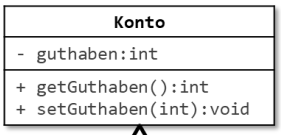
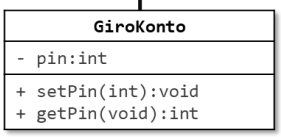
- Konstruktoren werden **nicht** Vererbt. Sie müssen gegebenenfalls implementiert werden.
- Beim Erzeugen eines Unterklasseobjektes wird auch ein Oberklasseobjekt angelegt.
- Der Compiler erzeugt (**falls kein Konstruktor vorhanden**) den Default-Konstruktor und **in jedem** Konstruktor der kein **`this()`**-Aufruf hat den Aufruf des Default-Konstruktors **`super()`** der Basisklasse ein

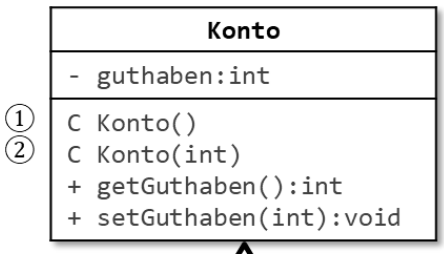
Nr	Quellcode	Beschreibung - Hinweise

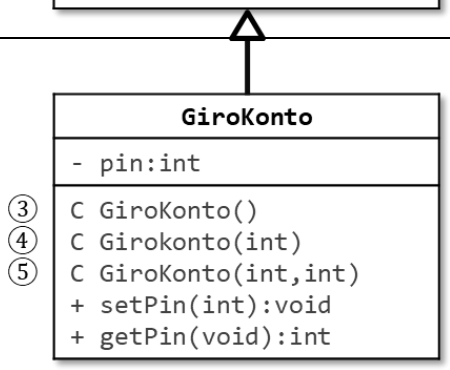
	`public class Konto {`
	`  public int guthaben;`
	`  :`
	 
		`public class Konto extends Object {`
		`  public int guthaben;`
		`  public Konto() {`
		`  super();`
		`  }`
		`  :`

	`public class GiroKonto`
	`  extends Konto {`
	`  public int pin;`
	`  :`
	 
		`public class GiroKonto extends Konto {`
		`  public int pin;`
		`  public GiroKonto() {`
		`  super();`
		`  }`
		`  :`

Mit **`super()`** bzw **`super(`**Paramater,…**`)`** kann der Konstruktor der direkten Oberklasse aufgerufen werden.

Nr	Quellcode	Beschreibung - Hinweise

	`public class Konto {`
	`  public int guthaben;`
	① `  public Konto() {`
	`  this(500); }`
	② `  public Konto(int guthaben) {`
	`  super() // automatisch Object`
	`  this.guthaben = guthaben; }`
	`  :`
		

	`public class GiroKonto extends Konto {`
	`  public int pin;`
	③ `  public GiroKonto() {`
	`  this(9999); }`
	④ `  public GiroKonto(int pin) {`
	`  super() // automatisch Konto`
	`  this.pin = pin; }`
	⑤ `  public GiroKonto(int pin, int start) {`
	`  super(start);`
	`  this.pin = pin; }`
	`  :`
		

	`Konto karl = new Konto();`
	`Konto kurt = new Konto(500);`
	`GiroKonto gerd = new GiroKonto();`
	`GiroKonto gido = new GiroKonto(777);`
	`GiroKonto gisa = new GiroKonto(777,200);`
		① ② *(super() Object)*
		② *(super() Object)*
		③ ④ *(① super() Konto) ②(super() Object)*
		④ *(① super() Konto) ②(super() Object)*
		⑤ ② *(super() Object)*

Empfehlung: Immer mindestens den leeren Default-Konstruktor schreiben !

Wichtige Eigenschaften und Hinweise

- Konstruktoren werden nicht vererbt und müssen ggf. implementiert werden.
- Der Aufruf eines Konstruktors der vererbenden Klasse aus der erbenden Klasse kann mittels super() erfolgen(super() muss die 1. Anweisung im Konstruktor der erbenden Klasse sein).
- Der Aufruf eines anderen Konstruktors der eigenen Klasse kann mittels this() erfolgen. (this() muss die 1. Anweisung im aufrufenden Konstruktor sein)
- this() und super() nicht gleichzeitig verwendet werden können.
- Die Systemklasse Objekt ist die oberste Klasse, so dass alle Klassen (automatisch) von Objekt erben.
- Zyklische (wechselseitige) Konstruktor-Weiterleitungen sind nicht möglich (A -> B , B -> C und C -> A).