

Klaviertrio Nr. 3 (2022)

Ergon 90, Musikwerknummer 2168 (2022)

Kompositionsprotokoll

Durchgesehen am 5.8.2022

Inhalt

• Ausdruck	2
• Format	2
• Klang / Aufnahme	2
• Dynamik	2
• Daten / Tagesprotokoll	2
• Zusammenklänge / Akkordstrukturen (16.07.2022)	5
• Unisoni (29.07.2022)	7
• Arbeiten	8
• Die Programmcodes	9
• Programmcode 1 für Teil 1 (Version 4a)	9
• Programmcode 2 für Teil 2 (Version 4c)	11
• Programmcode 3 für Teil 3 (Version 5a)	13
• Programmcode 4 für Teil 6 (Version 3a)	14
• Programmcode 5 für Teil 7 (Version 3b)	16
• Programmcode 6 für Teil 8 (Version 4d)	18
• Programmcode 7 für Teil 9 (Version 5b)	20
• Programmcode 8 für Teil 10 (Version 4e)	21
• Programmcode 9 für Teil 11 (Version 3c)	23
• Programmcode 10 für Teil 12 (Version 3d)	25
• Vorgehen	26
• Brainstorming / Planung / Texte	26
• Form	27
• Kontrollen	27
• Konzept / Werkkommentar	31
• Vorwort	32
• Text	32
• Technik	32
• Fassungen	32
• Kritik / Fragen	32
• Titel	32

Ausdruck

Alle Ebenen.
Gefahrenpunkte:

Format

Ansicht 100%.

Klang / Aufnahme

Computersimulation

Dynamik

Anschlagsstärken Finale: pppp = 10, ppp = 23, pp = 36, p = 49, mp = 62, mf = 75, f = 88, ff = 101, fff = 114, ffff = 127 (Ambitus 0 - 127 vgl. 1/94). Mittelwert: 64.

Veränderungsmöglichkeiten: pppp = **34**, ppp = **43**, pp = **52**, p = **61**, mp = **70**, mf = **79**, f = 88, ff = 101, fff = 114, ffff = 127 (Ambitus 0 - 127 vgl. 1/94). Mittelwert: 64.

Daten / Tagesprotokoll:

- 23.06.2022, Donnerstag, 24.15-1.40 Uhr: Die Idee notieren: Violine, Cello und Klavier rechte Hand spielen die gleichen Rhythmen, aber verschiedene Töne. Klavier linke Hand spielt einzelne Achtel. Gleichzeitige Viertel-Pausen in Vn-Vc-PfrH.
In Opusmodus legte ich die Datei "Klaviertrio 3.opmo" an, übernahm als 1. Version die Version 7 aus "TrioSopVnKlav2022.opmo" und versuchte in einer 2. Version den Code für ein Klaviertrio einzurichten. Das Kompositionsprotokoll einrichten.
- 25.06.2022, Samstag, 9.35-9.50 Uhr: Unterwegs zum Konzert in Lörrach weitere Möglichkeiten notieren:
Einfache Tonrepetition, d.h. pro Phrase eine Tonhöhe.
Zusammengesetzte Tonrepetitionen, d.h. pro Phrase verschiedene Tonhöhen: 2-3-4er Repetitionen in 5-6-7er Phrasen.
- 27.06.2022, Montag, 18.00-19.00 Uhr: In Opusmodus die Versionen 2b, 3a, 4a und 4b schreiben:
 - Version 2b: Vn, Vc u. Klavier r.H. haben den gleichen Rhythmus, Klavier l.H. spielt einzelne Achtel. Chromatische Tonleitern mit gelegentlichen Tonrepetitionen.
 - Version 3a: Vn, Vc u. Klavier r.H. haben den gleichen Rhythmus, Klavier l.H. spielt einzelne Achtel. Chromatische Tonleitern ohne Tonrepetitionen.
 - Version 4a: Vn, Vc u. Klavier r.H. haben den gleichen Rhythmus, Klavier l.H. spielt einzelne Achtel. Einfache Tonrepetitionen, d.h. pro Phrase eine Tonhöhe.
 - Version 4b: Vn, Vc u. Klavier r.H. haben den gleichen Rhythmus, Klavier l.H. spielt einzelne Achtel. Einfache Tonrepetitionen, d.h. pro Phrase eine Tonhöhe. Ohne Oktaven und ohne Unisioni.

Warum eigentlich ohne Oktaven? Wegen Boulez? Oktaven klingen gut. Darauf zu verzichten wäre ein instrumentationstechnischer und klangsinnlicher Verlust.
- 28.06.2022, Dienstag, 10.05-10.30 Uhr: Das Finale-Dokument einrichten (Hilfswerkzeuge programmieren, Olen einrichten, Halslänge korrigieren: Fin25: Dokument: Dokument-Optionen:

Notenhäse. Normale Halslänge 0.35167 statt 0.29167 Zoll, verkürzte Halslänge 0.27611 statt 0.23611 Zoll (Maßeinheiten: Einstellungen unter „Finale“). Die Dateien ordnen.

11.05-12.40 / 14.15-15.40 Uhr: Die verschiedenen Fassungen im Opusmodus-Code "Klaviertrio 3.opmo" anschreiben. Das Protokoll nachführen. Da mich keine der bisherigen Opusmodus-Fassungen zufriedenstellte, entschied ich mich, die Version 4a als Rohmaterial zu nehmen und sie manuell zu bearbeiten.

Deshalb exportierte ich die OM-Version 4a als **Klaviertrio 3, Version 4a, Pdf aus OM.pdf** und als **Klaviertrio 3, Version 4a, OM.xml** und setzte den Programmcode ins Protokoll. Ich importierte das **Klaviertrio 3, Version 4a, OM.xml** ins Finale als **Klaviertrio 3, Version 4a, OM im Finale.musx** und editierte es grob gemäß den **xml-Import-Kontrollen**.

Am Nachmittag begann ich mit der Bearbeitung, indem ich stückchenweise Takte aus **Klaviertrio 3, Version 4a, OM im Finale.musx** ins Hauptdokument **Klaviertrio Nr. 3-1 (2022).musx** kopierte und Pausen zum Atmen einfügte. So wurden die Takte 1-4 von **Klaviertrio 3, Version 4a, OM im Finale.musx** zu den Takten 1-5 des Hauptdokuments **Klaviertrio Nr. 3-1 (2022).musx**. Die Takte 5-6 von **Klaviertrio 3, Version 4a, OM im Finale.musx** wurden zu den Takten 6-9 des Hauptdokuments **Klaviertrio Nr. 3-1 (2022).musx**. Die Takte 7-8 von **Klaviertrio 3, Version 4a, OM im Finale.musx** wurden zu den Takten 11-13 des Hauptdokuments **Klaviertrio Nr. 3-1 (2022).musx**.

Da die l.H. des Klaviers zuwenig Achtel hatte, mußte ich in Opusmodus die **Version 4a** von **Klaviertrio 3.opmo** anpassen und **Klaviertrio 3, Version 4a, Pdf aus OM.pdf** sowie **Klaviertrio 3, Version 4a, OM.xml** aus Opusmodus exportieren und ersetzen und **Klaviertrio 3, Version 4a, OM im Finale.musx** ergänzen.

- 29.06.2022, Mittwoch, 8.50-9.50 Uhr: Die Takte 9-10 von **Klaviertrio 3, Version 4a, OM im Finale.musx** wurden zu den Takten 15-17 des Hauptdokuments **Klaviertrio Nr. 3-1 (2022).musx**. Die Takte 11-16 von **Klaviertrio 3, Version 4a, OM im Finale.musx** wurden zu den Takten 19-24 des Hauptdokuments **Klaviertrio Nr. 3-1 (2022).musx**.
Ich ging wieder zu Opusmodus, verlängerte die Version 4c, exportierte sie als **Klaviertrio 3, Version 4c, Pdf aus OM.pdf** und als **Klaviertrio 3, Version 4c, OM.xml** und setzte den Programmcode ins Protokoll. Ich importierte das **Klaviertrio 3, Version 4c, OM.xml** ins Finale als **Klaviertrio 3, Version 4c, OM im Finale.musx** und editierte es grob gemäß den **xml-Import-Kontrollen**.
- 16.15-16.25 Uhr: Nach dem Schwimmen Ideen für den 3. und 4. Teil skizzieren, siehe Brainstorming.
- 16.50-17.20 Uhr: Erster Versuch, in Opusmodus unter **Version 5a** die Idee für den 3. Teil zu programmieren.
- 30.06.2022, Donnerstag, 9.30-10.00 Uhr: Den Inhalt von **Klaviertrio 3, Version 4c, OM im Finale.musx** ins Hauptdokument **Klaviertrio Nr. 3-1 (2022).musx** einfügen (Takte 27-46) und editieren.
- 06.07.2022, Mittwoch, 11.50-12.45 / 14.00-14.15 Uhr: In Opusmodus die **Version 5a** verbessern, als **Klaviertrio 3, Version 5a, Pdf aus OM.pdf** und als **Klaviertrio 3, Version 5a, OM.xml** exportieren, das **Klaviertrio 3, Version 5a, OM.xml** als **Klaviertrio 3, Version 5a, OM im Finale.musx** ins Finale importieren und grob editieren. Dann nahm ich davon die 20 ersten Takte und setzte sie als Violin-Cello-Duo in die Takte 47-66 der Hauptpartitur **Klaviertrio Nr. 3-1 (2022).musx**. Von den OM-Takten 21-25 nahm ich nur noch Ausschnitte und setzte sie in die Takte 67-71 der Hauptpartitur.
- 07.07.2022, Donnerstag, 10.40-11.50 Uhr: Ich komponierte von Hand (nach traditioneller Art, ohne Opusmodus) die Takte 72-86.
- 08.07.2022, Freitag, 10.00-10.45 Uhr: Das Bisherige durchhören. Kleine Ergänzungen in den Takten 9-10, 14, 18, 72, 75, 80, 82. Ab Takt 86 weiterkomponieren bis Takt 100.
- 09.07.2022, Samstag, 15.40-16.34 Uhr: Das Bisherige durchhören. Kleinigkeiten ergänzen (z.B. Cello-Töne der Takte 67-70) und editieren (z.B. Vorzeichen sichtbar machen). Ich speicherte das Dokument unter **Klaviertrio Nr. 3-2 (2022).musx** als 2. Fassung ab, änderte den Takt 72, schob nach Takt 76 vier Takte ein, um die Idee von Takt 77 weiter auszuarbeiten. Einen Takt später schob ich

wieder zwei Takte ein, um die dortige Idee mehr auszuarbeiten bzw. in mehreren Schritten besser vorzubereiten.

