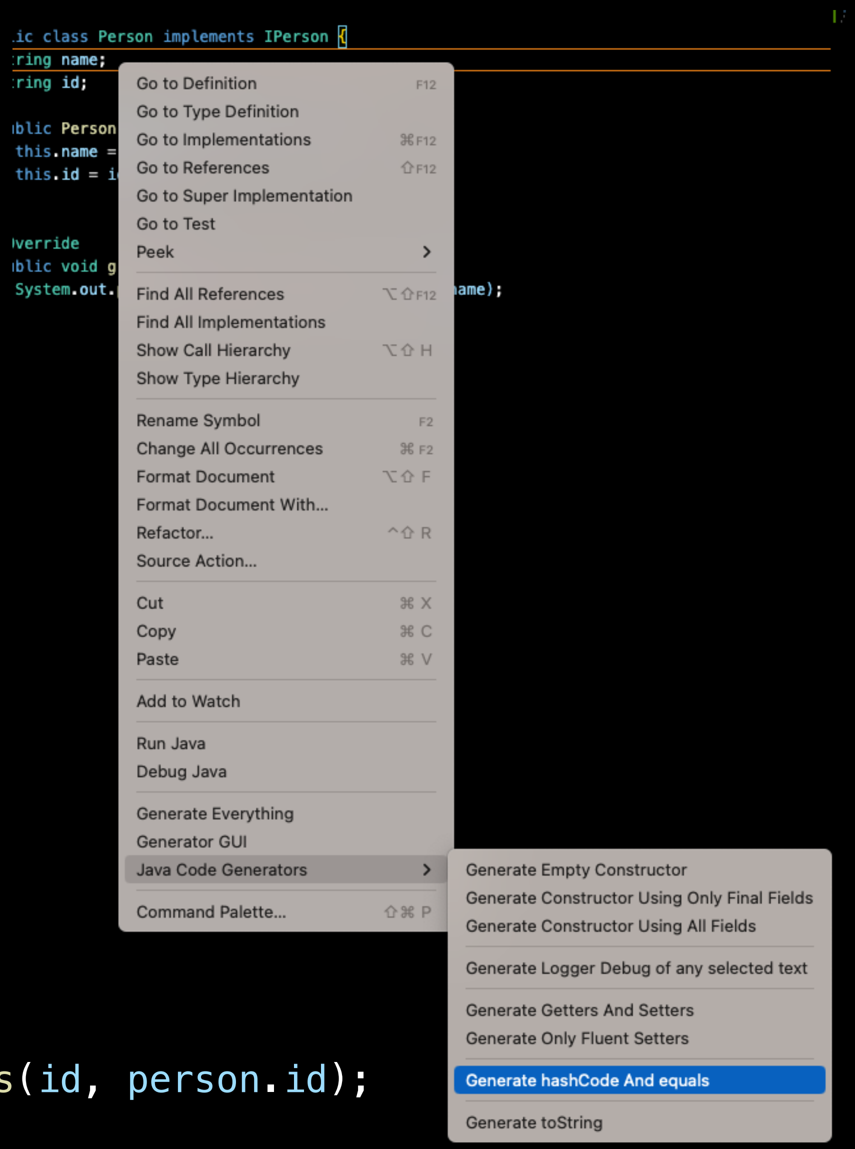
Viktig!

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

```java
@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;

ıblic Person
 this.name =
 this.id = i

)verride
ıblic void g
 System.out.                                  ıame);
```