Aufgabe 1

1 a)

std :: unique_ptr ist ein Zeiger auf ein Objekt, wessen existenz allein auf diesen Zeiger beruht. Verliert dieser Zeiger den Verweis auf das Objekt, so wird das Objekt zerstört.

Wohingegen ein std :: shared_ptr mehrere Zeiger auf dasselbe Objekt erlaubt, sobald der letzte dieser Zeiger diesen Verweis verliert wird das Objekt zerstört.

std :: weak_ptr verhalten sich wie std :: shared_ptr , welche die Zerstörung des Objektes nicht verhindern, sobald also nur noch std :: weak_ptr auf das Objekt verweisen wird das Objekt gelöscht.

std :: auto_ptr ist eine veraltete Version des std :: unique_ptr . Aufgrund der Implementation dieser sind sie ausserdem nicht für die Verwendung in Containern wie Vektoren geeignet.

std :: move wird verwendet, um den std :: unique_ptr eines Objektes zu tauschen. Der alte Zeiger ist danach leer und der neue ist von dort an für die Speicherverwaltung dieses Objektes zuständig. Damit ist move für std :: shared_ptr nutzlos.

std :: make_unique und std :: make_shared erstellen sowohl die jeweiligen Objekte, als auch die Zeiger darauf. Es gibt keine Funktion std :: make_weak, da ein so erstelltes Objekt direkt nach der Erstellung nur einen einzelnen std :: weak_ptr hat und damit sofort Zerstört werden sollte.

1 c)

Der Verweis von Kunde zurück auf das Konto darf kein std :: shared_ptr sein, da dann immer mindestens ein std :: shared_ptr auf jedes Konto zeigt und die automatische Zerstörung von Konten durch std :: shared_ptr niemals in Kraft treten kann.