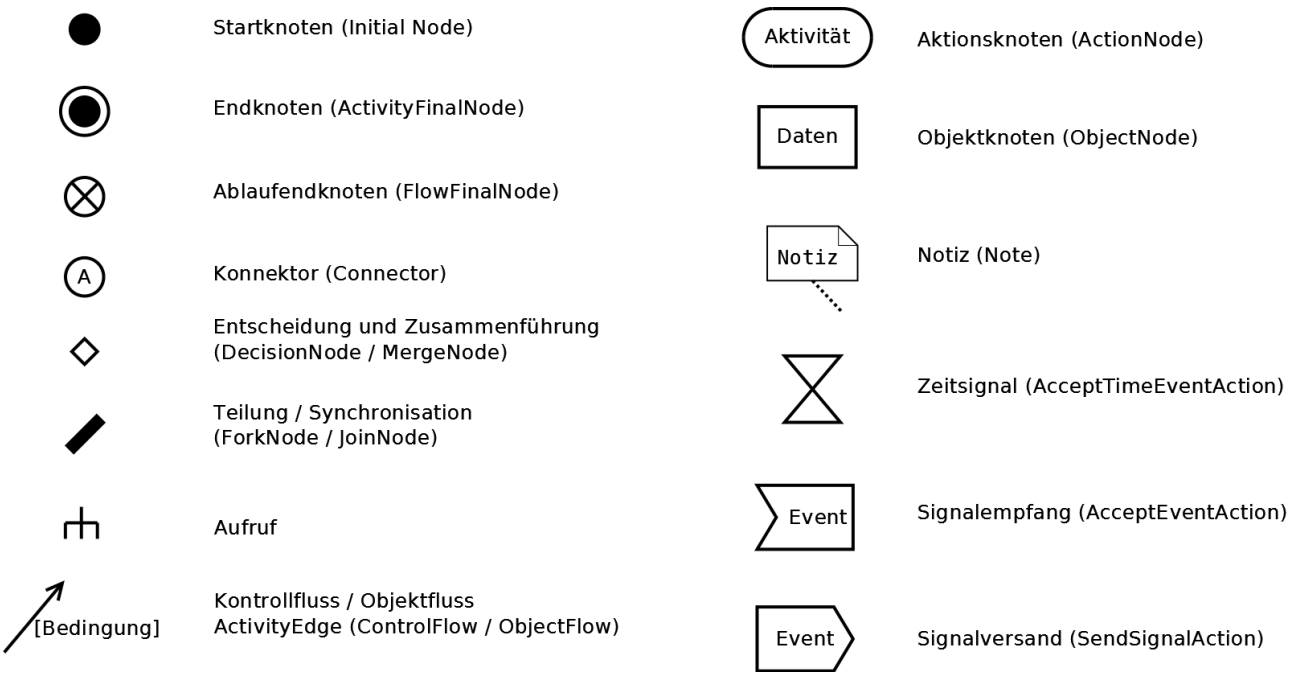
HOCHSCHULE FURTWANGEN UNIVERSITY | HFU

Activity Diagrams

and their analogies in code (C#)

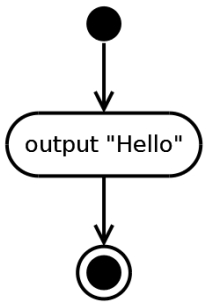
Prof. Dipl.-Ing. Jirka R. Dell'Oro-Friedl

1. Elements



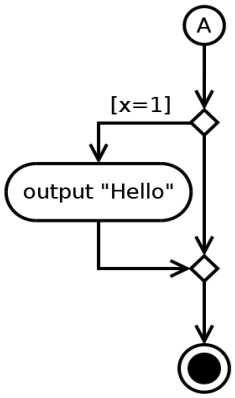
2. Basic flow structures

Linear



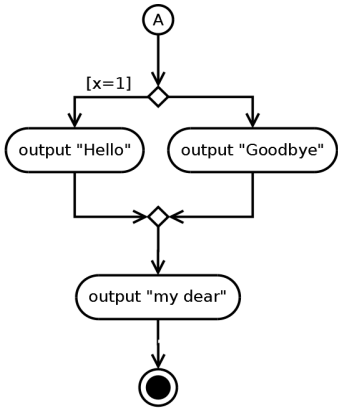

```
Console.Write("Hello");
```

Conditional



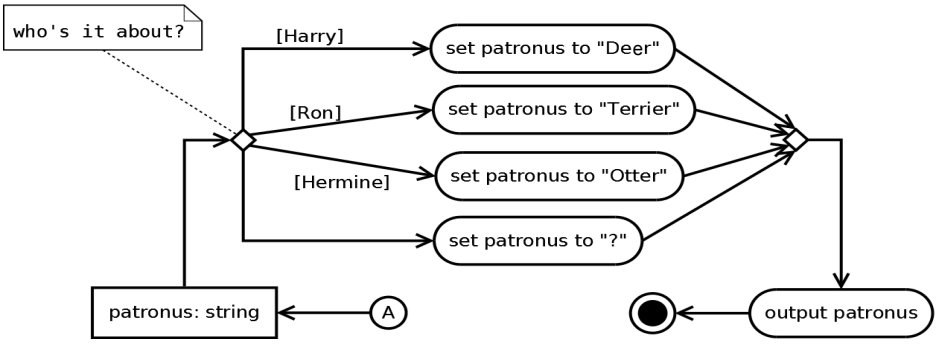

```
...
if (x == 1)
 Console.Write("Hello");
```