- 10.07.2022, Sonntag, 11.40-13.00 Uhr: Unter „Form“ die Formteile vom 1.-6. Teil auflisten. In Opusmodus die **Version 3a** verbessern, als **Klaviertrio 3, Version 3a, Pdf aus OM.pdf** und als **Klaviertrio 3, Version 3a, OM.xml** exportieren. Ich setzte den Programmcode ins Protokoll und editierte ihn. Ich setzte auch noch den Programmcode von **Version 5a** ins Protokoll und editierte ihn. Schließlich editierte ich auch noch die Programmcodes von **Version 3a** und **Version 3c**. Dann importierte ich das **Klaviertrio 3, Version 3a, OM.xml** ins Finale als **Klaviertrio 3, Version 3a, OM im Finale.musx**, editierte es grob und kopierte es in die Takte 108-126 des Hauptdokuments **Klaviertrio Nr. 3-2 (2022).musx**.
- 11.07.2022, Montag, 11.25-12.50 Uhr: In Opusmodus programmierte ich eine Variante von **Version 3a**, die nun **Version 3b** heißt. Ich exportiere das Noten-Resultat des Opusmodus-Codes als **Klaviertrio 3, Version 3b, Pdf aus OM.pdf** und als **Klaviertrio 3, Version 3b, OM.xml**, setzte den Programmcode ins Protokoll und editierte ihn, importierte das **Klaviertrio 3, Version 3b, OM.xml** als **Klaviertrio 3, Version 3b, OM im Finale.musx** ins Finale, editierte es grob (die Wiedergabe des Tempos einstellen, die Wiedergabe der dynamischen Angaben einstellen, die Wiedergabe der Instrumente einstellen, die Marginalien editieren und einstellen, Titel und Seitenbeschriftung setzen, Halslänge einstellen) und kopierte es ins Hauptdokument **Klaviertrio Nr. 3-2 (2022).musx**, wo es als 7. Teil zu den Takten 127-145 wurde. Die „Form“ ergänzen. Im Hauptdokument an den Teilen 6 und 7 editieren (Takte 108-115).
- 12.07.2022, Dienstag, 11.05-12.00 Uhr: Die Takte 116-145 editieren. Durchhören. In den Takten 92-96 das Klavier einsetzen. Details editieren und gestalten (u.a. Dynamik der Takte 103-106).
- 13.07.2022, Mittwoch, 11.15-12.55 Uhr: Ich programmierte für den 8. Teil in Opusmodus die **Version 4d**. Es ist eine veränderte Wiederaufnahme der Tonrepetitionen des Anfangs, die nun die Mitte des Stückes markieren, gefolgt von einem längeren Nachspiel der linken Hand des Klaviers. In der Partitur die Teile anschreiben.
Ich exportiere die Noten des Opusmodus-Codes als **Klaviertrio 3, Version 4d, Pdf aus OM.pdf** und als **Klaviertrio 3, Version 4d, OM.xml**, setzte den Programmcode ins Protokoll und editierte ihn, importierte das **Klaviertrio 3, Version 4d, OM.xml** als **Klaviertrio 3, Version 4d, OM im Finale.musx** ins Finale, editierte es grob (die Wiedergabe des Tempos einstellen, die Wiedergabe der dynamischen Angaben einstellen, die Wiedergabe der Instrumente einstellen, die Marginalien editieren und einstellen, Titel und Seitenbeschriftung setzen, Halslänge einstellen) und kopierte es ins Hauptdokument **Klaviertrio Nr. 3-2 (2022).musx**, wo es als 8. Teil zu den Takten 147-187 wurde. Die „Form“ ergänzen.
- 14.07.2022, Donnerstag, 10.50-12.50 Uhr: Das Bisherige durchhören.
Ich programmierte für den 8. Teil in Opusmodus die **Version 5b**. Es ist eine Variante des 3. Teils, aber mit Stimmtausch: Ein Duo Violine-Violoncello ohne Klavier. Das Cello spielt chromatische Tonleitern in Gruppen von 4, 5, 6, oder 13 Tönen. Die Violine spielt (wie vorher in der l.H. des Klaviers) einzelne Achtel als Pizzicati. Keine Tonrepetitionen.
Ich exportiere die Noten des Opusmodus-Codes als **Klaviertrio 3, Version 5b, Pdf aus OM.pdf** und als **Klaviertrio 3, Version 5b, OM.xml**, setzte den Programmcode ins Protokoll und editierte ihn, importierte das **Klaviertrio 3, Version 5b, OM.xml** als **Klaviertrio 3, Version 5b, OM im Finale.musx** ins Finale, editierte es grob (die Wiedergabe des Tempos einstellen, die Wiedergabe der dynamischen Angaben einstellen, die Wiedergabe der Instrumente einstellen, die Marginalien editieren und einstellen, Titel und Seitenbeschriftung setzen, Halslänge einstellen) und kopierte es ins Hauptdokument **Klaviertrio Nr. 3-2 (2022).musx**, wo es als 9. Teil zu den Takten 188-228 wurde. Die „Form“ ergänzen.
Das Cello-Nachspiel der Takte 208-227 muß noch bearbeitet werden, um nicht zu gleichförmig zu bleiben. Die einfachste Möglichkeit ist das Löschen von Tönen. Die ausgeführten Änderungen im Hauptdokument **Klaviertrio Nr. 3-2 (2022).musx** können mit **Klaviertrio 3, Version 5b, OM im Finale.musx** verglichen werden.

Ich speicherte das Hauptdokument unter neuem Namen als **Klaviertrio Nr. 3-3 (2022).musx** ab und schob nach Takt 26 einen weiteren Wartetakt ein. Dadurch verschieben sich alle nachfolgenden Taktzahlen um eine Nummer. Das weitere Vorgehen planen.

14.20-14.50 Uhr: Editieren, u.a. im 9. Teil Legatobögen setzen und die Balkenrichtung gemäß der melodischen Richtung ausrichten, leere Systeme ausblenden.

16.30-17.20 Uhr: Im Protokoll die Taktzahlen in „Form“ und in den Programmcodes korrigieren. Im ganzen Stück in der linken Hand des Klaviers Legatobögen setzen.

21.50-23.00 Uhr: Kontrollen: In den Takten 1-147 Vorzeichenkontrolle, Balkenrichtung und Rhythmen zusammenfassen gemacht.

23.00-00.10 Uhr: Ich programmierte für den 10. Teil in Opusmodus die **Version 4e**. Wie Version 4a, aber keine Tonrepetitionen, sondern Blöcke mit längeren Pausen zwischen jedem Block. Nur einzelne Punkte. (Nichttonale) Dreiklänge im Klavier, Zweiklänge in Violine und Cello (Tritoni).

Ich exportiere die Noten des Opusmodus-Codes als **Klaviertrio 3, Version 4e, Pdf aus OM.pdf** und als **Klaviertrio 3, Version 4e, OM.xml**, setzte den Programmcode ins Protokoll und editierte ihn, importierte das **Klaviertrio 3, Version 4e, OM.xml** als **Klaviertrio 3, Version 4e, OM im Finale.musx** ins Finale, editierte es grob (die Wiedergabe des Tempos einstellen, die Wiedergabe der dynamischen Angaben einstellen, die Wiedergabe der Instrumente einstellen, die Marginalien editieren und einstellen, Titel und Seitenbeschriftung setzen, Halslänge einstellen), erstellte noch eine Version mit Tritoni in Violine, Violoncello und rechter Hand des Klaviers und kopierte es ins Hauptdokument **Klaviertrio Nr. 3-2 (2022).musx**, wo es als 10. Teil zu den Takten 230-250 wurde. Die „Form“ ergänzen.

00.15-0.25 Uhr: Editieren.

- 15.07.2022, Freitag, 10.05-12.05 Uhr: Das Ganze durchhören. Viel editieren. Ab Takt 224 das Cello-Nachspiel ins tiefste Register hinabsinken lassen, das heißt: des Umfang ändern.

Ich programmierte für den 11. Teil in Opusmodus die **Version 3c**. Wie Version 3a bzw. wie 3b, aber noch kürzere Phrasen und noch längere Pausen.

Ich exportiere die Noten des Opusmodus-Codes als **Klaviertrio 3, Version 3c, Pdf aus OM.pdf** und als **Klaviertrio 3, Version 3c, OM.xml**, setzte den Programmcode ins Protokoll und editierte ihn, importierte das **Klaviertrio 3, Version 3c, OM.xml** als **Klaviertrio 3, Version 3c, OM im Finale.musx** ins Finale, editierte es grob (die Wiedergabe des Tempos einstellen, die Wiedergabe der dynamischen Angaben einstellen, die Wiedergabe der Instrumente einstellen, die Marginalien editieren und einstellen, Titel und Seitenbeschriftung setzen, Halslänge einstellen) und kopierte es ins Hauptdokument **Klaviertrio Nr. 3-2 (2022).musx**, wo es als 11. Teil zu den Takten 251-279 wurde. Die „Form“ ergänzen. Den Teil editieren.

- 16.07.2022, Samstag, 11.10-12.55 / 13.30-15.30 / 23.00-1.20 Uhr: Ich speicherte das Dokument unter **Klaviertrio Nr. 3-4 (2022).musx** ab und bearbeitete die **Zusammenklänge / Akkordstrukturen** in den Teilen 1-11 bzw. in den Takten 1-279, indem ich die Partitur bis hierhin ausdrückte und langsam am Flügel durchspielte. Die Änderungen schrieb mit Bleistift in die ausgedruckten Noten (siehe „Kompositionsskizzen“). Man kann die Änderungen im Vergleich von **Klaviertrio Nr. 3-3 (2022).musx** zu **Klaviertrio Nr. 3-4 (2022).musx** sehen: Welche Zusammenklänge dürfen vorkommen? Oktaven, Quinten, Sekunden, Septimen. Keine Dur- und Moll-Dreiklänge, keine Dominantseptakkorde (nur in Ausnahmebereichen, siehe unten). Dur- und Molldreiklänge werden in Sekunde+Quarte umgewandelt. Dominantseptakkorde werden in Sekund+Sexte umgewandelt. Die bearbeiteten Takte verzeichnen, die Bearbeitung begründen und die Bearbeitung in die Finale-Partitur **Klaviertrio Nr. 3-4 (2022).musx** übertragen.

