

Table of Contents

GRASP

3

GRASP

General Responsibility Assignment Software Patterns (GRASP) consist of guidelines for assigning responsibility to classes and objects in object-oriented design.

Snippet from [Wikipedia](#): **GRASP (object-oriented design)**

General Responsibility Assignment Software Patterns (or **Principles**), abbreviated **GRASP**, is a set of "nine fundamental principles in object design and responsibility assignment"⁶ first published by Craig Larman in his 1997 book *Applying UML and Patterns*.

The different patterns and principles used in GRASP are controller, creator, indirection, information expert, low coupling, high cohesion, polymorphism, protected variations, and pure fabrication. All these patterns solve some software problems common to many software development projects. These techniques have not been invented to create new ways of working, but to better document and standardize old, tried-and-tested programming principles in object-oriented design.

Larman states that "the critical design tool for software development is a mind well educated in design principles. It is not UML or any other technology."²⁷² Thus, the GRASP principles are really a mental toolset, a learning aid to help in the design of object-oriented software.

[Creative Commons Attribution-Share Alike 4.0](#)

[kb](#), [programming](#)

From:

<https://almbok.com/> - **ALMBoK.com**

Permanent link:

<https://almbok.com/kb/grasp>

Last update: **2022/08/10 06:33**

