

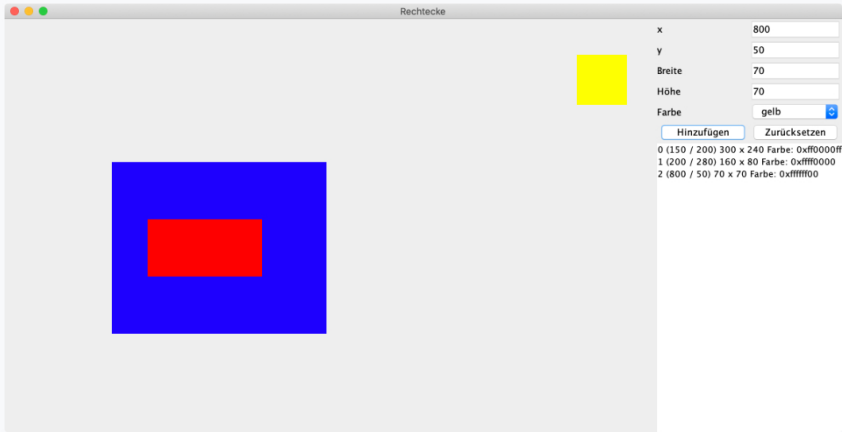
Ü 5b.2: Figuren

Done: Make a submission

Done: Receive a grade

Due: Sunday, 18 December 2022, 11:00 PM

Gegeben ist ein Programm, das es ermöglicht auf einer Oberfläche beliebig viele Rechtecke zu zeichnen:



Der Link für das Github-Repository dafür lautet: <https://classroom.github.com/a/kz8PkjTQ>

Nun soll es ermöglicht werden, statt nur Rechtecken, auch andere Figuren zu zeichnen, wie z.B. Kreise, Linien, Dreiecke, Erstelle dazu eine Klasse `Figur`, die als Superklasse von all diesen Figuren fungieren kann. Sinnvoller Weise soll zu diesem Zweck die Klasse `Figur` folgende Attribute und Methoden besitzen (das sind die Attribute und Methoden, die wirklich jede Figur hat - breite und höhe ist nicht für jede Figur sinnvoll):

- `x`-Koordinate: ganzzahliger Wert
- `y`-Koordinate: ganzzahliger Wert
- `farbe`: `color`-Objekt
- passende Konstruktoren
- setter und getter-Methoden
- `draw`: Methode mit einem `graphics`-Objekt als Parameter, die in der Klasse `Figur` nichts tut.
- `toString`

Damit dieses Programm auch entsprechend den objektorientierten Richtlinien unter Ausnutzung der Polymorphie arbeitet, sollen folgende Punkte berücksichtigt werden:

- Die Klasse `Figur` ist nur die Basis, aber es soll nicht möglich sein eine allgemeine Figur auszuwählen und zu erzeugen. Dementsprechend soll auch die Methode `draw` in der Klasse `Figur` keinen Inhalt haben.
- Es soll neben Rechteck zumindest Dreieck, Kreis und Linie als Subklassen zu `Figur` geben, die auch eine konkrete Umsetzung der Methode `draw` haben.
- Die GUI soll entsprechend angepasst werden, dass die neuen Figur-Arten ebenfalls hinzugefügt werden können. Die GUI verwendet jedoch ausschließlich die `Figur`-Klasse als Referenz! Lediglich beim Hinzufügen muss zwischen den genauen Figur-Arten unterschieden werden

Erstelle auch ein UML-Klassendiagramm zu deinem Programm und gib dieses hier ab.

Abgabe:

Eine Aufgabe gilt erst und nur dann als erfolgreich abgegeben, wenn folgende Bedingungen erfüllt sind:

- Auf dem Github-Projekt sind zumindest 5 commits in regelmäßigen Abständen. Zumindest 1 commit ist am Ende der Schulstunde.
- Die Klassen stecken alle im Package `nachname.Figur` wobei die Namen ohne Umlaute in Kleinbuchstaben angegeben werden müssen.
- die Applikation ist mitsamt Sourcecode in einem Jar-Archiv verpackt.
- das Jar-Archiv ist ausführbar und enthält einen Unterdorder doc, der die JavaDocs zu allen Klassen enthält.

