Concurrent Object Oriented Languages

JCSP

wiki.eecs.yorku.ca/course/6490A

JCSP

JCSP is a Java package based on CSP. It has been developed by Peter Welch (University of Kent). Information about JCSP can be found at the url `www.cs.kent.ac.uk/projects/ofa/jcsp`.

JCSP

The key difference with the version of CSP that we studied before: channels are used to communicate instead of process names.

Semaphore in CSP

With process names:

```
process ::
  semaphore ! P
  critical section
  semaphore ! V

semaphore ::
  val := 1;
  *[val > 0; process ? P -> val := val - 1;
    []       process ? V -> val := val + 1 ]
```

Semaphore in CSP

With channels:

```
P ! ()
critical section
V ! ()

val := 1;
*[val > 0; P ? () -> val := val - 1;
  []       V ? () -> val := val + 1 ]
```

Channels in JCSP

There are different types of channels including

One2OneChannel	:	one writer and one reader
Any2OneChannel	:	many writers and one reader
One2AnyChannel	:	one writer and many readers
Any2AnyChannel	:	many writers and many readers

These are all interfaces.

Channels in JCSP

The Channel class has factory methods to create those channels.

For example, `Channel.one2one()` creates a `One2OneChannel`.

Implement semaphore in JCSP

Question

Assume we have a single semaphore and multiple processes. What kind of channel do we need?

Implement semaphore in JCSP

Question

Assume we have a single semaphore and multiple processes. What kind of channel do we need?

Answer

`Any2OneChannel`: many writers (processes) and one reader (semaphore).

Implement semaphore in JCSP

Question

Assume we have a single semaphore and multiple processes. What kind of channel do we need?

Answer

`Any2OneChannel`: many writers (processes) and one reader (semaphore).

Question

How do we create such a channel?

Implement semaphore in JCSP

Question

Assume we have a single semaphore and multiple processes. What kind of channel do we need?

Answer

`Any2OneChannel`: many writers (processes) and one reader (semaphore).

Question

How do we create such a channel?

Answer

`Channel.any2one()`

Implement semaphore in JCSP

Question

How many of such channels do we need?

Implement semaphore in JCSP

Question

How many of such channels do we need?

Answer

Two:

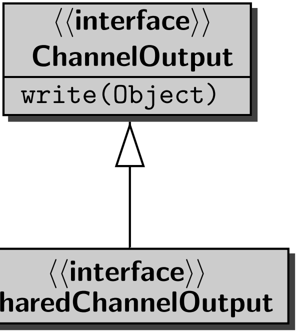
```
Any2OneChannel verhoog = Channel.any2one();
Any2OneChannel prolaag = Channel.any2one();
```

Channels in JCSP

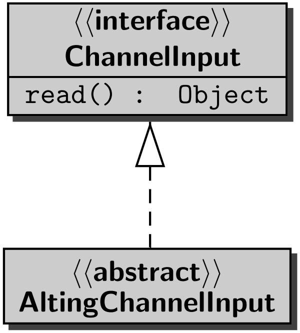
Each channel has an input end and an output end.

```
Any2OneChannel channel = Channel.any2one();
AltingChannelInput input = channel.in();
SharedChannelOutput output = channel.out();
```

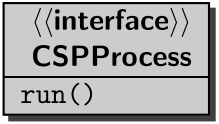
AltingChannelInput



SharedChannelOutput



CSPProcess



Process

The class `Process` implements the interface `CSPProcess`.

Question

Which attributes do we need to introduce?

Process

The class `Process` implements the interface `CSPProcess`.

Question

Which attributes do we need to introduce?

Answer

```
private ChannelOutput verhoog;
private ChannelOutput prolaag;
```

Process

The class `Process` implements the interface `CSPProcess`.

Question

Which attributes do we need to introduce?

Answer

```
private ChannelOutput verhoog;
private ChannelOutput prolaag;
```

Problem

Implement the constructor.

Process

The class `Process` implements the interface `CSPProcess`.

Question

Which attributes do we need to introduce?

Answer

```
private ChannelOutput verhoog;
private ChannelOutput prolaag;
```

Problem

Implement the constructor.

Problem

Implement the `run` method.

Semaphore

The class `Semaphore` implements the interface `CSPProcess`.

Question

Which attributes do we need to introduce?

Semaphore

The class `Semaphore` implements the interface `CSPProcess`.

Question

Which attributes do we need to introduce?

Answer

```
private int value;
private AltingChannelInput verhoog;
private AltingChannelInput prolaag;
```

Semaphore

The class `Semaphore` implements the interface `CSPProcess`.

Question

Which attributes do we need to introduce?

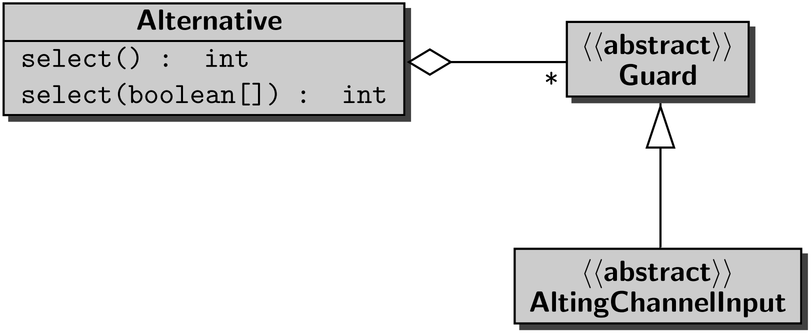
Answer

```
private int value;
private AltingChannelInput verhoog;
private AltingChannelInput prolaag;
```

Problem

Implement the constructor.

CSPProcess


Alternative
select() : int
select(boolean[]) : int
*
⟨⟨abstract⟩⟩
Guard
⟨⟨abstract⟩⟩
AltingChannelInput

run method of Semaphore class

Question

How many alternatives are there?

`run` method of `Semaphore` class

Question

How many alternatives are there?

Answer

Two.

run method of Semaphore class

Question

How many alternatives are there?

Answer

Two.

```
final int ALTERNATIVES = 2;

final int V = 0;
final int P = 1;

final Guard[] guard = new Guard[ALTERNATIVES];
```

`run` method of `Semaphore` class

Question

What are the guards?

`run` method of `Semaphore` class

Question

What are the guards?

Answer

```
guard[V] = this.verhoog;
guard[P] = this.verlaag;
```

`run` method of `Semaphore` class

Question

What are the guards?

Answer

```
guard[V] = this.verhoog;
guard[P] = this.verlaag;
```

Question

What are the preconditions?

`run` method of `Semaphore` class

Question

What are the guards?

Answer

```
guard[V] = this.verhoog;
guard[P] = this.verlaag;
```

Question

What are the preconditions?

Answer

```
final boolean[] precondition = new boolean[ALTERNAT
precondition[V] = true;
precondition[P] = this.value > 0;
```