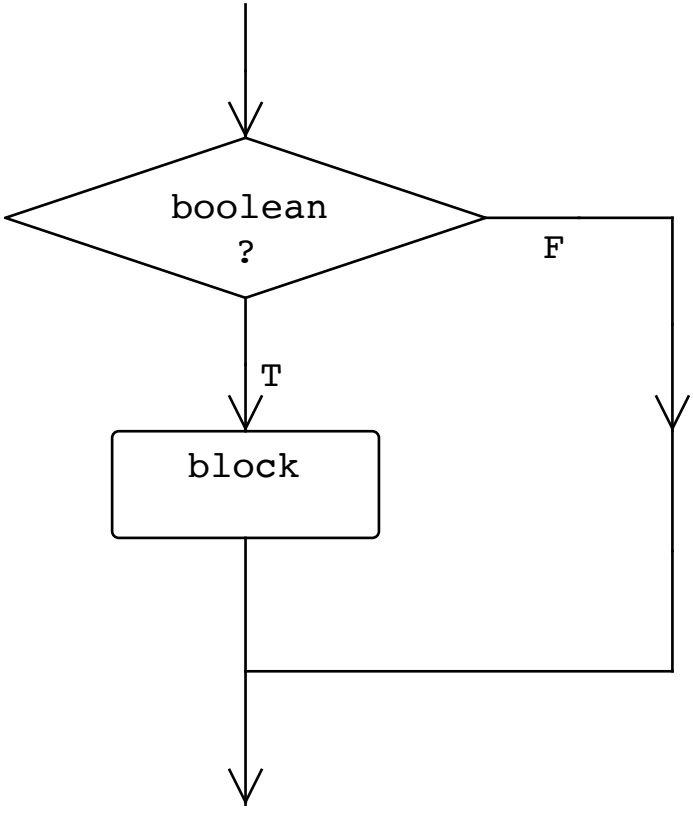
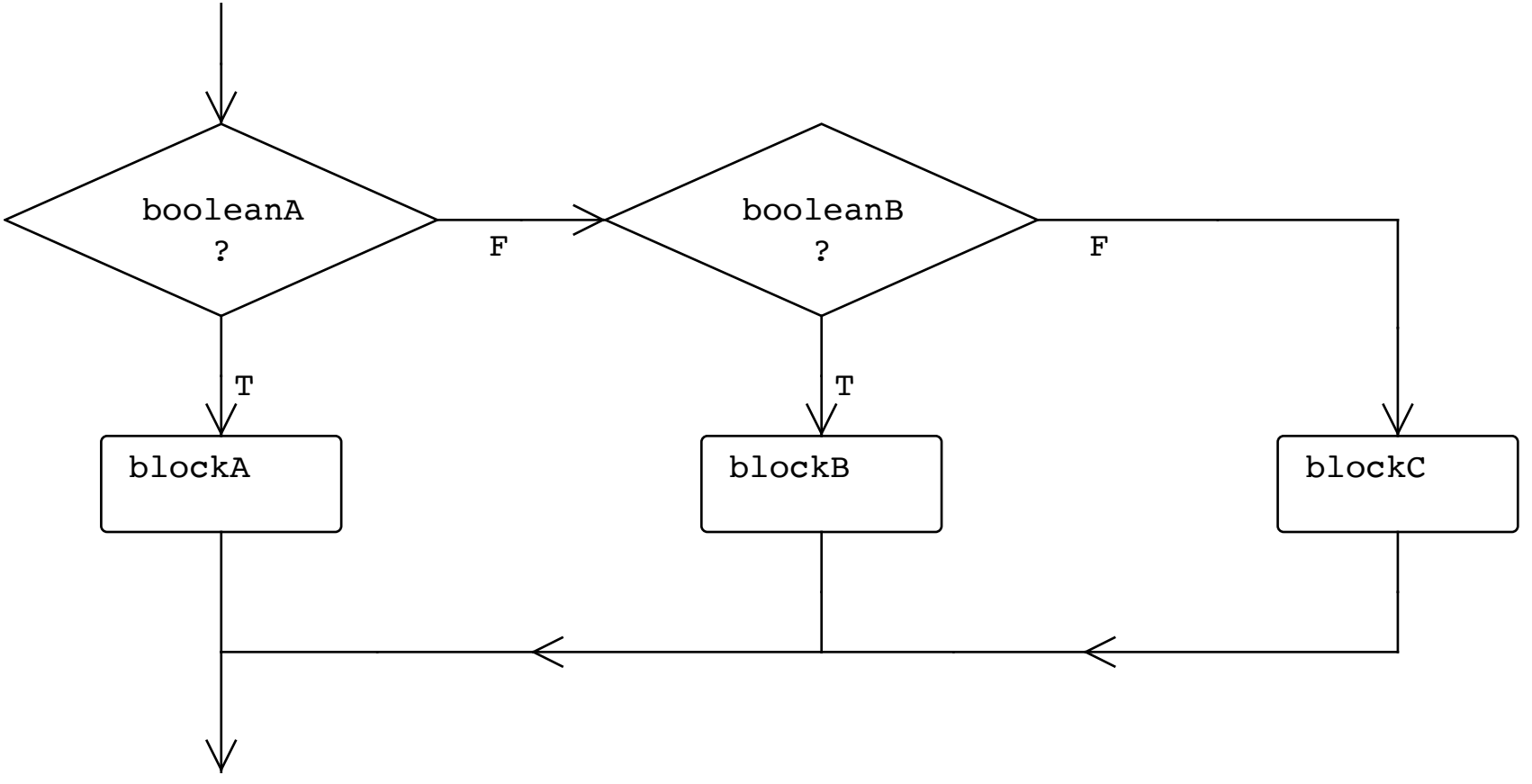
This flowchart models the flow of control through an if statement:

```
if (boolean) {
    // block
}
```