Submission status

Submission status	Submitted for grading	
Grading status	Graded	
Time remaining	Assignment was submitted 32 days 9 hours late	
Last modified	Friday, 20 January 2023, 8:59 AM	
File submissions	<div>Figuren.jar20 January 2023, 8:59 AM</div>	
Submission comments	<div>Comments (0)</div>	

Grading criteria

Programmierstil	Programmierstil wenig berücksichtigt 0 points	Einrückungen und Kommentare vorhanden aber verbesserungswürdig 1 points	Einrückungen und Kommentare gut gemacht. 2 points
Funktion	Funktion mangelhaft 0 points	Funktion mit leichten Fehlern ok 2 points	Funktion vollständig 4 points
Abstrakte Klasse und Polymorphie	Figur nicht als als abstrakte Klasse umgesetzt. Polymorphie nicht angewendet. 0 points	Figur hat keine abstrakte Methode und/oder Polymorphie nicht sinnvoll angewendet. 3 points	Figur mit abstrakter Methode umgesetzt und Polymorphie sinnvoll angewendet. 6 points
Commits/JAR	Zu wenig Commits und/oder keine sinnvollen Messages. JAR nicht vorhanden. 0 points	Commit-Anzahl gerade ausreichend und/oder kaum sinnvolle Messages und/oder JAR nicht vorhanden. 1 points	Ausreichende Anzahl an Commits mit sinnvollen Messages. JAR vorhanden und ausführbar. 2 points
Abgabegespräch	Abgabegespräch mangelhaft 0 points	Fragen und Änderungen im Programm konnten mit leichten Unsicherheiten	Fragen und Änderungen im Programm konnten gut beantwortet werden.

		beantwortet werden. 1 points	2 points
Zeit	Nachfrist 0 points	Knapp verspätet abgegeben. 1 points	Im zeitlichen Rahmen abgegeben. 2 points

Feedback

Grade	16.00 / 18.00
Graded on	Friday, 20 January 2023, 9:18 AM
Graded by	RW Rateiner-Magor Walter

Grade breakdown

Programmierstil	Programmierstil wenig berücksichtigt 0 points	Einrückungen und Kommentare vorhanden aber verbesserungswürdig 1 points	Einrückungen und Kommentare gut gemacht. 2 points
Funktion	Funktion mangelhaft 0 points	Funktion mit leichten Fehlern ok 2 points	Funktion vollständig 4 points
Abstrakte Klasse und Polymorphie	Figur nicht als als abstrakte Klasse umgesetzt. Polymorphie nicht angewendet. 0 points	Figur hat keine abstrakte Methode und/oder Polymorphie nicht sinnvoll angewendet. 3 points	Figur mit abstrakter Methode umgesetzt und Polymorphie sinnvoll angewendet. 6 points
Commits/JAR	Zu wenig Commits und/oder keine sinnvollen Messages. JAR nicht vorhanden. 0 points	Commit-Anzahl gerade ausreichend und/oder kaum sinnvolle Messages und/oder JAR nicht vorhanden. 1 points	Ausreichende Anzahl an Commits mit sinnvollen Messages. JAR vorhanden und ausführbar. 2 points
Abgabegespräch	Abgabegespräch mangelhaft 0 points	Fragen und Änderungen im Programm konnten mit leichten Unsicherheiten beantwortet werden. 1 points	Fragen und Änderungen im Programm konnten gut beantwortet werden. 2 points
Zeit	Nachfrist 0 points	Knapp verspätet abgegeben. 1 points	Im zeitlichen Rahmen abgegeben. 2 points