Abiturprüfung ab 2019	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

Zuweisung	Code (C-Notation)	
dieVariable ← derAusdruck	dieVariable = derAusdruck;	
dieVariable := derAusdruck		
Sequenz		
Anweisung1	Anweisung1;	
Anweisung2	Anweisung2; Anweisung3;	
Anweisung3	Anweisungs,	
Auswahl		Sequenzdiagramm
einseitige Auswahl		
Bedingung erfüllt	if (Bedingung)	opt [Bedingung]
a nein	A	
Anweisung(en)	Anweisung(en); }	
zweiseitige Auswahl	,	
Bedingung erfüllt	if (Bedingung)	alt [Bedingung]
a nein	{ Anyoioung(on)1:	
Anweisung(en)1 Anweisung(en)2	Anweisung(en)1; }	
logischer Ausdruck	else	[else]
vahr falsch	{ Anweisung(en)2;	
Anweisung(en)1 Anweisung(en)2	}	
Mehrfachauswahl	switch (Selektor)	alt [Selektor = Fall 1]
Selektor =	{	
Fall 1 Fall 2 Fall 3 sonst	case Fall1: Anweisung1; break; case Fall2: Anweisung2; break;	[Selektor = Fall 2]
Anw. 1 Anw. 2 Anw. 3 Anw. 4	case Fall3: Anweisung3: break;	[Selektor = Fall 3]
	default: Anweisung4;	
Wiederholung (Iteration)	}	[Gibe]
Schleife mit Eintrittsbedingung		
solange Bedingung erfüllt	while (Bedingung)	loop [Bedingung]
Anweisungen(en)	`Anweisung(en);	
Allweisungen(en)	}	
Schleife mit Austrittsbedingung	do	ſ. T
Anweisung(en)	{	loop
solange Bedingung erfüllt	Anweisung(en); }	
solarige bedingurig erruiit	while (Bedingung);	[Bedingung]
Zählschleife	for (i=0) i	
für i:=0 bis n, Schritt: +1	for (i=0; i<=n; i=i+1) {	loop [für i:=0 bis n, Schritt: +1]
Anweisung(en)	Anweisung(en);	[iui io bis ii, ociiiiii. +i]
/o.oag(o11)	}	
Schleife mit Abbruchmöglichkeit		
für i:=n bis 0, Schritt: -1	for (i=n; i>=0; i=i-1)	loop [für i:=n bis 0, Schritt: -1]
	for (i=n; i>=0; i=i-1) {	
Logischer Ausdruck		
wahr falsch	if (logischer Ausdruck) break;	break [logischer Ausdruck]
	}	1 1

Abiturprüfung ab 2019	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

3.2 Klassen, Datentypen und Notationen

Klasse

Analysesichten

Klasse attribut operation()

Klasse

AbstrakteKlasse	
abstrakteOperation()	

AbstrakteKlasse {abstract}	
abstrakteOperation() {abstract}	

Bedeutung Datentypen und Abkürzungen

Boolsches Attribut: Boolean, bool GZ, Integer, int Ganzzahlattribut: FKZ, Real, double Fließkommaattribut: Zeichenattribut: Zeichen, char Textattribut: Text, string Währungsattribut: Geld, currency Datumsattribut: Datum, date Zeitattribut: Zeit, time

(auch Basisdatentypen der Programmiersprachen sind erlaubt)

Designsicht

Klasse	
- privatesAttribut:Typ {Zusicherung}	
# geschütztesAttribut:Typ	
# geschütztesAttributMitAnfangswert:Typ = Anfangswert	
- feldAttribut1[0maxldx]:Typ	
- feldAttribut2[anzElemente]:Typ	
feldAttribut3:Typ[anzElemente]klassenAttribut:Typ	
- / abgeleitetesAttribut:Typ	
31	
+ Klasse() «constructor» + Klasse(parameter:Typ) «constructor»	
+ ~Klasse() «destructor»	
+ operation(parameter:Typ)	
+ operation1(parameter:Typ):Ergebnistyp[]	
+ operation2(parameter1:Typ1, parameter2:Typ2)	
+ operation2(parameter1:Typ1=Wert):Ergebnistyp	
- privateOperation()	
# geschützteOperation()	
+ öffentlicheOperation()	
+ klassenOperation():Typ	

Attribute

Die Variablen einer Klasse heißen **Attribute.**Der Bezeichner eines Attributs beginnt in der UML mit einem Kleinbuchstaben.