This flowchart models the flow of control through
an if...else if...else statement:

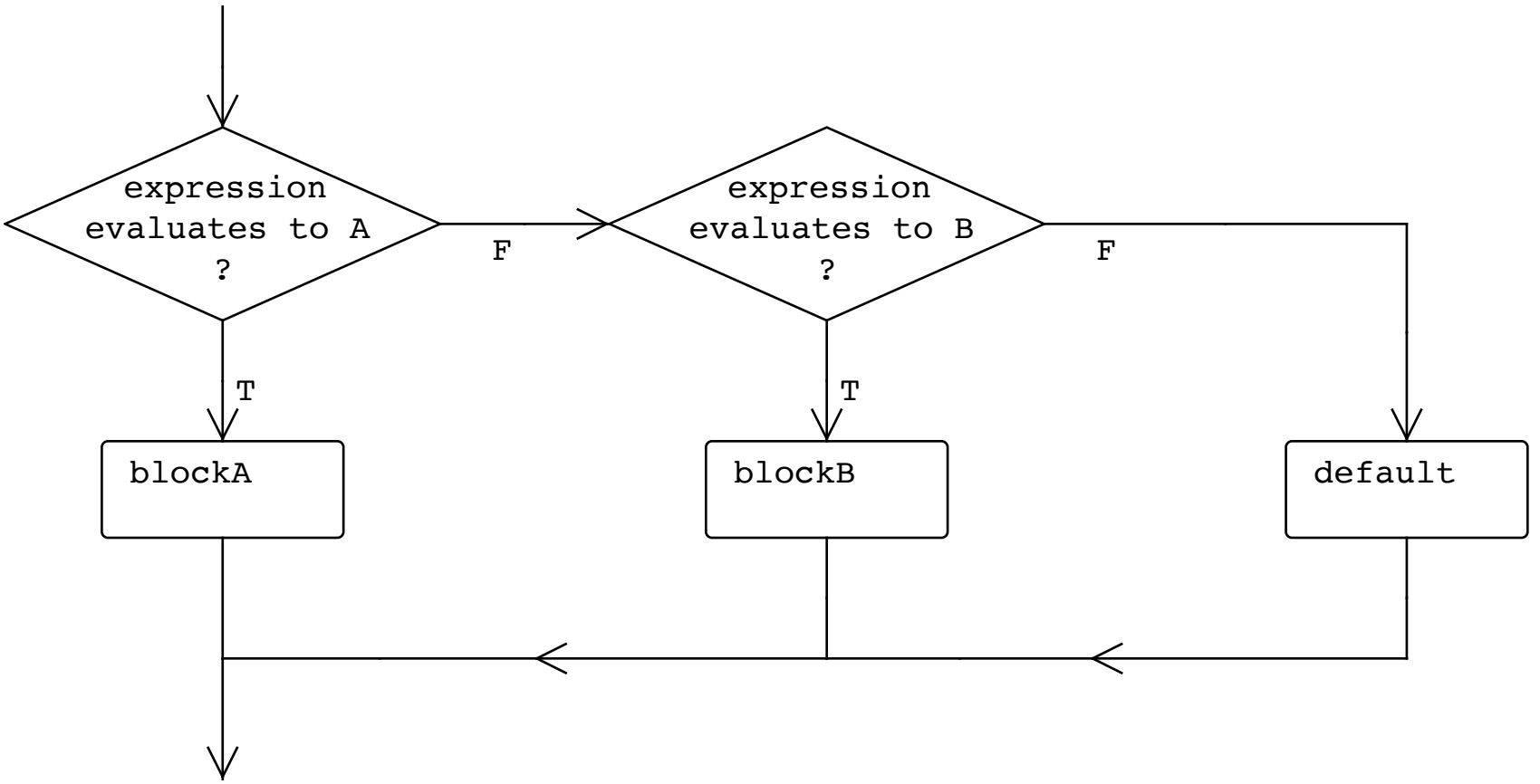
```
if (booleanA) {
    // blockA
}
else if (booleanB) {
    // blockB
}
else {
    // blockC
}
```