Folgende Takte wurden bearbeitet:

- Takt 7, 3. Akkord: C-Dominantseptakkord -> d-e-b
- Takt 8, 2. Akkord: Ges-Dur-Akkord -> c-as-b (Ganztonakkord)
- Takt 11, 1. Akkord: Ges-Dur-Akkord -> c-as-b (Ganztonakkord)
- Takt 12, letzter Akkord: G-Dur-Akkord -> a-h (Ganztonakkord)
- Takt 13, erster und letzter Akkord: G-Dur-Akkord -> a-h (Ganztonakkord)

- Takt 19, 1. Akkord: A-Dominantseptakkord -> a-h-g (Ganztonakkord)
- Takt 19, letzter Akkord: B-Dominantseptakkord -> b-c-gis (Ganztonakkord)
- Takt 20, 1. Akkord: A-Dur-Akkord -> a-h (Ganztonakkord)
- Takt 20, letzter Akkord: B-Dur-Akkord -> c-b (Ganztonakkord)
- Takt 21, 1. Akkord: A-Dur-Akkord -> a-h (Ganztonakkord)
- Takt 21, letzter Akkord: B-Dominantseptakkord -> c-b-gis (Ganztonakkord)
- Takt 22, letzter Akkord bis Takt 35, 1. Akkord: Ausnahmebereich, in dem ich die tonalen Akkorde lasse. Sie werden jeweils nur durch zwei Töne angedeutet und sind somit mehrdeutig (G-Dur oder e-Moll, Ges-Dur oder es-Moll, G-Dur oder e-Moll, As-Dur oder f-Moll, Ganztonakkord, As-Dur oder f-Moll).
- Takt 38 bis Takt 39, 1. Akkord: Zweiter Ausnahmebereich. Es sind auch hier nur große Terzen bzw. kleine Sexten.
- Takt 39, 2. Akkord: B-Dur-Akkord -> b-c (Ganztonakkord)
- Takt 40, 1. Akkord: B-Dur-Akkord -> b-c (Ganztonakkord)
- Takt 54: Die Oktave cis-cis2 wirkte plump, deshalb setzte ich ein c2 davor.
- Takt 58: Die Oktavparallele es-e wirkte plump, deshalb korrigierte ich das Cello.
- Takt 62: Die Oktave es-es2 wirkte plump, deshalb setzte ich ein d2 davor.
- (In Takt 63 störte mich die Oktave c-c3 nicht, da sie durch das vorangehende cis im Baß abgeschwächt wird.)
- Takt 65: Die Oktave cis-cis2 wirkte plump, deshalb setzte ich ein c2 davor.
- In den Takten 66 und 67 strich ich die Töne g und c in der Geige, um die Phrasen offener zu halten bzw. um dem g und c nicht zuviel Gewicht zu geben.
- (In den Teilen 4 und 5 mußte ich nichts korrigieren, da diese Teile nicht vom Computerprogramm generiert wurden, sondern von Hand komponiert wurden und dadurch bereits den Kriterien entsprachen.)
- Takt 115, die Töne 5-7 in der rechten Hand: Eine zu deutliche Wendung nach F-Dur. Korrektur: -> Die Parallelbewegung weiterführen.
- Takt 142, 2. Akkord: Als Ausnahme lasse ich den unvollständigen G-Dur-Akkord stehen, um durch das Spielen mit den Tonalität etwas Kontur, das heißt: klanglich Abwechselung hineinzubringen.
- Takt 157, 2. Akkord: Durch das abschließende fis in Takt 158 im Baß bekommt der Ges-Dur-Akkord zuviel Gewicht -> as-b (Ganztonakkord)
- (Den unvollständigen G-Durakkord in Takt 160 lasse ich, da er durch das vorangehende gis im Baß abgeschwächt wird.)
- (In Takt 161 lasse ich den 2. Akkord, um die unübliche Auflösung in den f-Moll-Akkord zu erhalten.)
- Im 9. Teil gibt es viele Änderungen im Cello, die darauf zurückzuführen sind, das die umfangbildenden Töne, nachdem sie gleich am Anfang deutlich in Erscheinung treten, danach nicht zu penetrant wiederholt werden sollen.
- Takt 190: Die leere Quinte fis-cis klingt in diesem Kontext fremd.
- Takt 191: Das c ist zu penetrant.
- Takt 198: Mit der Phrasen-Erweiterung bis zum cis1 wird der Umfang erweitert bzw. in Frage gestellt.
- Takt 199: Die Oktavparallele es-e wirkte plump, deshalb korrigierte ich sie in der Geige. Und durch das c1 im Cello gibt es wieder eine Infragestellung des Umfangs, um der Gleichförmigkeit entgegenzuwirken.
- Takt 203: Durch das H entsteht nun eine Umfangerweiterung nach unten.
- Takt 204: Durch H und B wird die Umfangerweiterung nach unten weitergeführt.
- Dito in Takt 205 und 207.
- Die Töne im Taktübergang 211-212 werden gestrichen, um die Umfangmarkierung abzuschwächen und die Fragmentierung zu verstärken. Usw.

- Takt 255: Die Auflösung des übermäßigen Sextakkords in den D-Dominantseptakkord wirkt in diesem Zusammenhang stilfremd.
- Takt 256: Der letzte Akkord würde eine isolierte Struktur aufweisen. (Harte statt weiche Dissonanzen.)
- Taktübergang 260-261: Fremde Akkordstruktur und zu penetrantes g.
- Takt 264: Die zu traditionelle Trugschlußauflösung wird in eine Ganztonverbindung geändert.
- Takt 270: Die Auflösung nach E-Dur wird in eine Ganztonverbindung geändert.

Die geänderten Takte verzeichnen und begründen.

Die geänderten Stellen in die Finale-Partitur **Klaviertrio Nr. 3-4 (2022).musx** übertragen.

Am Werkkommentar arbeiten.

- 19.07.2022, Dienstag, 15.45-17.00 Uhr: Ich programmierte für den 12. Teil in Opusmodus die **Version 3d**. Wie Version 3a, aber Mischung von Tonwiederholungen und Läufen. Viele Gegenbewegungen.
Ich exportiere den Opusmodus-Code als **Klaviertrio 3, Version 3d, Pdf aus OM.pdf** und als **Klaviertrio 3, Version 3d, OM.xml**, setzte den Programmcode ins Protokoll und editierte ihn, importierte das **Klaviertrio 3, Version 3d, OM.xml** als **Klaviertrio 3, Version 3d, OM im Finale.musx** ins Finale, editierte es grob (die Wiedergabe des Tempos einstellen, die Wiedergabe der dynamischen Angaben einstellen, die Wiedergabe der Instrumente einstellen, die Marginalien editieren und einstellen, Titel und Seitenbeschriftung setzen, Halslänge einstellen) und kopierte es ins Hauptdokument, wo es als 12. Teil zu den Takten 280-298 wurde. Die „Form“ ergänzen. Den 12. Teil editieren. Dann hängte ich noch die Coda der Takte 299-307. Damit ist nun der **Endpunkt des Stückes** erreicht.
- 29.07.2022, Freitag, 12.10-12.35 14.35-15.05 / 16.55-18.35 Uhr: Kontrollen ausführen (Pausen, Vorzeichen). Dabei tauchte eine andere Frage auf: Dürfen **Unisoni** vorkommen? (Takte 270-271 Vn-Vc, Takte 281, 283, 287-291 Vn-Vc.) Am Schluß des Stückes, sozusagen als Erschöpfung des Materials und als Vorbereitung auf die Schlußoktaven (ab Takt 304) macht es Sinn.

Wenn drei Stimmen nur in kleinen Sekunden fortschreiten, gibt es nur die Möglichkeiten der Gegenbewegung oder der gleichgerichteten Bewegung. Bei der gleichgerichteten Bewegung gibt es nur die Parallelbewegung. Ob die Parallelbewegung nun in großen Sekunden oder in Unisoni fortschreitet, ist nur ein gradueller (und kein prinzipieller) Unterschied.

Die anderen Stellen im Stück (mit chromatischen Läufen in allen Instrumenten) auf **Unisoni** kontrollieren.

4. Teil: Nur Gegenbewegungen, da von Hand komponiert.

6. Teil: Unisoni in den Takten 110, 113-116 Vn-Vc, 119-121 Vn-Klavier. Es kommen in diesem Teil viele Parallelbewegungen vor, meistens in großen Sekunden oder kleinen Septimen, aber auch in großen Terzen, Tritoni und kleinen Sexten, d.h. in Intervallstrukturen aus der Ganztonskala. Warum nicht auch ab und zu als Unisoni?

7. Teil: Unisoni in den Takten 135-138 Vc-Klavier (z.T. oktaviert), 141-143 Vn-Klavier. Gleiche Situation wie im 6. Teil.

11. Teil: Unisoni in den Takten 251-252 Vc-Klavier, 268-269, 270-271 Vn-Vc.

12. Teil: Unisoni in den Takten 280, 281, 283, 284, 287-291 Vn-Vc. In diesem Teil werden die Unisoni, als Kulmination der oben beschriebenen Parallelbewegungen, geradezu zu einem konstituierenden Bestandteil der Satzstruktur.

Vergleiche zu dieser Thematik auch die am 16.7.2022 vorgenommenen Kontrollen der Akkordstrukturen.

- 30.07.2022, Samstag, 11.30-12.45 Uhr: Das Protokoll nachführen. Das Ganze durchhören, ab Takt 306 noch 4 Takte einschieben und im Schlußklang Cello und Klavier um 2 Oktaven nach unten setzen. Die Balken-Richtung (steigende Melodie = steigende Balkenrichtung und umgekehrt für die

fallende Bewegung) ab Takt 148 bis zum Schluß anpassen. Im Cello überall den richtigen Schlüssel festlegen. Die Musikwerknummer zuordnen und das Werk in den Werklisten verzeichnen.

15.45-16.15 Uhr: Stimmen ziehen und editieren.

