Übungsblatt 1

Maximilian Peresunchak 3232875
Nico Reng 3731402
Viorel Tsigos 3720183

Aufgabe 1:

Siehe Code

Aufgabe 2:

2.1.1 Wir haben uns hier für den Container Ärrayëntschieden, da wir hier eine feste Größe bestimmen können, was in der Aufgabenstellung mit 3 Punkten vorgegeben ist. Wir haben hier auch keinen Laufzeit-Overhead also müssen wir nicth unsere Kapazität prüfen wie bei einem Vektor Container. Hierdurch haben wir mehr Leistung.

2.1.2 Siehe Code

2.2.1 Hier haben wir "Listäls Container gewählt. Erstmals ist mit einer doppelt verketteten Liste das Einfügen und Löschen in $\mathcal{O}(1)$ möglich. Der Hauptgrund war aber, dass wenn wir einen Punkt in der List löschen, verlieren wir nicht die Referenz auf die anderen anliegenden Punkte. Bei einem Vektor z.B. würden wir die Referenz zu den anderen Punkten verlieren.

2.2.2 Siehe Code

2.3.1 Hier haben wir jetzt den Container "Vector"verwendet. Wir haben hier auch wieder einen niedrigeren Overhead. Brauchen also keinen zusätzlichen Speicherplatz für einen Pointer. Wir speichern nur die Daten. Außerdem wenn wir über die Liste der Linienzüge iterieren, kann der CPU die Datenblöcke schneller abrufen und wir haben schnellere Verabeitungszeiten.

2.3.2 Siehe Code

2.3.3 Siehe Code

Aufgabe 3:

1. Siehe Code
2. Das hat mit den Referenzen zu tun. Wir befüllen ja die Matrix mit Daten aus matrix1.txt und matrix2.txt. Wir übergeben das hier als Nicht-konstante Referenz. Heißt also dass wir das Objekt MatrixInt verändern können. Dadurch entstehen keine unnötigen Kopien des Matrix-Objekts. Das würde z.B. bei ganz großen Matrizen zu hohem Speicherverbrauch führen und ggf. auch zu Performanceproblemen (Leistung fällt ab). Also Vermeidung von Kopien und durch die const ist auch sichergestellt, dass die Funktion die übergebene Matrix nicht aus Versehen verändern kann. Ein Vorteil von RAII ist die automatische Speicherverwaltung, also die automatische Freigabe einer Ressource ⇒ weniger Ressourcenlecks.

Aufgabe 4:

1. Siehe Code
2. In einer .hpp dürfen nur Deklarationen stehen, keine Definitionen. Deswegen muss man eine neue Definition.hpp Datei erstellen und die Definitionen dort reinpacken. C++ erlaubt nämlich keine Mehrfachdefinitionen von Funktionen. Deshalb könnte es hier passieren, dass hier mehrere Kopien der Funktion in verschiedenen .cpp Dateien entstehen könnte.

Aufgabe 5:

Siehe Code