This flowchart models the flow of control through
a switch statement:

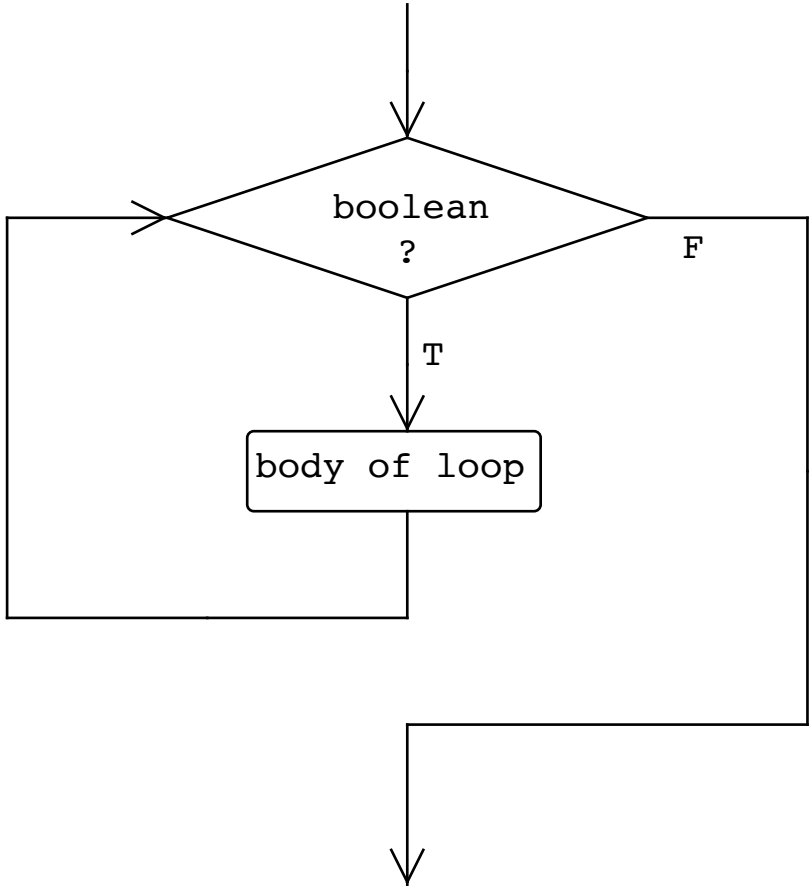
```
switch (expression) {
case A:
    // blockA
    break;
case B:
    // blockB
    break;
default:
    // default
}
```

Note that switch statements work with expressions that evaluate to int, short, byte or char (and with their corresponding wrapper classes Integer, Short, Byte and Character), and with enumerations and Strings.



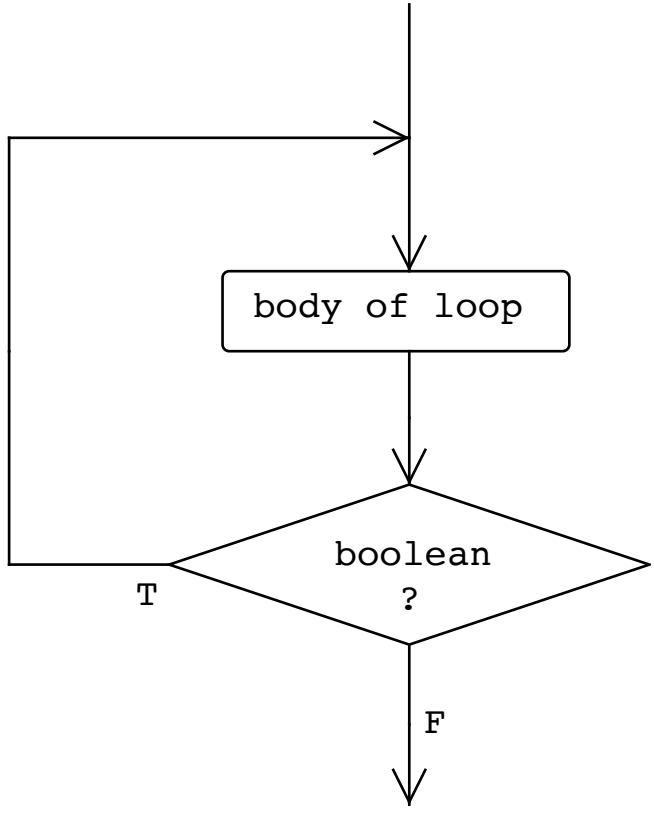
This flowchart models the flow of control through a while loop:

```
while (boolean) {
    // body of loop
}
```