- 31.07.2022, Sonntag, 11.55-13.00 / 13.15-13.30 / 14.00-14.30 / 16.35-18.05 Uhr: Die Einzelstimmen editieren. Ich schob im Schlußklang nochmals 4 Takte ein und oktavierte die Violine nach oben. Systemtrennstriche in die Partitur setzen.
- 01.08.2022, Montag, 11.05-12.45 / 14.30-14.45 Uhr: Das Werk in der Werkliste auf der Homepage und in der Werkliste-Word in den verschiedenen Kategorien verzeichnen. Das Ganze durchhören. Ich transponierte die Töne der Takte 299-301 einen Halbton nach oben, damit in Takt 302 das Absinken um einen Halbton als Entspannung wirken kann. Die Partitur editieren und ausdrucken. Das Einzelstimmen-Layout editieren.
- 02.08.2022, Dienstag, 2.10-2.20 / 12.45-12.55 Uhr: Am Werkkommentar arbeiten.
15.25-16.10 Uhr: Die Einzelstimmen mit der Partitur vergleichen.
23.15-23.45 Uhr: Partitur und Einzelstimmen editieren und ausdrucken.
- 03.08.2022, Mittwoch, 11.05-12.30 Uhr: Den Werkkommentar durchlesen, in ein eigenes Word-Dokument setzen, auf die Homepage setzen und dort mit dem Inhaltsverzeichnis (der Werkkommentare) und mit der Werkliste verlinken. Titelblatt, Innenblatt, Vorwort, Biographie herstellen. Titelblatt, Innenblatt, Vorwort, Werkkommentar und Bio für die Kopiervorlage ausdrucken. Gesamt-Pdf-Fassung und Druckfassung herstellen
14.40-14.50 Uhr: Editieren.
- 04.08.2022, Donnerstag, 11.10-12.30 Uhr: Partiturschublade erstellen. Das Werk bei der GEMA Anmelden, bei der SME anmelden und an verschiedenen Orten verzeichnen. Die Fassungen verzeichnen. Partitur editieren.
14.10-14.50 Uhr: Den Entwicklungsgang zusammenstellen und editieren.
17.10-18.55 Uhr: Die Kompositionsskizzen einscannen. Auf der Homepage für dieses Stück den Artikel "Downloads" einrichten.
Die Gesamtpartitur-Fassung, die Druckfassung, die Einzelstimmen, den Entwicklungsgang und die Kompositionsskizzen auf die Homepage („Downloads“) setzen, und mit dem Inhaltsverzeichnis und der Werkliste verlinken. Das Protokoll bis am 28.6.2022 durchlesen und ergänzen. Das stellte ich fest, daß die Opusmodus-Dateien noch nicht im Entwicklungsgang sind.
00.10-1.25 Uhr: Die Opusmodus-Dateienbeschriftungen ergänzen: „OM.musx“ wird zu „OM im Finale.musx“ (z.B. „Klaviertrio 3, Version 3a, OM.musx“ wird zu -> „Klaviertrio 3, Version 3a, OM im Finale.musx“) und „OM.pdf“ wird zu „Pdf aus OM.pdf“ (z.B. „Klaviertrio 3, Version 3a, OM.pdf“ wird zu -> „Klaviertrio 3, Version 3a, Pdf aus OM.pdf“), damit bei der Umwandlung der Finale-Dateien in Pdfs klar bleibt, welche Dateien als Pdf direkt aus Opusmodus exportiert wurden, und welche Pdfs eine Umwandlung von Finale-Dateien darstellen. Alle Datei-Namen entsprechend ergänzen und in den Dateien alle Titel / Überschriften entsprechend ergänzen bzw. in den Pdfs aus OM die Überschriften überhaupt erst setzen. Dann ergänzte ich auch im Protokoll alle Datei-Namen entsprechend. Nun stellte ich den Entwicklungsgang neu zusammen und setzte ihn auf die Homepage unter „Downloads“.
- 05.08.2022, Freitag, 11.00-12.30 Uhr: Das Protokoll ab dem 28.06.2022 kontrollierend durchlesen. Nochmals die Seitenzahlen im Inhaltsverzeichnis kontrollieren und das Protokoll auf die Homepage setzen.

Arbeiten:

- ✓ Im 6. Teil (Takte 108-126) und 7. Teil (Takte 127-145) die Legatobögen und Balken-Richtungen editieren. Vorzeichen setzen. Gemacht.
- ✓ 9. Teil: Legatobögen, Balkenrichtungen: Gemacht am 14.7.2022
- ✓ Im Protokoll die Taktzahlen in „Form“ und „Programmcodes“ korrigieren: Gemacht am 14.7.2022

- ✓ Im ganzen Stück in der linken Hand des Klaviers Legatobögen setzen: Gemacht am 14.7.2022.
- ✓ Leere Systeme ausblenden: Gemacht am 14.7.2022
- ✓ 9. Teil: Neues Vn-Vc-Duo: Gegenteil vom 3. Teil: Vn pizz. und Cello chrom. Läufe (in tiefer Lage). Gemacht am 14.7.2022
- 10. Teil: Wie Version 4a, aber keine Tonrepetitionen, sondern Blöcke mit längeren Pausen zwischen jedem Block. Nur einzelne Punkte. Dreiklänge im Klavier, Zweiklänge in Violine und Cello (Tritoni). Gemacht am 14.7.2022.
- Vn, Vc u. Klavier r.H. spielen einzeln nacheinander fallende Chromatik, 4-5-6-7-13, Klavier l.H. spielt einzelne Achtel. Einfache Tonrepetitionen, d.h. pro Phrase eine Tonhöhe.
; Im Gegensatz zu 4a: Engere Umfänge für die einzelnen Instrumente. Mehr Achtel in der l.H. Anderer seed.
- 11. Teil: Wie 1. Teil, aber kürzere Phrasen (mit den längeren Pausen).
Letzter Teil: Da nun im 10. Teil auch der Tritonus eingeführt wurde, melodisch nun Kombinationen von chrom. Läufen, Tritoni und Tonrepetitionen. Ev. als einstimmige Melodie beginnend, die über die Zweistimmigkeit zur dreistimmigen linearen Polyphonie wird. Version 3 ohne swallow t im make-omn? (16.7.2022.) Nicht ausgeführt.
-> 3d: Mischung von Tonwiederholungen und Läufen. Viele Gegenbewegungen.
Oder in parallelen Oktaven, dann Tritoni, dann Tritonus-Quart, dann gr. Sept.-kl. None zur führenden Stimme. (19.7.2022)
- Halbtonkontrolle im Baß: unnötig. Es darf Verschiedenes vorkommen.
- ✓ Die Fassungen verzeichnen: Gemacht am 4.8.2022.
- **ArbeitenAktuellerPunkt [ar]**

Programmcodes

1)

; Klaviertrio 3.opmo, Version 4a (27.6.2022): 1. Teil (Takte 1-27).

Vn, Vc u. Klavier r.H. haben den gleichen Rhythmus, Klavier l.H. spielt einzelne Achtel. Einfache Tonrepetitionen, d.h. pro Phrase eine Tonhöhe.

(progn

(init-seed 33)

(setf Rhy-Tutti (rnd-sample 39 '((s s s -s) (s s s s -e) (s s s s s -e.)(-q)))) ; 3-4-5

(setf Rhy-Pf-lH (rnd-sample 39 '((e) (-q))))

(setf count-Tutti (mclength Rhy-Tutti))

(setf count-Pf-rH (mclength Rhy-Pf-lH))

; mclength zählt die Anzahl Ereignisse einer definierten Variablen, z.B. von Rhy-Tutti (4 oder 6 oder 8 Ereignisse in (s s s -e) usw.

; Damit es funktioniert, müssen die Rhy-Variablen aber nicht nur eine Klammer haben, sondern eine Gruppen von Klammern sein, also nicht:

; (setf Rhy-Pf-rH (rnd-sample 50 '(e -e))), sondern

; (setf Rhy-Pf-rH (rnd-sample 50 '((e) (-e)))). Somit kann er jeweils die Anzahl pro Klammer anzeigen.

; mclength wird also auf die folgende Sequenz angewandt.

; Bei (setf Rhy-Sop (rnd-sample 50 '(q q. h h. -e))) müßte man nur "count-Sop (length" schreiben, weil dann nur eine Gesamtanzahl ausgegeben wird.

(setf Int-Vn (gen-repeat count-Tutti (mclist (gen-weight 50 '((0 9)(0 4))))))

(setf Int-Vc (gen-repeat count-Tutti (mclist (gen-weight 50 '((0 9)(0 4))))))

```
(setf Int-Pf-rH (gen-repeat count-Tutti (mclist (gen-weight 50 '((0 9)(0 4))))))
(setf Int-Pf-lH (gen-repeat count-Pf-rH (mclist (gen-weight 50 '((0 9)(0 9))))))
; gen-weight gewichtet die Intervall-Häufigkeit: (-1 9) bedeutet: fallende kleine Sekunde 9x
gewichtet.
; mclist = mapcar list, damit nicht eine lange Liste mit (1 -1 -1 1 usw.) entsteht, sondern damit
jeder ausgegebene Wert in einer eigenen Liste / Klammer steht ((1)(-1)(-1) usw.). Somit können
diese Klammern den Rhythmus-Klammern zugeordnet werden. Z.B. gilt dann (1) für (s s s -e),
und es werden nur steigende kl. Sekunden erklingen (vorerst aber nur 1).
; Vergleiche gen-weight ohne und mit mclist
; Mit "gen-repeat count-Vn" wird die Intervallfortschreitung (z.B. kl2 aufwärts) so oft wiederholt,
wie in der Variablen "count-Vn" in "Rhy-Vn" rhythmische Ereignisse gezählt wurden. Also bei (s
s s -e) 4x, wobei das 4. Mal als Pause nicht erklingt.
```

```
(setf Pitch-Vn (pitch-transpose 24 (interval-to-pitch Int-Vn)))
(setf Pitch-Vc (pitch-transpose 24 (interval-to-pitch Int-Vc)))
(setf Pitch-rH (pitch-transpose 24 (interval-to-pitch Int-Pf-rH)))
(setf Pitch-lH (pitch-transpose 24 (interval-to-pitch Int-Pf-lH)))
; interval-to-pitch verwandelt die Zahlenfolgen von Int-Vn in Intervalle.
; pitch transpose betrifft nur den Ausgangston. Ohne nähere Definition starten die Tonhöhen
immer beim eingestrichenen "c" (c4).
; (setf Pitch-Sop (pitch-transpose 12 (interval-to-pitch Int-Sop)))
; (setf Pitch-Klav (interval-to-pitch Int-Klav))
```

```
(setf Vn-Part (make-omn
:pitch Pitch-Vn
:length Rhy-Tutti
:velocity '(f)
:swallow t
))
(setf Vc-Part (make-omn
:pitch Pitch-Vc
:length Rhy-Tutti
:velocity '(f)
:swallow t
))
(setf Pf-rH-Part (make-omn
:pitch Pitch-rH
:length Rhy-Tutti
:velocity '(f)
:swallow t
))
(setf Pf-lH-Part (make-omn
:pitch Pitch-lH
:length Rhy-Pf-lH
:velocity '(f)
:swallow t
))
```

; Tonhöhen und Dauern werden pro Stimme zusammengefügt.

```
(setf list-int11 (rnd-sample 10 '(1 -1)))
(setf list-int12 (rnd-sample 10 '(1 -1)))
```

```
(setf list-int13 (rnd-sample 10 '(1 -1)))
(setf list-int14 (rnd-sample 10 '(1 -1)))
; Eine Liste von Intervallen, die zwischen den Gruppen zur Anwendung kommen sollen.

(setf violin (ambitus '(g3 c7)(bind-to-interval list-int11 Vn-Part)))
(setf cello (ambitus '(c2 c5)(bind-to-interval list-int12 Vc-Part)))
(setf piano-rH (ambitus '(c4 c6) (bind-to-interval list-int13 Pf-rH-Part)))
(setf piano-lH (ambitus '(fs1 b3) (bind-to-interval list-int14 Pf-lH-Part)))
; bind-to-interval: bind (connect) a series of lists (patterns) to each-other with the given interval
; bzw. with given intervals

(ps 'gm :vn* (list violin) :vc (list cello) :p (list piano-rH piano-lH) :tempo 120 :time-signature '(4 4))
(init-seed nil)
)
; Mit 'gm :vn* erklingt der Midi-Klang für Solo-Violine. Sie Utilities: L: GM Instrument Set.lisp
; Mit (list Vn-Part ... werden die einzelnen Stimmen zu einer Partitur zusammengefügt.

; Evaluieren: cmd-e
; [27"]

; --
```

2)

;; Klaviertrio 3.opmo, Version 4c (27.6.2022): 2. Teil (Takte 28-47).

