# Implementatietemplate SOLID – Liskov Substitution Principle (LSP)

## Toelichting

Geef in 3–5 zinnen aan hoe jullie hebben voorkomen dat het Liskov Substitution Principle werd geschonden bij het uitvoeren van deze user story ( het inbouwen van een HintJoker en een KeyJoker

|  |
| --- |
| De speler heeft een HintJoker en een KeyJoker. SpecialJoker extend van Joker, dat wil zeggen dat een SpecialJoker wordt bezien als een Joker. DailyScrumKamer en SprintReviewKamer kunnen SpecialJokers en Jokers accepteren. Alle andere kamers kunnen alleen Jokers accepteren. Dit voorkomt dat er if statements gemaakt moeten worden om te kijken of een Joker of SpecialJoker wordt meegegeven. |

## Geef de betrokken interfaces, classes en gebruik.

|  |  |
| --- | --- |
| **Interface of classnaam** | **Rol van deze interface/ class in deze user story** |
| Joker | Joker is een interface voor alle kamers |
| SpecialJoker | SpecialJoker is een interface voor de DailyScrumKamer en SprintReviewKamer. SpecialJoker extend Joker. Hierdoor hoeven er geen if statements gebruikt te worden. |
| HintJoker | Speler heeft alleen maar 1 HintJoker. HintJoker geeft een handige hint aan de speler |
| KeyJoker | Speler heeft 1 KeyJoker. KeyJoker maakt de deur gelijk open voor de speler zodat de speler niet meer hoeft te beantwoorden |
|  |  |

## Inspanning per student

Geef aan per student wat zijn of haar bijdrage was in het realiseren van dit SOLID-principe.

|  |  |
| --- | --- |
| **Naam student** | **Bijdrage** |
| Dami Kastaneer | Liskov toegepast in code en geimplementeerd |
| Anson Mok | Code zodanig aangepast dat Jokers in de inventaris worden opgeslagen |
| Nathaniel Rasmijn | Code getest |
|  |  |
|  |  |