Operationen

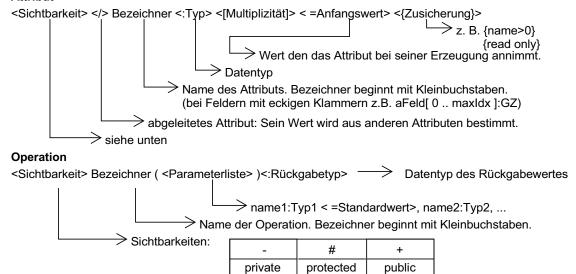
Die Funktionen bzw. Prozeduren der Programmiersprache heißen in der UML **Operationen.** Der Bezeichner einer Operation beginnt in der UML mit einem Kleinbuchstaben.

Bezeichner von Operationen sollen mit einem Verb beginnen.

Notation

<optionale Bestandteile> in spitzen Klammern

Attribut



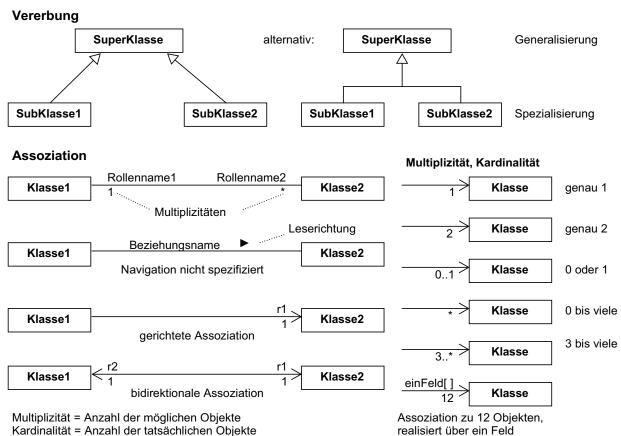
privat

geschützt

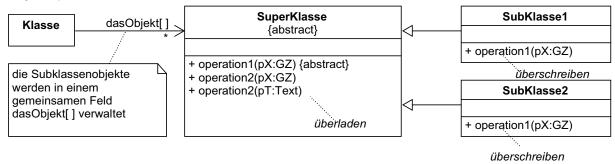
öffentlich

Abiturprüfung ab 2019	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

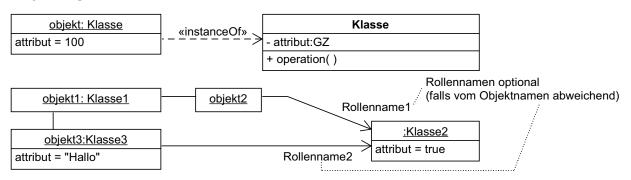
3.3 Klassendiagramm, Vererbung, Assoziation, Polymorphie, Objektdiagramm



Polymorphie

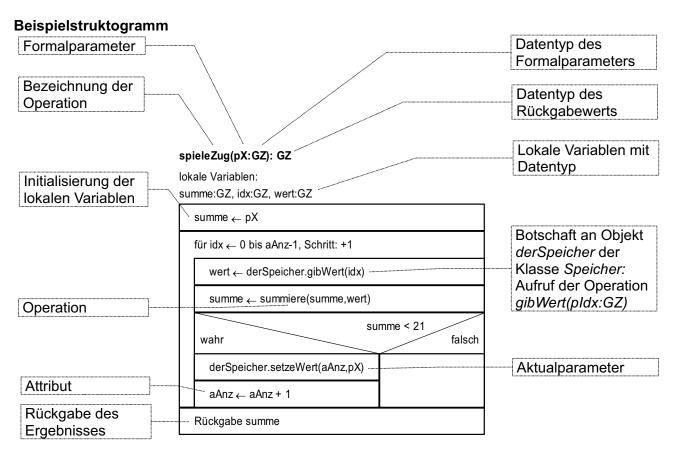


Objektdiagramm

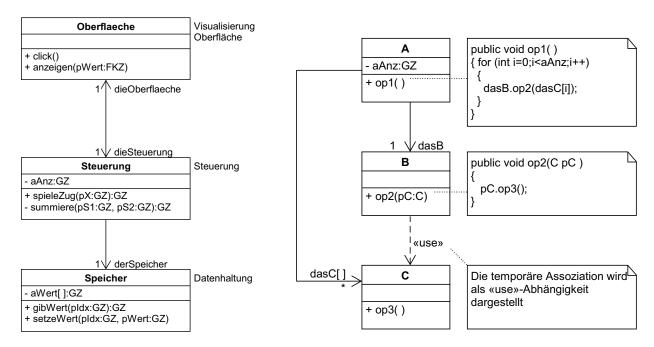


Abiturprüfung ab 2019	Berufliches Gymnasium (TG)
Formelsammlung	1.5.2 Informationstechnik

3.4 Beispiele



Beispielklassendiagramme



Klassendiagramm 3-Schichtenarchitektur

Klassendiagramm mit temporärer Assoziation