; Vn, Vc u. Klavier r.H. haben den gleichen Rhythmus, Klavier l.H. spielt einzelne Achtel.
Einfache Tonrepetitionen, d.h. pro Phrase eine Tonhöhe.
; Im Gegensatz zu 4a: Engere Umfänge für die einzelnen Instrumente. Mehr Achtel in der l.H.
Anderer seed.

```
(progn
(init-seed 34)
(setf Rhy-Tutti (rnd-sample 50 '((s s s -s) (s s s s -e) (s s s s s -e.)(-q)))) ; 3-4-5
(setf Rhy-Pf-lH (rnd-sample 127 '((e) (e) (e) (-e) (-q))))
```

```
(setf count-Tutti (mclength Rhy-Tutti))
(setf count-Pf-rH (mclength Rhy-Pf-lH))
; mclength zählt die Anzahl Ereignisse einer definierten Variablen, z.B. von Rhy-Tutti (4 oder 6
; oder 8 Ereignisse in (s s s -e) usw.
; Damit es funktioniert, müssen die Rhy-Variablen aber nicht nur eine Klammer haben, sondern
; eine Gruppen von Klammern sein, also nicht:
; (setf Rhy-Pf-rH (rnd-sample 50 '(e -e))), sondern
; (setf Rhy-Pf-rH (rnd-sample 50 '((e) (-e)))). Somit kann er jeweils die Anzahl pro Klammer
; anzeigen.
; mclength wird also auf die folgende Sequenz angewandt.
; Bei (setf Rhy-Sop (rnd-sample 50 '(q q. h h. -e))) müßte man nur "count-Sop (length" schreiben,
; weil dann nur eine Gesamtanzahl ausgegeben wird.
```

```
(setf Int-Vn (gen-repeat count-Tutti (mclist (gen-weight 50 '((0 9)(0 4)))))
(setf Int-Vc (gen-repeat count-Tutti (mclist (gen-weight 50 '((0 9)(0 4)))))
(setf Int-Pf-rH (gen-repeat count-Tutti (mclist (gen-weight 50 '((0 9)(0 4)))))
```

```
(setf Int-Pf-IH (gen-repeat count-Pf-rH (mclist (gen-weight 50 '((0 9)(0 9))))))
; gen-weight gewichtet die Intervall-Häufigkeit: (-1 9) bedeutet: fallende kleine Sekunde 9x
gewichtet.
; mclist = mapcar list, damit nicht eine lange Liste mit (1 -1 -1 1 usw.) entsteht, sondern damit
jeder ausgegebene Wert in einer eigenen Liste / Klammer steht ((1)(-1)(-1) usw.). Somit können
diese Klammern den Rhythmus-Klammern zugeordnet werden. Z.B. gilt dann (1) für (s s s -e),
und es werden nur steigende kl. Sekunden erklingen (vorerst aber nur 1).
; Vergleiche gen-weight ohne und mit mclist
; Mit "gen-repeat count-Vn" wird die Intervallfortschreitung (z.B. kl2 aufwärts) so oft wiederholt,
wie in der Variablen "count-Vn" in "Rhy-Vn" rhythmische Ereignisse gezählt wurden. Also bei (s
s s -e) 4x, wobei das 4. Mal als Pause nicht erklingt.
```

```
(setf Pitch-Vn (pitch-transpose 24 (interval-to-pitch Int-Vn)))
(setf Pitch-Vc (pitch-transpose 24 (interval-to-pitch Int-Vc)))
(setf Pitch-rH (pitch-transpose 24 (interval-to-pitch Int-Pf-rH)))
(setf Pitch-IH (pitch-transpose 24 (interval-to-pitch Int-Pf-IH)))
; interval-to-pitch verwandelt die Zahlenfolgen von Int-Vn in Intervalle.
; pitch transpose betrifft nur den Ausgangston. Ohne nähere Definition starten die Tonhöhen
immer beim eingestrichenen "c" (c4).
; (setf Pitch-Sop (pitch-transpose 12 (interval-to-pitch Int-Sop)))
; (setf Pitch-Klav (interval-to-pitch Int-Klav))
```

```
(setf Vn-Part (make-omn
:pitch Pitch-Vn
:length Rhy-Tutti
:velocity '(f)
:swallow t
```

```
))
```

```
(setf Vc-Part (make-omn
:pitch Pitch-Vc
:length Rhy-Tutti
:velocity '(f)
:swallow t
```

```
))
```

```
(setf Pf-rH-Part (make-omn
:pitch Pitch-rH
:length Rhy-Tutti
:velocity '(f)
:swallow t
```

```
))
```

```
(setf Pf-IH-Part (make-omn
:pitch Pitch-IH
:length Rhy-Pf-IH
:velocity '(f)
:swallow t
```

```
))
```

; Tonhöhen und Dauern werden pro Stimme zusammengefügt.

```
(setf list-int11 (rnd-sample 10 '(1 -1)))
```

```
(setf list-int12 (rnd-sample 10 '(1 -1)))
```

```
(setf list-int13 (rnd-sample 10 '(1 -1)))
```

```
(setf list-int14 (rnd-sample 10 '(1 -1)))
; Eine Liste von Intervallen, die zwischen den Gruppen zur Anwendung kommen sollen.

(setf violin (ambitus '(c6 b6)(bind-to-interval list-int11 Vn-Part)))
(setf cello (ambitus '(c4 b4)(bind-to-interval list-int12 Vc-Part)))
(setf piano-rH (ambitus '(c5 b5) (bind-to-interval list-int13 Pf-rH-Part)))
(setf piano-lH (ambitus '(c3 b3) (bind-to-interval list-int14 Pf-lH-Part)))
; bind-to-interval: bind (connect) a series of lists (patterns) to each-other with the given interval
; bzw. with given intervals

(ps 'gm :vn* (list violin) :vc (list cello) :p (list piano-rH piano-lH) :tempo 120 :time-signature '(4 4))
(init-seed nil)
)
; Mit 'gm :vn* erklingt der Midi-Klang für Solo-Violine. Sie Utilities: L: GM Instrument Set.lisp
; Mit (list Vn-Part ... werden die einzelnen Stimmen zu einer Partitur zusammengefügt.

; Evaluieren: cmd-e
; [40"]
```

--

3)

;; Klaviertrio 3.opmo, Version 5a (29.6.2022): 3. Teil (Takte 48-72).

; Duo Vn-Vc ohne Klavier (der Einfachheit halber als Klaviersystem zusammengefaßt). Vn spielt zuerst fallende, dann steigende, dann wieder fallende Chromatik, 4-5-6-13, Cello spielt (wie vorher in der l.H. des Klaviers) einzelne Achtel. Keine Tonrepetitionen.

```
(progn
(init-seed 34)
(setf rhy-tutti (rnd-sample 50 '((s s s s -e) (s s s s s -e.) (s s s s s s -q.)(s s s s s s s s s s s s -e. -h))))
(setf rhy-pf-lh (rnd-sample 127 '((e) (e) (e) (-e) (-q))))
```

```
(setf count-tutti (mclength rhy-tutti))
(setf count-pf-lh (mclength rhy-pf-lh))
```

```
(setf int-vn (gen-repeat count-tutti (gen-weight 50 '((-1 1)(-1 1))))
(setf int-pf-lh (gen-repeat count-pf-lh (gen-weight 50 '((1 1)(-1 1))))
```

```
(setf pitch-tutti (pitch-transpose 24 (interval-to-pitch int-vn)))
(setf pitch-lh (pitch-transpose 24 (interval-to-pitch int-pf-lh)))
; interval-to-pitch verwandelt die Zahlenfolgen von Int-Vn in Intervalle.
; pitch transpose betrifft nur den Ausgangston. Ohne nähere Definition starten die Tonhöhen immer beim eingestrichenen "c" (c4).
```

```
(setf tutti-part (make-omn
:pitch pitch-tutti
:length rhy-tutti
:velocity '(pp)
))
```

```
(setf pf-lh-part (make-omn
```

```

:pitch pitch-lh
:length rhy-pf-lh
:velocity '(pp)
))
; Tonhöhen und Dauern werden pro Stimme zusammengefügt.

```

```

(setf tutti-amb (ambitus '(c4 c6) tutti-part))
(setf piano-lh-amb (ambitus '(c2 b3) pf-lh-part))

```

```

(ps 'gm :p (list tutti-amb piano-lh-amb) :tempo 120 :time-signature '(4 4)
(init-seed nil)
)

```

```

; Mit 'gm :vn* erklingt der Midi-Klang für Solo-Violine. Sie Utilities: L: GM Instrument Set.lisp
; Mit (list Vn-Part ... werden die einzelnen Stimmen zu einer Partitur zusammengefügt.

```

```

; Evaluieren: cmd-e
; [40"]

```

--

4)

```

;; Klaviertrio 3.opmo, 3. Version 3a (27.6.2022/10.07.2022): 6. Teil (Takte 109-127).
; Vn, Vc u. Klavier r.H. haben den gleichen Rhythmus, Klavier l.H. spielt einzelne Achtel. Chromatische
Tonleitern ohne Tonrepetitionen.

```

```

(progn
(init-seed 33)
(setf Rhy-Tutti (rnd-sample 39 '((s s s -s) (s s s s -e) (s s s s s -e.)(-q))))
(setf Rhy-Pf-lH (rnd-sample 99 '((e) (-q))))

```

```

(setf count-Tutti (mclength Rhy-Tutti))
(setf count-Pf-lH (mclength Rhy-Pf-lH))
; mclength zählt die Anzahl Ereignisse einer definierten Variablen, z.B. von Rhy-Tutti (4 oder 6
oder 8 Ereignisse in (s s s -e) usw.
; Damit es funktioniert, müssen die Rhy-Variablen aber nicht nur eine Klammer haben, sondern
eine Gruppen von Klammern sein, also nicht:
; (setf Rhy-Pf-rH (rnd-sample 50 '(e -e))), sondern
; (setf Rhy-Pf-rH (rnd-sample 50 '((e) (-e)))). Somit kann er jeweils die Anzahl pro Klammer
anzeigen.
; mclength wird also auf die folgende Sequenz angewandt.
; Bei (setf Rhy-Sop (rnd-sample 50 '(q q. h h. -e))) müßte man nur "count-Sop (length" schreiben,
weil dann nur eine Gesamtanzahl ausgegeben wird.

