

Name: _____

Mat.Nr.: _____

Aufgabe 1

In einer Fabriksimulation soll die Klasse *AssemblyMaschine* implementiert werden. In dieser werden zwei Zwischenprodukte zu einem Endprodukt zusammengesetzt und dieses wird an ein Lager weitergereicht. Vorhanden ist bereits die Klasse *Produkt* mit dem Konstruktor

Produkt(Produkt p),

der ein weiteres Produkt der Art *p* erzeugt, und die Methode

boolean equals(Produkt p),

die *true* liefert, wenn *p* von der selben Art wie *this* ist. Weiters vorhanden ist die Klasse *Lager* mit der Methode

void einlagern(Produkt p),

die das Produkt *p* an das Lager übergibt.

Implementieren Sie die Klasse *AssemblyMaschine* mit den folgende Methoden: einen Konstruktor

*AssemblyMaschine(Produkt zwischenprodukt1, Produkt zwischenprodukt2,
Produkt endprodukt, Lager endlager),*

der eine *AssemblyMaschine* erzeugt, die Produkte von der Art *zwischenprodukt1* und *zwischenprodukt2* zu einem *endprodukt* zusammensetzt und an das *endlager* weitergibt (Sie können davon ausgehen, dass *zwischenprodukt1* und *zwischenprodukt2* zwei verschiedene Produktarten sind),

und eine Methode

void verarbeite(Produkt p),

mit dem ein Produkt *p* an die *AssemblyMaschine* weitergereicht wird. Sind nach dem Erhalt von Produkt *p* in der *AssemblyMaschine* sowohl ein *zwischenprodukt1* wie auch *zwischenprodukt2* vorhanden, soll sofort ein *endprodukt* gefertigt und an das *endlager* weitergereicht werden. Zwischenprodukte, die noch nicht zu einem *endprodukt* verarbeitet werden können, sollen von der Maschine in einem Zwischenproduktvorrat selbst verwaltet werden.

Aufgabe 2

Implementieren Sie die Methode

```
int findeKleinstenAbstand(int[] a),
```

die den kleinsten Abstand (d.h. die kleinste absolute Differenz) zwischen zwei Zahlen in *a* zurückliefert.

Aufgabe 3

Gegeben ist folgendes Java Interface:

```
interface Formula
{
    int getLength();
    boolean evaluate(boolean [] v);
}
```

Dabei gibt *getLength()* die Länge des Arrays an, die an *evaluate()* übergeben werden kann.

Schreiben Sie eine Methode

```
boolean[] findSolution(Formula f),
```

die ein *boolean*-Array *ba* der Länge *f.getLength()* zurückliefert, sodass *f.evaluate(ba)* den Wert *true* ergibt. Existiert ein solches Array *ba* nicht, soll *null* zurückgeliefert werden.

Hinweis: Es müssen alle *boolean*-Arrays ausprobiert werden, bis ein passendes gefunden ist.