



Name: _____ Prüf-Account _____

Bitte achten Sie darauf, nur kompilierfähige Programme abzugeben. Sollten Sie einen Compilerfehler nicht beheben können, kommentieren Sie das entsprechende Codestück aus. Nicht kompilierfähige Programme werden negativ bewertet.

Aufgabe 1 Implementierung der Klasse *Computer*

Dieses Beispiel modelliert einen Computer mit diversen Eigenschaften:

Computer
- modell: String = "Lenovo Thinkpad" - ssd: boolean = false - memory: int = 4
+ Computer() + Computer(modell: String, ssd: boolean, memory: int) + getModell(): String + setModell(modell: String): void + hasSSD(): boolean + getMemory(): int + upgradeToSSD(): boolean + memoryUpgrade(gb: int): int + toString(): String + print(): void

Implementieren Sie die Klasse **Computer** des UML-Diagrammes. Das Modell soll nicht **null** sein, gültige Werte für den Hauptspeicher (**memory**) sind 1 bis 32 GB.

Falls er Computer noch keine SSD-Festplatte hat (sondern eine HDD) kann ein Upgrade auf SSD durchgeführt werden (Methode **upgradeToSSD**). Hat der Computer schon eine SSD, solle eine Fehlermeldung ausgegeben werden.

Bei **memoryUpgrade** wird übergeben, wieviele GB Hauptspeicher hinzugefügt werden sollten (mindestens 1 GB). Der Rückgabewert gibt an, wieviele GB tatsächlich hinzugefügt wurden (32 GB dürfen nicht überschritten werden!).

Die Methode **print** soll den mittels **toString** erzeugten String auf der Konsole ausgeben. Die Formatierung soll wie folgt aussehen, falls **modell="Lenovo"**, **ssd=true** und **memory=8**:

Marke: **Lenovo** (SSD), 8 GB RAM

Im Falle von **marke="Hewlett-Packard"**, **ssd=false** und **memory=16**:

Marke: **Hewlett-Packard** (HDD), 16 GB RAM

Gutes Gelingen!