```

```

(setf Int-Vn (gen-repeat count-Tutti (mclist (gen-weight 50 '((-1 9)(1 4)))))
(setf Int-Vc (gen-repeat count-Tutti (mclist (gen-weight 50 '((-1 9)(1 4)))))
(setf Int-Pf-rH (gen-repeat count-Tutti (mclist (gen-weight 50 '((-1 9)(1 4)))))
(setf Int-Pf-lH (gen-repeat count-Pf-lH (mclist (gen-weight 50 '((-1 9)(1 9)))))

```

; gen-weight gewichtet die Intervall-Häufigkeit: (-1 9) bedeutet: fallende kleine Sekunde 9x gewichtet.
 ; mclist = mapcar list, damit nicht eine lange Liste mit (1 -1 -1 1 usw.) entsteht, sondern damit jeder ausgegebene Wert in einer eigenen Liste / Klammer steht ((1)(-1)(-1) usw.). Somit können diese Klammern den Rhythmus-Klammern zugeordnet werden. Z.B. gilt dann (1) für (s s s -e), und es werden nur steigende kl. Sekunden erklingen (vorerst aber nur 1).
 ; Vergleiche gen-weight ohne und mit mclist
 ; Mit "gen-repeat count-Vn" wird die Intervallfortschreitung (z.B. kl2 aufwärts) so oft wiederholt, wie in der Variablen "count-Vn" in "Rhy-Vn" rhythmische Ereignisse gezählt wurden. Also bei (s s s -e) 4x, wobei das 4. Mal als Pause nicht erklingt.

```
(setf Pitch-Vn (pitch-transpose 24 (interval-to-pitch Int-Vn)))
(setf Pitch-Vc (pitch-transpose 24 (interval-to-pitch Int-Vc)))
(setf Pitch-rH (pitch-transpose 24 (interval-to-pitch Int-Pf-rH)))
(setf Pitch-lH (pitch-transpose 24 (interval-to-pitch Int-Pf-lH)))
  ; interval-to-pitch verwandelt die Zahlenfolgen von Int-Vn in Intervalle.
  ; pitch transpose betrifft nur den Ausgangston. Ohne nähere Definition starten die Tonhöhen immer beim eingestrichenen "c" (c4).
  ; (setf Pitch-Sop (pitch-transpose 12 (interval-to-pitch Int-Sop)))
  ; (setf Pitch-Klav (interval-to-pitch Int-Klav))
```

```
(setf Vn-Part (make-omn
  :pitch Pitch-Vn
  :length Rhy-Tutti
  :velocity '(f)
  :swallow t
```

```
))
```

```
(setf Vc-Part (make-omn
  :pitch Pitch-Vc
  :length Rhy-Tutti
  :velocity '(f)
  :swallow t
```

```
))
```

```
(setf Pf-rH-Part (make-omn
  :pitch Pitch-rH
  :length Rhy-Tutti
  :velocity '(f)
  :swallow t
```

```
))
```

```
(setf Pf-lH-Part (make-omn
  :pitch Pitch-lH
  :length Rhy-Pf-lH
  :velocity '(f)
  :swallow t
```

```
))
```

; Tonhöhen und Dauern werden pro Stimme zusammengefügt.

```
(setf violin (ambitus '(g3 c7) Vn-Part))
(setf cello (ambitus '(c2 c5) Vc-Part))
(setf piano-rH (ambitus '(c4 c6) Pf-rH-Part))
(setf piano-lH (ambitus '(fs1 b3) Pf-lH-Part))
```

```
(ps 'gm :vn* (list violin) :vc (list cello) :p (list piano-rH piano-lH) :tempo 120 :time-signature '(4 4))
(init-seed nil)
)
```

; Mit 'gm :vn* erklingt der Midi-Klang für Solo-Violine. Sie Utilities: L: GM Instrument Set.lisp
; Mit (list Vn-Part ... werden die einzelnen Stimmen zu einer Partitur zusammengefügt.

```
; Evaluieren: cmd-e
; [36"]
```

--

5)

;; Klaviertrio 3.opmo, 3. Version 3b (11.07.2022): 7. Teil (Takte 128-147).

; Vn, Vc u. Klavier r.H. haben den gleichen Rhythmus, Klavier l.H. spielt einzelne Achtel.
Chromatische Tonleitern ohne Tonrepetitionen.
; Wie 3a, aber kürzere Phrasen und längere Pausen

```
(progn
(init-seed 33)
(setf Rhy-Tutti (rnd-sample 39 '((s -e.) (s s -q) (s s s -q.)(-q -q))))
(setf Rhy-Pf-lH (rnd-sample 99 '((e) (-q))))
```

```
(setf count-Tutti (mclength Rhy-Tutti))
```

```
(setf count-Pf-lH (mclength Rhy-Pf-lH))
```

; mclength zählt die Anzahl Ereignisse einer definierten Variablen, z.B. von Rhy-Tutti (4 oder 6 oder 8 Ereignisse in (s s s -e) usw.

; Damit es funktioniert, müssen die Rhy-Variablen aber nicht nur eine Klammer haben, sondern eine Gruppen von Klammern sein, also nicht:

```
; (setf Rhy-Pf-rH (rnd-sample 50 '(e -e))), sondern
```

```
; (setf Rhy-Pf-rH (rnd-sample 50 '((e) (-e)))). Somit kann er jeweils die Anzahl pro Klammer anzeigen.
```

; mclength wird also auf die folgende Sequenz angewandt.

; Bei (setf Rhy-Sop (rnd-sample 50 '(q q. h h. -e))) müßte man nur "count-Sop (length" schreiben, weil dann nur eine Gesamtanzahl ausgegeben wird.

```
(setf Int-Vn (gen-repeat count-Tutti (mclist (gen-weight 50 '((-1 9)(1 4)))))
```

```
(setf Int-Vc (gen-repeat count-Tutti (mclist (gen-weight 50 '((-1 9)(1 4)))))
```

```
(setf Int-Pf-rH (gen-repeat count-Tutti (mclist (gen-weight 50 '((-1 9)(1 4)))))
```

```
(setf Int-Pf-lH (gen-repeat count-Pf-lH (mclist (gen-weight 50 '((-1 9)(1 9)))))
```

; gen-weight gewichtet die Intervall-Häufigkeit: (-1 9) bedeutet: fallende kleine Sekunde 9x gewichtet.

; mclist = mapcar list, damit nicht eine lange Liste mit (1 -1 -1 1 usw.) entsteht, sondern damit jeder ausgegebene Wert in einer eigenen Liste / Klammer steht ((1)(-1)(-1) usw.). Somit können diese Klammern den Rhythmus-Klammern zugeordnet werden. Z.B. gilt dann (1) für (s s s -e), und es werden nur steigende kl. Sekunden erklingen (vorerst aber nur 1).

; Vergleiche gen-weight ohne und mit mclist

; Mit "gen-repeat count-Vn" wird die Intervallfortschreitung (z.B. kl2 aufwärts) so oft wiederholt, wie in der Variablen "count-Vn" in "Rhy-Vn" rhythmische Ereignisse gezählt wurden. Also bei (s s s -e) 4x, wobei das 4. Mal als Pause nicht erklingt.

```
(setf Pitch-Vn (pitch-transpose 24 (interval-to-pitch Int-Vn)))
(setf Pitch-Vc (pitch-transpose 24 (interval-to-pitch Int-Vc)))
(setf Pitch-rH (pitch-transpose 24 (interval-to-pitch Int-Pf-rH)))
(setf Pitch-lH (pitch-transpose 24 (interval-to-pitch Int-Pf-lH)))
  ; interval-to-pitch verwandelt die Zahlenfolgen von Int-Vn in Intervalle.
  ; pitch transpose betrifft nur den Ausgangston. Ohne nähere Definition starten die Tonhöhen
  immer beim eingestrichenen "c" (c4).
  ; (setf Pitch-Sop (pitch-transpose 12 (interval-to-pitch Int-Sop)))
  ; (setf Pitch-Klav (interval-to-pitch Int-Klav))
```

```
(setf Vn-Part (make-omn
  :pitch Pitch-Vn
  :length Rhy-Tutti
  :velocity '(f)
  :swallow t
))
```

```
(setf Vc-Part (make-omn
  :pitch Pitch-Vc
  :length Rhy-Tutti
  :velocity '(f)
  :swallow t
))
```

```
(setf Pf-rH-Part (make-omn
  :pitch Pitch-rH
  :length Rhy-Tutti
  :velocity '(f)
  :swallow t
))
```

```
(setf Pf-lH-Part (make-omn
  :pitch Pitch-lH
  :length Rhy-Pf-lH
  :velocity '(f)
  :swallow t
))
```

; Tonhöhen und Dauern werden pro Stimme zusammengefügt.

```
(setf violin (ambitus '(g3 c7) Vn-Part))
(setf cello (ambitus '(c2 c5) Vc-Part))
(setf piano-rH (ambitus '(c4 c6) Pf-rH-Part))
(setf piano-lH (ambitus '(fs1 b3) Pf-lH-Part))
```

```
(ps 'gm :vn* (list violin) :vc (list cello) :p (list piano-rH piano-lH) :tempo 120 :time-signature '(4 4))
(init-seed nil)
)
```

; Mit 'gm :vn* erklingt der Midi-Klang für Solo-Violine. Sie Utilities: L: GM Instrument Set.lisp
; Mit (list Vn-Part ... werden die einzelnen Stimmen zu einer Partitur zusammengefügt.

```
; Evaluieren: cmd-e
; [36"]
```

```
--
```

6)

;; Klaviertrio 3.opmo, Version 4d (13.7.2022): 8. Teil (Takte 148-188)

; Vn, Vc u. Klavier r.H. haben den gleichen Rhythmus, Klavier l.H. spielt einzelne Achtel.

Einfache Tonrepetitionen, d.h. pro Phrase eine Tonhöhe.

; Wie Version 4a, aber längere Tonrepetitionen, die manchmal auch ohne Pausen aneinander anschließen können. Längeres Nachspiel.

```
(progn
  (init-seed 33)
  (setf Rhy-Tutti (rnd-sample 39 '((s s s s s) (s s s s s s) (s s s s s s s -s) (s s s s s s s s -q) (s s s s s s s s s -q.)
  (s s s s s s s s s -q.) (-q) (-q) (-q) (-q) (-q) (-q))))
  (setf Rhy-Pf-lH (rnd-sample 200 '((e) (-q))))
```

```
(setf count-Tutti (mclength Rhy-Tutti))
```

```
(setf count-Pf-rH (mclength Rhy-Pf-lH))
```

; mclength zählt die Anzahl Ereignisse einer definierten Variablen, z.B. von Rhy-Tutti (4 oder 6 oder 8 Ereignisse in (s s s -e) usw.

