
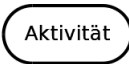














Activity Diagrams

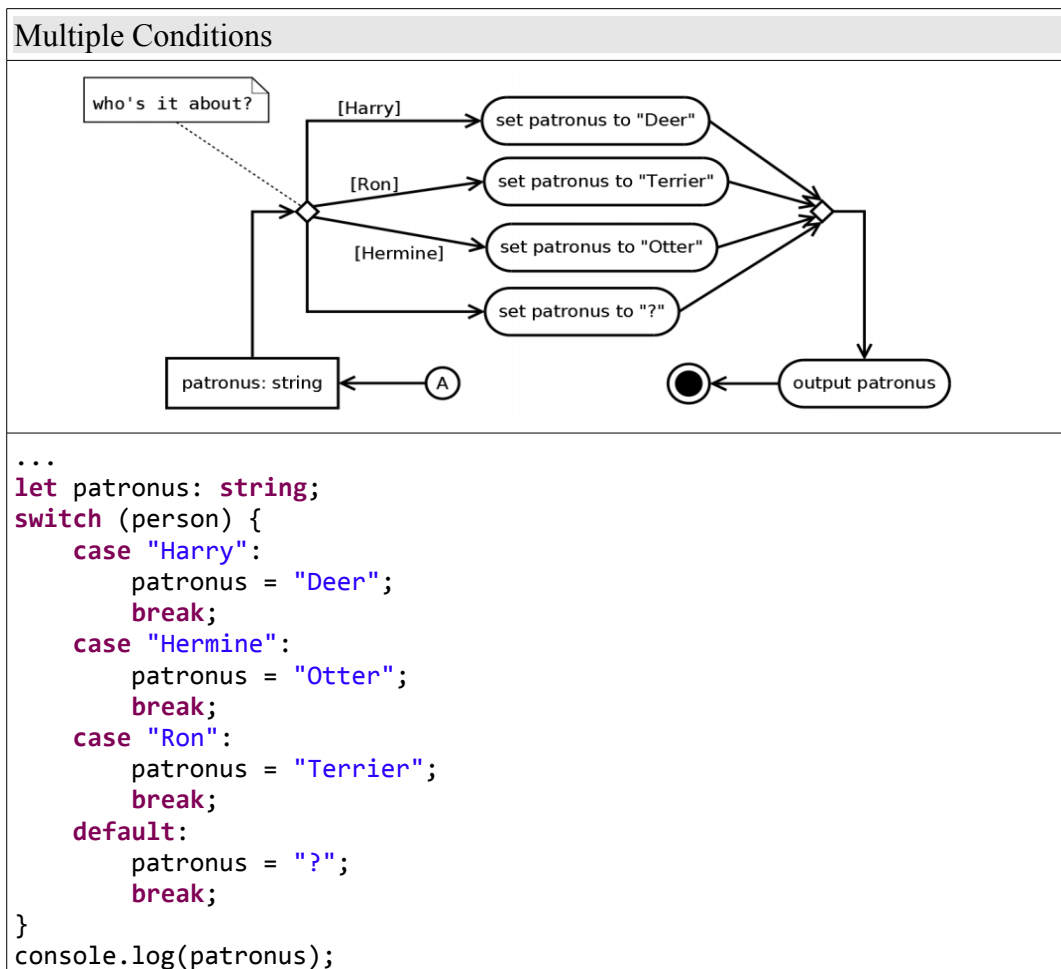
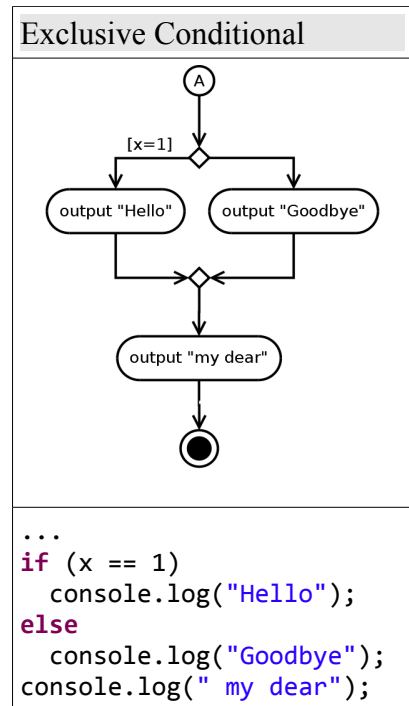
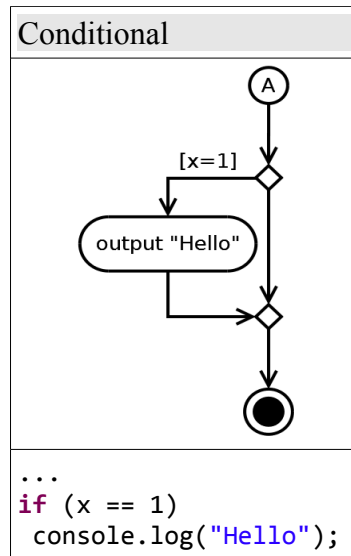
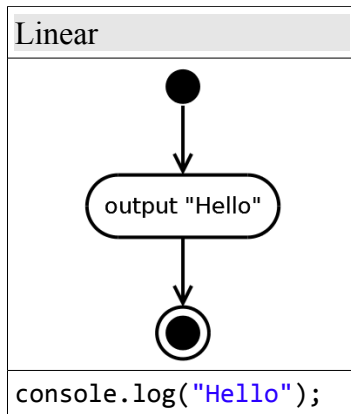
and their analogies in code (TypeScript)

