

Aufgabe 1. Templatisieren Sie die Klassen `QuadPol` aus Aufgabe 3 des vorigen Zettels, sodass für die Koeffizienten der Polynome verschiedene Typen genutzt werden können.

Aufgabe 2. Die folgende Klassendeklaration stammt aus der Vorlesung (sie kann mit der Klassendefinition von der Veranstaltungsseite heruntergeladen werden):

```
1 enum Dimension
2 {
3     x_dim, // 0
4     y_dim  // 1
5 };
6
7 class Rechteck
8 {
9 public:
10     Rechteck(
11         double x_min,
12         double x_max,
13         double y_min,
14         double y_max);
15
16     Rechteck(const Rechteck & other);
17     void print() const;
18     double get_min(Dimension dim) const;
19     void set_min(double new_min, Dimension dim);
20     void compute_flaecheninhalt();
21
22 private:
23     double _min[2];
24     double _max[2];
25     double _flaecheninhalt;
26 };
```

Vervollständigen Sie die Klasse, sodass man auf alle Koordinaten schreibend und lesend zugreifen kann.

Implementieren Sie auch mit `try ... catch` und `throw` einen Mechanismus, der dafür sorgt, dass das Programm sofort aussteigt, wenn Rechtecke entstehen, für welche die `min`-Koordinaten nicht in jeder Dimension höchstens so groß wie die `max`-Koordinaten sind.

Aufgabe 3. Schreiben Sie eine Klasse, die Punkte im n -dimensionalen Raum abspeichert, wobei die Dimension eine Variable sei, die dem Konstruktor übergeben werden soll. Außerdem soll in der Klasse die Länge des Vektors in der L_1 -Norm gespeichert werden. Legen Sie dazu selbst keinen Speicher mit `new[]` an. Es soll lesende und schreibende Zugriffsfunktionen für alle Koordinaten geben und lesenden Zugriff auf die L_1 -Länge. Die Koordinaten und die L_1 -Länge selbst sollen `private` sein. Die Klasse soll außerdem templatisiert sein, sodass man für die Koordinaten verschiedene Datentypen wählen kann.

Aufgabe 4. Schreiben Sie eine Klasse, um komplexe Zahlen zu verwalten. Realteil und Imaginärteil der Zahlen können Sie jeweils als `double` speichern. Insbesondere soll Ihre Klasse Operatoren für die vier Grundrechenarten Addition, Subtraktion, Multiplikation und Division enthalten. Die Division durch 0 sollte abgefangen werden.