; Damit es funktioniert, müssen die Rhy-Variablen aber nicht nur eine Klammer haben, sondern eine Gruppen von Klammern sein, also nicht:

```
; (setf Rhy-Pf-rH (rnd-sample 50 '(e -e))), sondern
```

```
; (setf Rhy-Pf-rH (rnd-sample 50 '((e) (-e))). Somit kann er jeweils die Anzahl pro Klammer anzeigen.
```

; mclength wird also auf die folgende Sequenz angewandt.

; Bei (setf Rhy-Sop (rnd-sample 50 '(q q. h h. -e))) müßte man nur "count-Sop (length" schreiben, weil dann nur eine Gesamtanzahl ausgegeben wird.

```
(setf Int-Vn (gen-repeat count-Tutti (mclist (gen-weight 50 '((0 9)(0 4)))))
```

```
(setf Int-Vc (gen-repeat count-Tutti (mclist (gen-weight 50 '((0 9)(0 4)))))
```

```
(setf Int-Pf-rH (gen-repeat count-Tutti (mclist (gen-weight 50 '((0 9)(0 4)))))
```

```
(setf Int-Pf-lH (gen-repeat count-Pf-rH (mclist (gen-weight 50 '((0 9)(0 9)))))
```

; gen-weight gewichtet die Intervall-Häufigkeit: (-1 9) bedeutet: fallende kleine Sekunde 9x gewichtet.

; mclist = mapcar list, damit nicht eine lange Liste mit (1 -1 -1 1 usw.) entsteht, sondern damit jeder ausgegebene Wert in einer eigenen Liste / Klammer steht ((1)(-1)(-1) usw.). Somit können diese Klammern den Rhythmus-Klammern zugeordnet werden. Z.B. gilt dann (1) für (s s s -e), und es werden nur steigende kl. Sekunden erklingen (vorerst aber nur 1).

; Vergleiche gen-weight ohne und mit mclist

; Mit "gen-repeat count-Vn" wird die Intervallfortschreitung (z.B. kl2 aufwärts) so oft wiederholt, wie in der Variablen "count-Vn" in "Rhy-Vn" rhythmische Ereignisse gezählt wurden. Also bei (s s s -e) 4x, wobei das 4. Mal als Pause nicht erklingt.

```
(setf Pitch-Vn (pitch-transpose 24 (interval-to-pitch Int-Vn)))
```

```
(setf Pitch-Vc (pitch-transpose 24 (interval-to-pitch Int-Vc)))
```

```
(setf Pitch-rH (pitch-transpose 24 (interval-to-pitch Int-Pf-rH)))
```

```
(setf Pitch-IH (pitch-transpose 24 (interval-to-pitch Int-Pf-IH)))
; interval-to-pitch verwandelt die Zahlenfolgen von Int-Vn in Intervalle.
; pitch transpose betrifft nur den Ausgangston. Ohne nähere Definition starten die Tonhöhen
immer beim eingestrichenen "c" (c4).
; (setf Pitch-Sop (pitch-transpose 12 (interval-to-pitch Int-Sop)))
; (setf Pitch-Klav (interval-to-pitch Int-Klav))
```

```
(setf Vn-Part (make-omn
:pitch Pitch-Vn
:length Rhy-Tutti
:velocity '(f)
:swallow t
```

```
))
```

```
(setf Vc-Part (make-omn
:pitch Pitch-Vc
:length Rhy-Tutti
:velocity '(f)
:swallow t
```

```
))
```

```
(setf Pf-rH-Part (make-omn
:pitch Pitch-rH
:length Rhy-Tutti
:velocity '(f)
:swallow t
```

```
))
```

```
(setf Pf-lH-Part (make-omn
:pitch Pitch-lH
:length Rhy-Pf-lH
:velocity '(f)
:swallow t
```

```
))
```

; Tonhöhen und Dauern werden pro Stimme zusammengefügt.

```
(setf list-int11 (rnd-sample 10 '(1 -1)))
```

```
(setf list-int12 (rnd-sample 10 '(1 -1)))
```

```
(setf list-int13 (rnd-sample 10 '(1 -1)))
```

```
(setf list-int14 (rnd-sample 10 '(1 -1)))
```

; Eine Liste von Intervallen, die zwischen den Gruppen zur Anwendung kommen sollen.

```
(setf violin (ambitus '(g3 c7)(bind-to-interval list-int11 Vn-Part)))
```

```
(setf cello (ambitus '(c2 c5)(bind-to-interval list-int12 Vc-Part)))
```

```
(setf piano-rH (ambitus '(c4 c6) (bind-to-interval list-int13 Pf-rH-Part)))
```

```
(setf piano-lH (ambitus '(fs1 b3) (bind-to-interval list-int14 Pf-lH-Part)))
```

; bind-to-interval: bind (connect) a series of lists (patterns) to each-other with the given interval
bzw. with given intervals

```
(ps 'gm :vn* (list violin) :vc (list cello) :p (list piano-rH piano-lH) :tempo 120 :time-signature '(4 4))
```

```
(init-seed nil)
```

```
)
```

; Mit 'gm :vn* erklingt der Midi-Klang für Solo-Violine. Sie Utilities: L: GM Instrument Set.lisp
; Mit (list Vn-Part ... werden die einzelnen Stimmen zu einer Partitur zusammengefügt.

```
; Evaluieren: cmd-e
; [1:15]
```

```
--
```

7)

;; Klaviertrio 3.opmo, 3. Version 5b (14.07.2022): 9. Teil (Takte 189-229).

```
; Wie 5a, aber mit Stimmtausch: Duo Vn-Vc ohne Klavier (der Einfachheit halber als
Klaviersystem zusammengefaßt). Vc spielt zuerst fallende, dann steigende, dann wieder fallende
Chromatik, 4-5-6-13, Vn spielt (wie vorher in der l.H. des Klaviers) einzelne Achtel. Keine
Tonrepetitionen.
```

```
(progn
  (init-seed 34)
  (setf rhy-tutti (rnd-sample 50 '((s s s s -e) (s s s s s -e.) (s s s s s s -q.) (s s s s s s s s s s s s s s -e. -h))))
  (setf rhy-pf-lh (rnd-sample 127 '((e) (e) (e) (-e) (-q))))
```

```
(setf count-tutti (mclength rhy-tutti))
(setf count-pf-lh (mclength rhy-pf-lh))
```

```
(setf int-vn (gen-repeat count-tutti (gen-weight 50 '((-1 1)(-1 1))))
(setf int-pf-lh (gen-repeat count-pf-lh (gen-weight 50 '((1 1)(-1 1))))
```

```
(setf pitch-tutti (pitch-transpose 24 (interval-to-pitch int-vn)))
(setf pitch-lh (pitch-transpose 24 (interval-to-pitch int-pf-lh)))
; interval-to-pitch verwandelt die Zahlenfolgen von Int-Vn in Intervalle.
; pitch transpose betrifft nur den Ausgangston. Ohne nähere Definition starten die Tonhöhen
immer beim eingestrichenen "c" (c4).
```

```
(setf tutti-part (make-omn
  :pitch pitch-tutti
  :length rhy-tutti
  :velocity '(pp)
))
```

```
(setf pf-lh-part (make-omn
  :pitch pitch-lh
  :length rhy-pf-lh
  :velocity '(pp)
))
```

```
; Tonhöhen und Dauern werden pro Stimme zusammengefügt.
```

```
(setf tutti-amb (ambitus '(c2 b3) tutti-part))
(setf piano-lh-amb (ambitus '(c4 c6) pf-lh-part))
```

```
(ps 'gm :p (list tutti-amb piano-lh-amb) :tempo 120 :time-signature '(4 4))
(init-seed nil)
```

)
 ; Mit 'gm :vn* erklingt der Midi-Klang für Solo-Violine. Sie Utilities: L: GM Instrument Set.lisp
 ; Mit (list Vn-Part ... werden die einzelnen Stimmen zu einer Partitur zusammengefügt.

; Evaluieren: cmd-e
 ; [beide: 34", mit Vc-Nachspiel: 1:21]

; --

8)

;; Klaviertrio 3.opmo, Version 4e (14.7.2022): 10. Teil (Takte 230-250)

; 10. Teil: Wie Version 4a, aber keine Tonrepetitionen, sondern Blöcke mit längeren Pausen zwischen jedem Block. Nur einzelne Punkte. Manchmal Zweiergruppen. Dreiklänge im Klavier (ev. auch nur Tritoni), Zweiklänge in Violine und Cello (Tritoni).

```
(progn
  (init-seed 35)
  (setf Rhy-Tutti (rnd-sample 39 '((s -e.) (s -e. -q) (s -e. -h) (e e -h.) (-q) (-q) (-q) (-q))))
  (setf Rhy-Pf-IH (rnd-sample 200 '((e) (-q))))
```

```
(setf count-Tutti (mclength Rhy-Tutti))
```

```
(setf count-Pf-rH (mclength Rhy-Pf-IH))
```

; mclength zählt die Anzahl Ereignisse einer definierten Variablen, z.B. von Rhy-Tutti (4 oder 6 oder 8 Ereignisse in (s s s -e) usw.

; Damit es funktioniert, müssen die Rhy-Variablen aber nicht nur eine Klammer haben, sondern eine Gruppen von Klammern sein, also nicht:

; (setf Rhy-Pf-rH (rnd-sample 50 '(e -e))), sondern

; (setf Rhy-Pf-rH (rnd-sample 50 '((e) (-e)))). Somit kann er jeweils die Anzahl pro Klammer anzeigen.

; mclength wird also auf die folgende Sequenz angewandt.

; Bei (setf Rhy-Sop (rnd-sample 50 '(q q. h h. -e))) müßte man nur "count-Sop (length" schreiben, weil dann nur eine Gesamtanzahl ausgegeben wird.

```
(setf Int-Vn (gen-repeat count-Tutti (mclist (gen-weight 50 '((0 9)(0 4)))))
```

```
(setf Int-Vc (gen-repeat count-Tutti (mclist (gen-weight 50 '((0 9)(0 4)))))
```

```
(setf Int-Pf-rH (gen-repeat count-Tutti (mclist (gen-weight 50 '((0 9)(0 4)))))
```

```
(setf Int-Pf-IH (gen-repeat count-Pf-rH (mclist (gen-weight 50 '((0 9)(0 9)))))
```

; gen-weight gewichtet die Intervall-Häufigkeit: (-1 9) bedeutet: fallende kleine Sekunde 9x gewichtet.

; mclist = mapcar list, damit nicht eine lange Liste mit (1 -1 -1 1 usw.) entsteht, sondern damit jeder ausgegebene Wert in einer eigenen Liste / Klammer steht ((1)(-1)(-1) usw.). Somit können diese Klammern den Rhythmus-Klammern zugeordnet werden. Z.B. gilt dann (1) für (s s s -e), und es werden nur steigende kl. Sekunden erklingen (vorerst aber nur 1).

