-Namenskonventionen für Objekte und Räume

-existieren mehrere gleiche Dinge: Durchnummerierung

<Name>\_<Nummer>

-Namen aus Spezifikation/Lastenheft

-keine Namen vorhanden 🡪 sinnvolle Namen (ähnlich wie andere)

-Leerzeichen als ‘\_‘

-printed name mit Leerzeichen

-eindeutige Namen für Variablen (Bsp.: Zähler für Schritte: Schrittzaehler)

-Kommentieren an bestimmten Stellen

-siehe Kommentierkonvention

-Segmente (Definitionen, Kontaminierte, …) mit [\*\*\*\*\*<Name>\*\*\*\*\*] kennzeichnen

(-darunter einrücken: gilt für Mehrfach-Definitionen: Definitionen, Aktionen, …)

-Block mit [<Name>] kennzeichnen

-Code möglichst so schreiben, dass keine Kommentare notwendig sind

-Einrücken von zusammengehörenden Dingen und Blöcken

-zusammengehörende Dinge zusammenschreiben

-zusammengehörende Blöcke und Quelltext

-Definitionen, Aktionen, … zusammenschreiben

-Platz lassen zwischen Code-Segmenten und -Blöcken

-zwischen Blöcken 1-2 Zeilen

-zwischen Segmenten 3-4 Zeilen

-ähnliche Code-Blöcke sollen gleich aussehen

-orientieren an existierenden Blöcken

-keine doppelten Dinge 🡪 Wiederverwertung

-nicht mehrere Anweisungen (=Sätze) in einer Zeile

-Ausnahme: Objektdefinition und printed name

-Groß- und Kleinschreibung beachten (Satzanfang groß, Objekte und Eigennamen groß)

-globale Variablen zuordnen:

-nur von einem Segment genutzt: zu Segment

-dürfen von anderen Segmenten nicht genutzt werden

-von mehreren Segmenten: nach ganz oben

-Code-Priorisierung (wenn möglich auch 2.)

-1.funktionierend und übersichtlich

-2.so kurz wie möglich

-vor Abgabe des Codes: Testen

Blöcke: eine große Anweisung

Schleife, mehrere Bedingungen, every turn, before …

Segmente: Code und ein oder mehrere Blöcke im sinnlichen Zusammenhang

Bsp.: Programmierung Kontaminierter