This flowchart models the flow of control through a do-while loop:

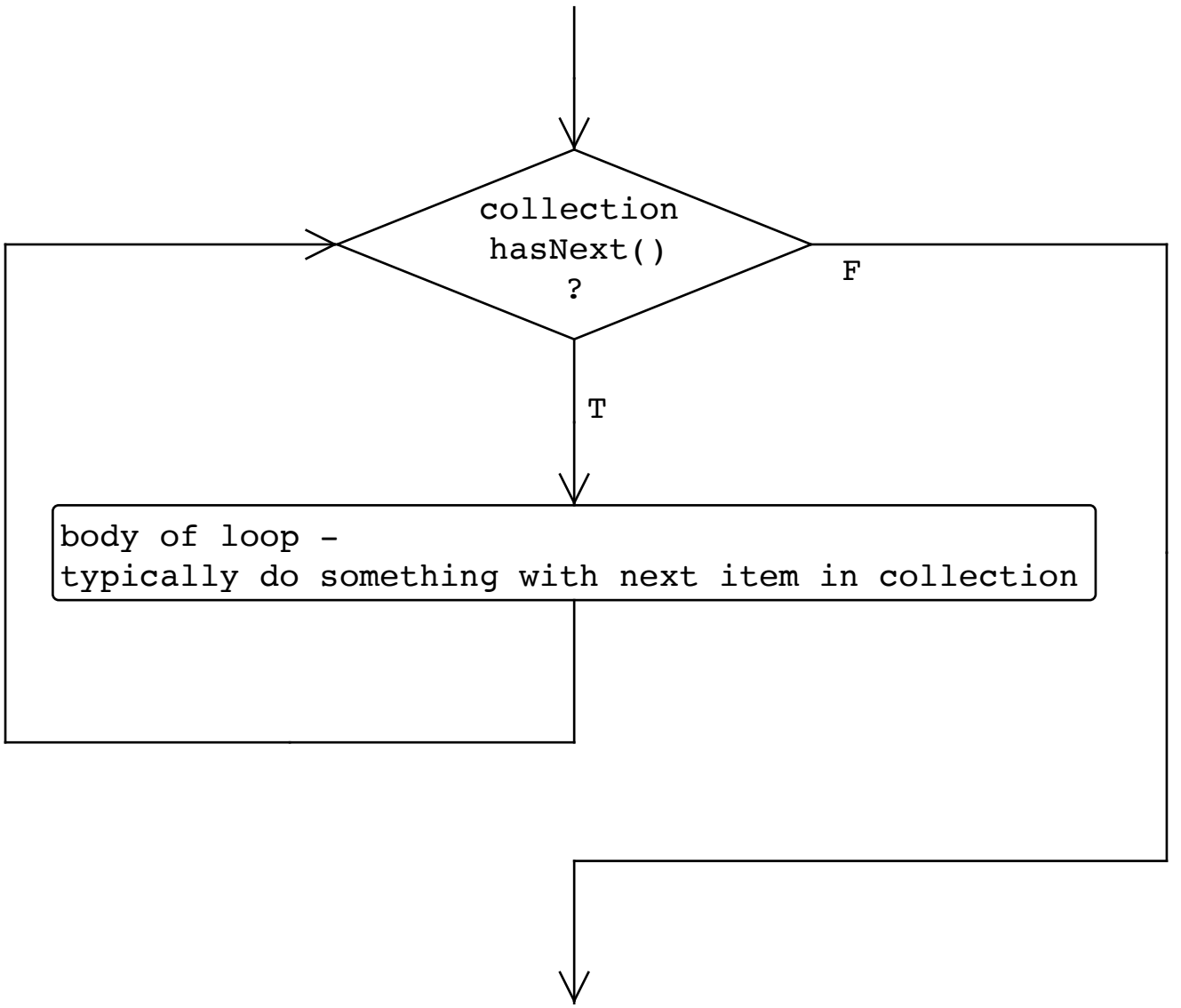
```
do {
    // body of loop
} while (boolean);
```



This flowchart models the flow of control through a for-each loop:

```
for (Type next : collection) {
    // body of loop -
    // typically do something with next item in collection
}
```

Note that collection must be a Java collection such as a List or Set (later in the course we'll see that it can also be used with an array or with a type that is Iterable).



This flowchart models the flow of control through a for loop:

```
for (init; boolean; post) {
    // body of loop
}
```