

```
package org.agreement_technologies.service.map_heuristic;
```

```
import java.util.ArrayList;
```

```
import java.util.BitSet;
```

```
import org.agreement_technologies.common.map_heuristic.HPlan;
```

```
import org.agreement_technologies.common.map_heuristic.Heuristic;
```

```
public class CustomHeuristic implements Heuristic {
```

```
    @Override
```

```
    public void evaluatePlan(HPlan p, int threadIndex) { ...3 lines }
```

```
    @Override
```

```
    public void evaluatePlan(HPlan p, int threadIndex, ArrayList<Integer> achievedLandmarks) { ...3 lines }
```

```
    @Override
```

```
    public void evaluatePlanPrivacy(HPlan p, int threadIndex) { ...3 lines }
```

```
    @Override
```

```
    public void waitEndEvaluation() { ...3 lines }
```

```
    @Override
```

```
    public void startEvaluation(HPlan basePlan) { ...3 lines }
```

```
    @Override
```

```
    public Object getInformation(int infoFlag) { ...3 lines }
```

```
    @Override
```

```
    public boolean supportsMultiThreading() { ...3 lines }
```

```
    @Override
```

```
    public boolean requiresHLandStage() { ...3 lines }
```

```
    @Override
```

```
    public int numGlobalLandmarks() { ...3 lines }
```

```
    @Override
```

```
    public ArrayList<Integer> checkNewLandmarks(HPlan plan, BitSet achievedLandmarks) { ...3 lines }
```

```
}
```