



Nr.	Aufgabenstellung	benötigte Materialien
1.	Lösung zum Struktogramm für das Binäre Suchen kontrollieren und verstehen.	PDF „Struktogramm binaeres Suchen“
2.	<u>Sortiervverfahren:</u> <ul style="list-style-type: none"> • Informiert euch über die Theorie der drei Sortiervverfahren „Selectionsort“, „Bubblesort“ und „Insertionsort“. 	<ul style="list-style-type: none"> • Videos Sortiervverfahren.txt • PDF „Bubblesort“, „Selectionsort“, „Insertionsort“
3.	<ul style="list-style-type: none"> • Erstelle ein Struktogramm für eines der drei Sortiervverfahren. 	
4.	<ul style="list-style-type: none"> • Implementiert alle drei Sortiervverfahren. Benutzt dafür das BlueJ Projekt „Sortieren_Vorlage“. <p>Nähere Erläuterungen zum Vorgehen unten.</p>	Ordner „Sortieren_Vorlage“
5.	<ul style="list-style-type: none"> • Kontrolle/Hilfe für die Implementierung der Sortieralgorithmen <p>Nähere Erläuterungen zum Vorgehen unten.</p>	PDF „Sortier_Lösungen“
6.	<u>2-Dimensionale-Arrays</u> <ul style="list-style-type: none"> • Informiert euch über die Deklaration, Erstellung und Wertzuweisung bei 2-D-Arrays. • Erstellt eine neue Klasse „zweiD“ und erstellt in dieser ein eigenes 2-Dimensionales-Array mit dem Datentyp „int“ und beliebigen Werten. • Schreibt eine Methode „public void ausgabe()“, welche die Werte des 2-D-Arrays in der Konsole übersichtlich ausgibt. 	<ul style="list-style-type: none"> • https://www.youtube.com/watch?v=ROYzQPBusAg • https://programmieren-starten.de/blog/java-tutorial-15-mehrdimensionale-arrays/
7.	<p>Ein 2-D-Array kann als Spielfläche eines einfachen Spiels mit zwei „Spielfiguren“ dienen. Hier die Anforderungen an das Spiel:</p> <ul style="list-style-type: none"> • Erstellt eine neue Klasse mit dem Namen „Spielfeld“. 	



	<ul style="list-style-type: none"> • Erstellt ein 2-D-Array der Größe 5x5 und vom Datentyp „int“. Dieses wird das „Spielfeld“ sein. • Alle Felder auf dem Spielfeld werden mit dem Wert 0 initialisiert. Eine 0 bedeutet dabei, dass das Feld nicht von einer Spielfigur besetzt ist. • Die erste Spielfigur wird mit dem Wert 1 auf dem Spielfeld dargestellt. Die Startposition soll oben links sein [0][0]. • Die zweite Spielfigur wird mit dem Wert 2 auf dem Spielfeld dargestellt. Die Startposition soll unten rechts sein [4][4]. • Durch die Methoden <code>public void links(int figur)</code>, <code>public void rechts(int figur)</code>, <code>public void oben(int figur)</code>, <code>public void unten(int figur)</code> wird eine der Spielfiguren auf dem Spielfeld um jeweils ein Feld bewegt. • Wird die Spielfeldgrenze erreicht, soll die Figur sich nicht weiterbewegen und einen Hinweis ausgeben, dass die Begrenzung des Spielfelds erreicht ist. 	
--	---	--

zu 4.:

Öffnet das Projekt. Jedes Sortiervverfahren hat seine eigene Klasse. Programmiert die Sortiervverfahren jeweils nur in der ersten Methode der Klasse. Also z.B. muss die Lösung für den Bubblesort Algorithmus in der Klasse „Bubblesort“ und dann in der Methode „`public int [] bubblesort (int [] sortieren)`“ implementiert werden.

Ändert also bitte nichts in der „main“-Methode. Die Klassen sind so konzipiert, dass ihr sie einfach starten könnt und seht, ob eure Implementierung funktioniert hat.

zu 5.:

Die Lösungen der drei Sortiervverfahren sind im PDF „Sortier_Lösungen“ zu finden. Dort ist aber nicht angegeben welcher Code zu welchem Sortieralgorithmus passt.

Dieses Dokument ist nur zur Hilfe gedacht bzw. nur zu verwenden, wenn keine eigene Lösung möglich ist.