

Name: _____

Mat.Nr.: _____

Aufgabe 1

Eine Fluggesellschaft legt für ihre Flüge Sitzplatzkontingente für verschiedenen Preise fest. Implementieren Sie die Klasse *Flug* mit einem Konstruktor

Flug(),

der einen Flug ohne verfügbare Sitzplätze anlegt, und einer Methode

addPlaetze(int anzahl, int preis),

die die angegebene Anzahl von Sitzplätzen für den angegebenen Preis zum Flug hinzufügt. Implementieren Sie in *Flug* auch die Methode

int getBestPreis(),

die den Preis des billigsten verfügbaren Sitzplatzes zurückliefert, und die Methode

boolean bucheSitz(int preis),

die einen Sitzplatz für den angegebenen Preis bucht. Ist ein solcher Sitzplatz nicht verfügbar, soll die Methode *false* zurückliefern, ansonsten *true*.

Aufgabe 2

Implementieren Sie die Methode

int[] getBestTriple(int[] a),

die ein Array zurückliefert, das genau jene 3 aufeinander folgende Elemente von *a* enthält, deren Summe maximal ist. Wenn das Array *a* weniger als 3 Elemente enthält, soll *null* zurückgeliefert werden.

Beispiel: Für das Array {41, 14, 38, 3, 22, 49, 38, 9, 16, 37} soll {22, 49, 38} zurückgeliefert werden.

Hinweis: Achten Sie darauf, nicht mit einem ungültigen Index auf das Array *a* zuzugreifen.

Aufgabe 3

Implementieren Sie in einer Klasse *Packen* die Methode

boolean istPackbar(int[] gewichte, int maxAnzahl, int maxGewicht),

die überprüft, ob die Gegenstände mit den in *gewichte* angegebenen Gewichten in *maxAnzahl* Kisten gepackt werden können, ohne dass in eine Kiste mehr als *maxGewicht* gepackt wird.

Beispiel: Für *gewichte*={4, 2, 4, 2, 3} liefert *istPackbar(gewichte, 3, 6)=true*, *istPackbar(gewichte, 2, 6)=false*, und *istPackbar(gewichte, 3, 5)=false*.