

EINFACHER PLOTTER, TEIL I: GRID

In diesem Aufgabenteil geht es darum, hardgecodete Funktionen zu plotten. Dies dient dazu, eine gute Umgebung zu schaffen, um dann den Parsing-Algorithmus Stück für Stück implementieren zu können. Du kannst dann nach jeder Teilaufgabe immer einen Plot sehen, der auf unterschiedliche Weise erzeugt wird.

Falls du irgendwo Probleme hast, zögere bitte nicht, um Hilfe zu fragen! Wenn du irgendwo hängst, helfen wir dir gerne weiter! Melde dich bei einem der zuständigen Betreuer*innen: Lorenz Braun, Fabian Sand, Robin Fritz, Furkan Keserci, Julia Kisela. Wir sind über Discord, Mattermost und E-Mail erreichbar. Wen könnt ihr für Hilfe ansprechen? (E-Mail, Mattermost, Discord)

- **Robin Fritz:** frro1020@h-ka.de, @robin.fritz, Snobo#4754
- **Julia Kisela:** kiju1012@h-ka.de, @julia.kisela, lohikäärme#3336
- **Furkan Keserci:** kefu1011@h-ka.de, @furkan, Killnexus#9999
- **Lorenz Braun:** brlo1014@h-ka.de, @lorenzbraun, MrRaptorious#7632
- **Fabian Sand:** safa1017@h-ka.de, @safa1017, Fabian S#7613

LERNMATERIAL

Wie zeichnet man eigentlich eine Funktion? Was sich einfach anhören mag, ist auch in der Realität nicht sonderlich schwierig. Die Idee ist, Punkte zu berechnen und diese dann durch Linien zu verbinden. (Wie man einfache gerade Linien zeichnet, hast du in den ersten Übungsblättern gelernt.) Mit etwas Feintuning kommen da ganz akzeptable Funktionsgraphen dabei heraus. Du solltest dir auch über folgende Frage Gedanken machen: Wie skaliere ich die Achsen gleichmäßig?

AUFGABENSTELLUNG

Wichtiger Hinweis: Im bereitgestellten Code finden sich Dokumentationskommentare, welche Aufschluss auf die Funktionsweise oder die zu implementierende Funktionalität geben. Bitte deshalb den Code und die Kommentare aufmerksam studieren! Bitte verändere gegebene Klassennamen und Methodennamen nicht, da diese für die Tests verwendet werden!

Lade dir aus dem Ilias das Eclipse-Projekt herunter und öffne es auf deinem Gerät. In ihm findest du schon eine Klasse Grid, eine Klasse Expression und eine HardCoded-Beispiel-Klasse.

In der Grid-Klasse sollen nun folgende Methoden implementiert werden, Methodensignaturen und Javadoc ist dafür im Code gegeben:

- Methode zum Zeichnen eines Koordinatensystems (coordinates)
- Methode zum Beschriften der Achsen (labels)

- Methode zum Zeichnen von beliebigen Funktionen (plot). (Probiere mal einen $\tan(x)$ mit deinem Plotter, was passiert bei den Polstellen? Siehst du einen einfachen Weg, dies richtigzustellen?)

Lies die Beispiel-Klasse `HardcodedSin` genau durch und schreibe noch drei weitere `HardCoded`-Klassen der gleichen Art. Mit diesen kannst du dann deine `plot`-Methode ausprobieren. `HardcodedSin` erbt von `Expression`, dies ist an dieser Stelle noch nicht relevant, macht aber, dass du später dieselbe `Grid`-Klasse für alle Aufgabenteile benutzen kannst.

Rufe nun in der `paint`-Methode der `Grid`-Klasse die Methode zum Zeichnen für die vier `HardCoded`-Funktionen auf, um Plots zu erzeugen. Am Ende des Blattes findest du Referenzbilder wie es ungefähr aussehen könnte.

CHECKLISTE

- Du hast ein Koordinatensystem gezeichnet
- Es ist richtig beschriftet
- Es zeichnet Funktionen korrekt darin ein
- Schreibe drei weitere `HardCoded`-Klassen
- Plote deine drei `HardCoded`-Klassen in `Grid`

