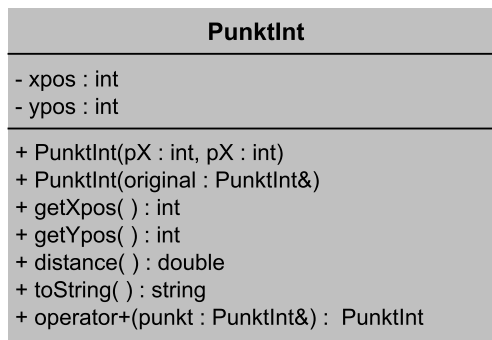


Betrachten Sie das folgende UML-Klassendiagramm zur Realisierung von Punkten im zweidimensionalen Raum mit ganzzahligen Koordinaten:



1. Erstellen Sie zu obigem Klassendiagramm die Klassendefinition von `PunktInt` in einer separaten Schnittstellendatei `PunktInt.h`. Achten Sie darauf, dass ein mehrfaches Einbinden von `PunktInt.h` verhindert wird.
2. Implementieren Sie den ersten Konstruktor der Klasse `PunktInt`. Dieser nimmt zwei `int`-Zahlen entgegen, die die Koordinaten des Punktes festlegen.
3. Implementieren Sie den Kopierkonstruktor. Verwenden Sie dafür den zuerst implementierten Konstruktor.
4. Implementieren Sie die beiden Getter-Methoden `getXpos` und `getYpos`.
5. Implementieren Sie die Methode `distance`, die die Entfernung des Punktes mit den Koordinaten $(xpos, ypos)$ zum Ursprung $(0, 0)$ berechnet.

Hinweis: Die Wurzel einer Zahl kann mit der Methode `sqrt` aus der Bibliothek `cmath` berechnet werden.

6. Implementieren Sie die Methode `toString`, die für einen Punkt (x, y) die Stringdarstellung (x, y) zurückgibt.

Hinweis: Die Methode `std::to_string` wandelt Zahlen in Strings um.

7. Überladen Sie den Operator `+`. Die Summe zweier Punkte (a, b) und (x, y) ist dabei ein neuer Punkt mit den Koordinaten $(a + x, b + y)$.
8. Ergänzen Sie jede Methode um das Schlüsselwort `const`, wo dies sinnvoll erscheint.
9. Angenommen, Sie möchten nun eine Klasse `PunktDouble` erstellen, die dieselbe Funktionalität wie `PunktInt` bereitstellt, aber Punkte mit Koordinaten vom Typ `double` realisiert. An welchen Stellen unterscheidet sich diese Klasse von der Klasse `PunktInt`? Wie sieht es aus, wenn Sie eine Klasse `PunktFloat` realisieren?