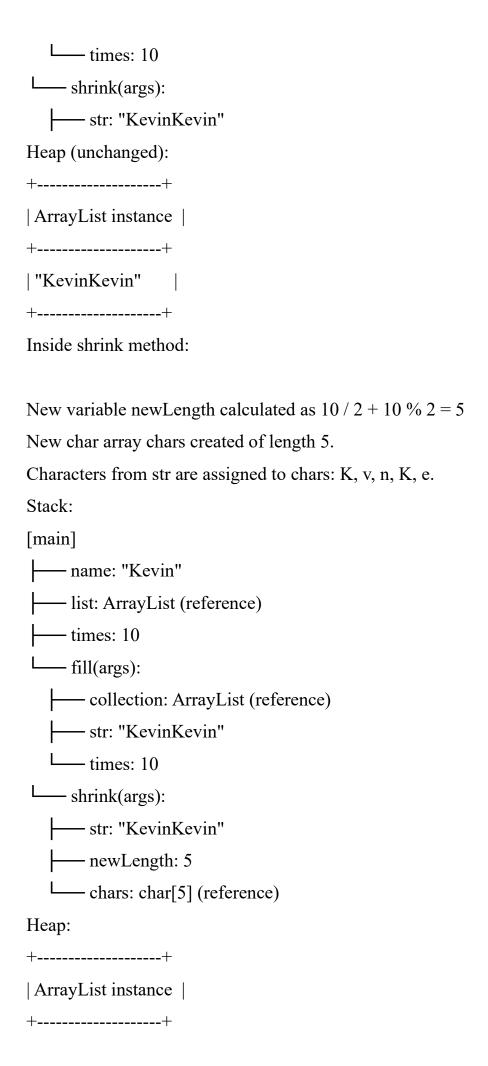
1. Start
Classes and Methods:
Main class with main method.
fill method: takes a collection, string, and an integer.
shrink method: shrinks a string by taking every second character.
2. Stack and Heap Memory Breakdown
Initial State:
At the start, the stack is empty.
Execution Flow
Call to main method:
Stack:
[main]
Heap:
(empty)
Inside main method:
Local variables are created:
name: "Kevin"
list: new ArrayList<>()
times: 10
Stack:
[main]
mame: "Kevin"
list: ArrayList (reference)
times: 10
Heap:
++
ArrayList instance
++
Call to fill method:

```
Parameters passed:
collection: list (reference to ArrayList)
str: "KevinKevin" (concatenation of name + name)
times: 10
Stack:
[main]
name: "Kevin"
list: ArrayList (reference)
├── times: 10
fill(args):
  — collection: ArrayList (reference)
  str: "KevinKevin"
  ____ times: 10
Heap:
+----+
| ArrayList instance |
+----+
| "KevinKevin" | <-- (String object)
+----+
Inside fill method:
Call to shrink method with str: "KevinKevin"
Stack:
[main]
— name: "Kevin"
list: ArrayList (reference)
____ times: 10
fill(args):
  — collection: ArrayList (reference)
     — str: "KevinKevin"
```



```
"KevinKevin"
+----+
+----+
Return from shrink method:
Create a new String object from chars ("KvnKe").
Stack:
[main]
— name: "Kevin"
list: ArrayList (reference)
____ times: 10
fill(args):
  — collection: ArrayList (reference)
  str: "KevinKevin"
  └── times: 10
  shrunk: "KvnKe" <-- new String object reference
Heap:
+----+
| ArrayList instance |
+----+
"KevinKevin"
+----+
| char[5] |
+----+
| "KvnKe" | <-- (new String object)
+----+
Back to fill method:
```

Update times using shrunk.length(): times = (10 + 5) / 2 = 7.5 (will be 7 since it's an int).

```
Loop for i = 0 to 3 (7 / 2 = 3):
Add shrunk ("KvnKe") to collection.
Stack:
[main]
— name: "Kevin"
 — list: ArrayList (reference)
____ times: 10
fill(args):
  — collection: ArrayList (reference)
  └── times: 7
  shrunk: "KvnKe"
Heap (after adding "KvnKe"):
+----+
| ArrayList instance |
+----+
| "KevinKevin" |
+----+
| char[5] |
+----+
| "KvnKe" |
+----+
| "KvnKe" | <-- first addition
| "KvnKe" | <-- second addition
| "KvnKe" | <-- third addition
+____+
Return from fill method:
Return times, which is 7.
Stack:
[main]
```

```
— name: "Kevin"
  — list: ArrayList (reference)
____ times: 10
fill(args):
  — collection: ArrayList (reference)
  str: "KevinKevin"
  times: 7
Heap (unchanged):
+----+
| ArrayList instance |
+----+
| "KevinKevin" |
+----+
| char[5]
+----+
| "KvnKe" |
+----+
| "KvnKe" | <-- three additions
| "KvnKe" |
| "KvnKe" |
+----+
Back to main method:
Call to System.out.println(times + fill(...)):
Update times to 10 + 7 = 17.
Stack:
[main]
  — name: "Kevin"
list: ArrayList (reference)
 — times: 17
Print Result:
```

The final output is printed: 17.
Final State Summary
Stack:
[main]
name: "Kevin"
list: ArrayList (reference)
times: 17
Heap:
++
ArrayList instance
++
"KevinKevin"
++
char[5]
++
"KvnKe"
++
"KvnKe" < three additions
"KvnKe"
"KvnKe"
++

Conclusion

Stack Memory: Contains method call hierarchy and local variables.

Heap Memory: Contains all the object instances created during the execution.