Exclusive Conditional




```
...
if (x == 1)
  Console.Write("Hello");
else
  Console.Write("Goodbye");
Console.Write(" my dear");
```

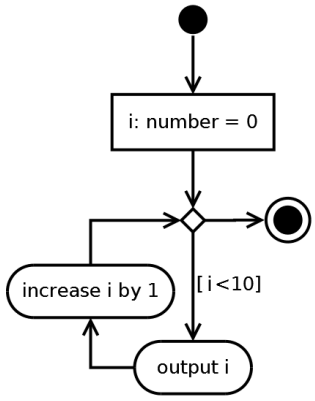
Multiple Conditions




```
...
string patronus;
switch (person) {
    case "Harry":
        patronus = "Deer";
        break;
    case "Hermine":
        patronus = "Otter";
        break;
    case "Ron":
        patronus = "Terrier";
        break;
    default:
        patronus = "?";
        break;
}
Console.Write(patronus);
```

3. Loops

Loop (Pre-Test)

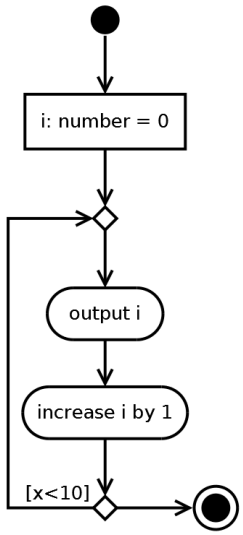



```
int i = 0;
while (i < 10) {
    Console.Write(i);
    i++;
}
```

oder

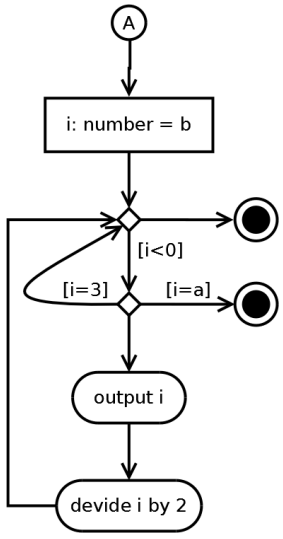
```
for (int i = 0; i < 10; i++)
    Console.Write(i);
```

Loop (Post-Test)




```
int i = 0;
do {
    Console.Write(i);
    i++;
} while (i < 10);
```

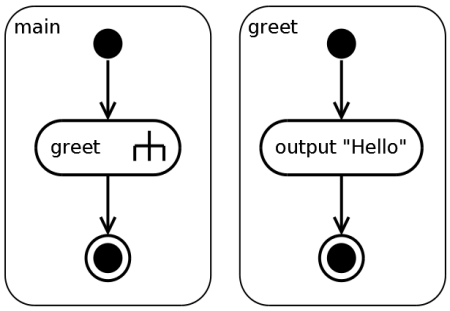
Loop with additional control conditions




```
for (int i = b; i > 1; i/=2) {
    if (i == 3)
        continue;
    if (i == a)
        break;
    Console.Write(i);
}
```

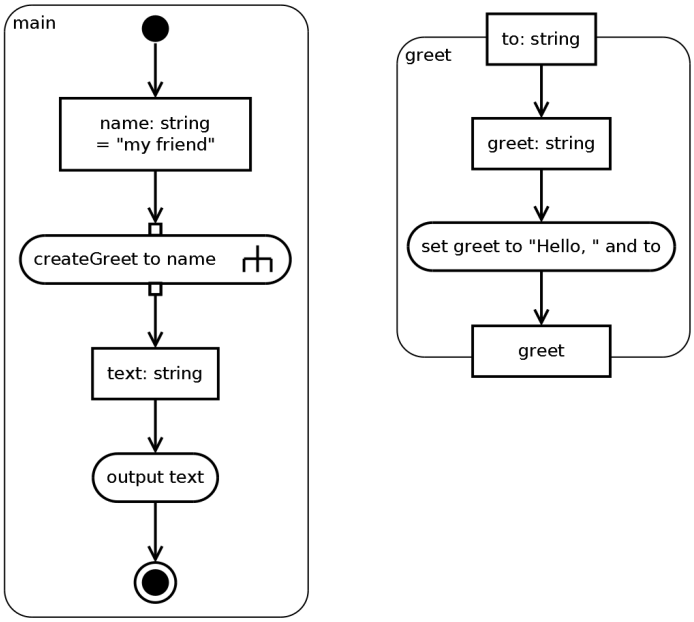
4. Subactivities

Subactivity (Function/Method)




```
greet();
void greet() {
    Console.Write("Hello");
}
```

Subactivity with Input/Output




```
string name = "my friend";
string text = createGreet(name);
Console.Write(text);

string createGreet(string to) {
    string greet;
    greet = "Hello, " + to;
    return greet;
}
```

5. Signals

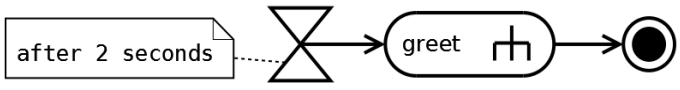
Accept Event


```
window.addEventListener("triggerGreet", greet);
```

Send Signal


```
let event: Event = new Event("triggerGreet");
window.dispatchEvent(event);
```

Accept Time Event




```
window.setTimeout(greet, 2000);
```