Prof. Dipl.-Ing. Jirka R. Dell'Oro-Friedl
V1.1 ©HFU2019

1. Elements

	Startknoten (Initial Node)		Aktivität Aktionsknoten (ActionNode)
	Endknoten (ActivityFinalNode)		Daten Objektknoten (ObjectNode)
	Ablaufendknoten (FlowFinalNode)		Notiz (Note)
	Konnektor (Connector)		Zeitsignal (AcceptTimeEventAction)
	Entscheidung und Zusammenführung (DecisionNode / MergeNode)		Event Signalempfang (AcceptEventAction)
	Teilung / Synchronisation (ForkNode / JoinNode)		Event Signalversand (SendSignalAction)
	Aufruf		
	Kontrollfluss / Objektfluss ActivityEdge (ControlFlow / ObjectFlow)		

2. Basic flow structures



3. Loops

Loop (Pre-Test)

```

let i: number = 0;
while (i < 10) {
  console.log(i);
  i++;
}
oder
for (let i: number = 0; i < 10; i++)
  console.log(i);
  
```

Loop (Post-Test)

```

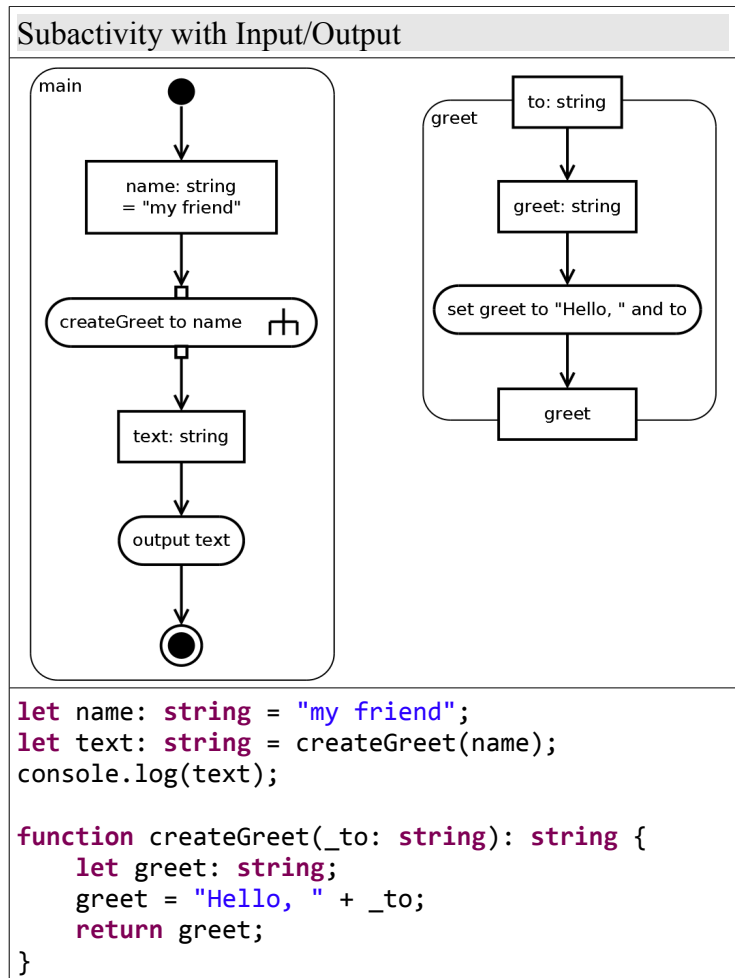
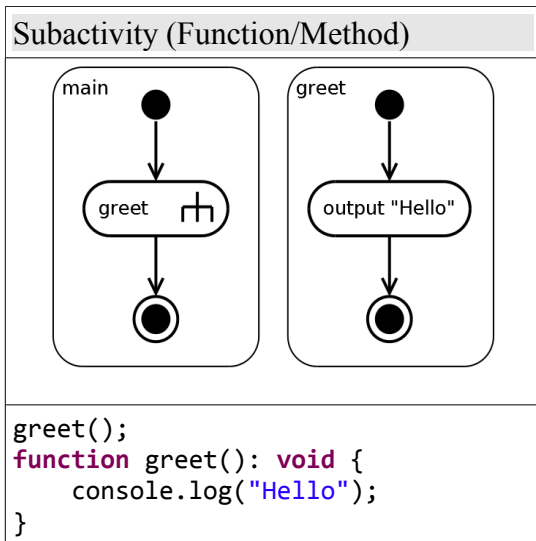
let i: number = 0;
do {
  console.log(i);
  i++;
} while (i < 10);
  
```

Loop with additional control conditions

```

for (let i: number = b; i > 1; i/=2)
{
  if (i == 3)
    continue;
  if (i == a)
    break;
  console.log(i);
}
  
```

4. Subactivities



5. Signals