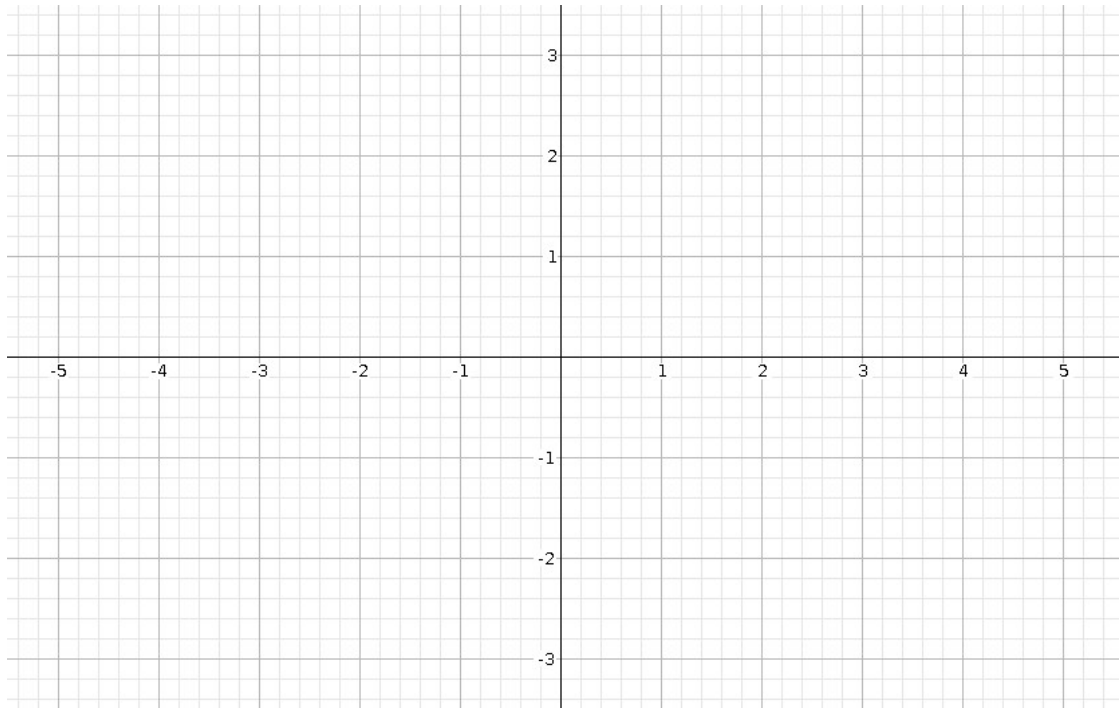


Abbildung 1: Leeres Koordinatensystem

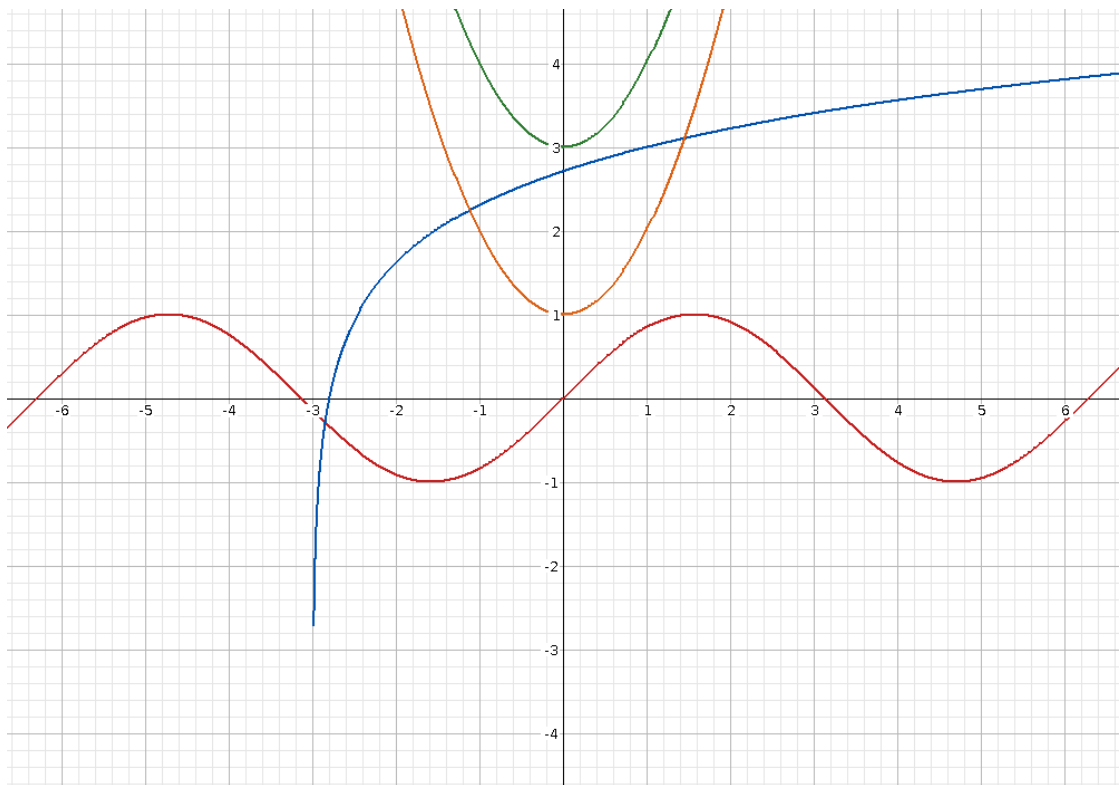


Abbildung 2: Mehrere Beispielfunktionen.