; Vergleiche gen-weight ohne und mit mclist

; Mit "gen-repeat count-Vn" wird die Intervallfortschreitung (z.B. kl2 aufwärts) so oft wiederholt, wie in der Variablen "count-Vn" in "Rhy-Vn" rhythmische Ereignisse gezählt wurden. Also bei (s s s -e) 4x, wobei das 4. Mal als Pause nicht erklingt.

```
(setf Pitch-Vn (pitch-transpose 24 (interval-to-pitch Int-Vn)))
```

```
(setf Pitch-Vc (pitch-transpose 24 (interval-to-pitch Int-Vc)))
(setf Pitch-rH (pitch-transpose 24 (interval-to-pitch Int-Pf-rH)))
(setf Pitch-lH (pitch-transpose 24 (interval-to-pitch Int-Pf-lH)))
  ; interval-to-pitch verwandelt die Zahlenfolgen von Int-Vn in Intervalle.
  ; pitch transpose betrifft nur den Ausgangston. Ohne nähere Definition starten die Tonhöhen
  immer beim eingestrichenen "c" (c4).
  ; (setf Pitch-Sop (pitch-transpose 12 (interval-to-pitch Int-Sop)))
  ; (setf Pitch-Klav (interval-to-pitch Int-Klav))
```

```
(setf Vn-Part (make-omn
  :pitch Pitch-Vn
  :length Rhy-Tutti
  :velocity '(f)
  :swallow t
))
```

```
(setf Vc-Part (make-omn
  :pitch Pitch-Vc
  :length Rhy-Tutti
  :velocity '(f)
  :swallow t
))
```

```
(setf Pf-rH-Part (make-omn
  :pitch Pitch-rH
  :length Rhy-Tutti
  :velocity '(f)
  :swallow t
))
```

```
(setf Pf-lH-Part (make-omn
  :pitch Pitch-lH
  :length Rhy-Pf-lH
  :velocity '(f)
  :swallow t
))
```

; Tonhöhen und Dauern werden pro Stimme zusammengefügt.

```
(setf list-int11 (rnd-sample 10 '(1 -1)))
(setf list-int12 (rnd-sample 10 '(1 -1)))
(setf list-int13 (rnd-sample 10 '(1 -1)))
(setf list-int14 (rnd-sample 10 '(1 -1)))
```

; Eine Liste von Intervallen, die zwischen den Gruppen zur Anwendung kommen sollen.

```
(setf violin (ambitus '(c5 c7)(bind-to-interval list-int11 Vn-Part)))
(setf cello (ambitus '(c3 b4)(bind-to-interval list-int12 Vc-Part)))
(setf piano-rH (ambitus '(c4 b5) (bind-to-interval list-int13 Pf-rH-Part)))
(setf piano-lH (ambitus '(fs1 b2) (bind-to-interval list-int14 Pf-lH-Part)))
  ; bind-to-interval: bind (connect) a series of lists (patterns) to each-other with the given interval
  bzw. with given intervals
```

```
(ps 'gm :vn* (list violin) :vc (list cello) :p (list piano-rH piano-lH) :tempo 120 :time-signature '(4 4))
(init-seed nil)
)
```

; Mit 'gm :vn* erklingt der Midi-Klang für Solo-Violine. Sie Utilities: L: GM Instrument Set.lisp
 ; Mit (list Vn-Part ... werden die einzelnen Stimmen zu einer Partitur zusammengefügt.

; Evaluieren: cmd-e

;--

9)

;; **Klaviertrio 3.opmo, 3. Version 3c (15.07.2022):** 11. Teil, Takte 251-279

; Vn, Vc u. Klavier r.H. haben den gleichen Rhythmus, Klavier l.H. spielt einzelne Achtel.
 Chromatische Tonleitern ohne Tonrepetitionen.

; Wie Version 3a bzw. wie 3b bzw. wie 7. Teil, aber noch kürzere Phrasen und noch längere
 Pausen.

(progn

(init-seed 35)

(setf Rhy-Tutti (rnd-sample 39 '((s -e. -e.) (s s -q -q) (s s s -q. -q.)(-q -q -q))))

(setf Rhy-Pf-lH (rnd-sample 99 '((e) (-q -q))))

(setf count-Tutti (mclength Rhy-Tutti))

(setf count-Pf-lH (mclength Rhy-Pf-lH))

; mclength zählt die Anzahl Ereignisse einer definierten Variablen, z.B. von Rhy-Tutti (4 oder 6
 oder 8 Ereignisse in (s s s -e) usw.

; Damit es funktioniert, müssen die Rhy-Variablen aber nicht nur eine Klammer haben, sondern
 eine Gruppen von Klammern sein, also nicht:

; (setf Rhy-Pf-rH (rnd-sample 50 '(e -e))), sondern

; (setf Rhy-Pf-rH (rnd-sample 50 '((e) (-e))). Somit kann er jeweils die Anzahl pro Klammer
 anzeigen.

; mclength wird also auf die folgende Sequenz angewandt.

; Bei (setf Rhy-Sop (rnd-sample 50 '(q q. h h. -e))) müßte man nur "count-Sop (length" schreiben,
 weil dann nur eine Gesamtanzahl ausgegeben wird.

(setf Int-Vn (gen-repeat count-Tutti (mclist (gen-weight 50 '((-1 9)(1 4)))))

(setf Int-Vc (gen-repeat count-Tutti (mclist (gen-weight 50 '((-1 9)(1 4)))))

(setf Int-Pf-rH (gen-repeat count-Tutti (mclist (gen-weight 50 '((-1 9)(1 4)))))

(setf Int-Pf-lH (gen-repeat count-Pf-lH (mclist (gen-weight 50 '((-1 9)(1 9)))))

; gen-weight gewichtet die Intervall-Häufigkeit: (-1 9) bedeutet: fallende kleine Sekunde 9x
 gewichtet.

; mclist = mapcar list, damit nicht eine lange Liste mit (1 -1 -1 1 usw.) entsteht, sondern damit
 jeder ausgegebene Wert in einer eigenen Liste / Klammer steht ((1)(-1)(-1) usw.). Somit können
 diese Klammern den Rhythmus-Klammern zugeordnet werden. Z.B. gilt dann (1) für (s s s -e),
 und es werden nur steigende kl. Sekunden erklingen (vorerst aber nur 1).

; Vergleiche gen-weight ohne und mit mclist

; Mit "gen-repeat count-Vn" wird die Intervallfortschreitung (z.B. kl2 aufwärts) so oft wiederholt,
 wie in der Variablen "count-Vn" in "Rhy-Vn" rhythmische Ereignisse gezählt wurden. Also bei (s
 s s -e) 4x, wobei das 4. Mal als Pause nicht erklingt.

(setf Pitch-Vn (pitch-transpose 24 (interval-to-pitch Int-Vn)))

(setf Pitch-Vc (pitch-transpose 24 (interval-to-pitch Int-Vc)))

```

(setf Pitch-rH (pitch-transpose 24 (interval-to-pitch Int-Pf-rH)))
(setf Pitch-lH (pitch-transpose 24 (interval-to-pitch Int-Pf-lH)))
  ; interval-to-pitch verwandelt die Zahlenfolgen von Int-Vn in Intervalle.
  ; pitch transpose betrifft nur den Ausgangston. Ohne nähere Definition starten die Tonhöhen
  immer beim eingestrichenen "c" (c4).
  ; (setf Pitch-Sop (pitch-transpose 12 (interval-to-pitch Int-Sop)))
  ; (setf Pitch-Klav (interval-to-pitch Int-Klav))

(setf Vn-Part (make-omn
  :pitch Pitch-Vn
  :length Rhy-Tutti
  :velocity '(p)
  :swallow t
))
(setf Vc-Part (make-omn
  :pitch Pitch-Vc
  :length Rhy-Tutti
  :velocity '(p)
  :swallow t
))
(setf Pf-rH-Part (make-omn
  :pitch Pitch-rH
  :length Rhy-Tutti
  :velocity '(p)
  :swallow t
))
(setf Pf-lH-Part (make-omn
  :pitch Pitch-lH
  :length Rhy-Pf-lH
  :velocity '(p)
  :swallow t
))
))
  ; Tonhöhen und Dauern werden pro Stimme zusammengefügt.

(setf violin (ambitus '(g3 c7) Vn-Part))
(setf cello (ambitus '(c2 c5) Vc-Part))
(setf piano-rH (ambitus '(c4 c6) Pf-rH-Part))
(setf piano-lH (ambitus '(fs1 b3) Pf-lH-Part))

(ps 'gm :vn* (list violin) :vc (list cello) :p (list piano-rH piano-lH) :tempo 120 :time-signature '(4 4))
(init-seed nil)
)
  ; Mit 'gm :vn* erklingt der Midi-Klang für Solo-Violine. Sie Utilities: L: GM Instrument Set.lisp
  ; Mit (list Vn-Part ... werden die einzelnen Stimmen zu einer Partitur zusammengefügt.

  ; Evaluieren: cmd-e
  ; [59"]

; --

```


10)

;; Klaviertrio 3.opmo, 3. Version 3d (19.07.2022): 12. Teil, Takte 280-298

; Vn, Vc u. Klavier r.H. haben den gleichen Rhythmus, Klavier l.H. spielt einzelne Achtel. Mischung von Tonwiederholungen und Läufen. Viele Gegenbewegungen.

```
(progn
  (init-seed 33)
  (setf Rhy-Tutti (rnd-sample 39 '((s s s -s) (s s s s -e) (s s s s s -e.)(-q))))
  (setf Rhy-Pf-lH (rnd-sample 99 '((e) (-q))))
```

```
(setf count-Tutti (mclength Rhy-Tutti))
(setf count-Pf-lH (mclength Rhy-Pf-lH))
```

; mclength zählt die Anzahl Ereignisse einer definierten Variablen, z.B. von Rhy-Tutti (4 oder 6 oder 8 Ereignisse in (s s s -e) usw.
 ; Damit es funktioniert, müssen die Rhy-Variablen aber nicht nur eine Klammer haben, sondern eine Gruppen von Klammern sein, also nicht:
 ; (setf Rhy-Pf-rH (rnd-sample 50 '(e -e))), sondern
 ; (setf Rhy-Pf-rH (rnd-sample 50 '((e) (-e)))). Somit kann er jeweils die Anzahl pro Klammer anzeigen.
 ; mclength wird also auf die folgende Sequenz angewandt.
 ; Bei (setf Rhy-Sop (rnd-sample 50 '(q q. h h. -e))) müßte man nur "count-Sop (length" schreiben, weil dann nur eine Gesamtanzahl ausgegeben wird.

```
(setf Int-Vn (gen-repeat count-Tutti (mclist (gen-weight 50 '((-1 9)(1 4)))))
(setf Int-Vc (gen-repeat count-Tutti (mclist (gen-weight 50 '((-1 9)(1 4)))))
(setf Int-Pf-rH (gen-repeat count-Tutti (mclist (gen-weight 50 '((-1 9)(1 4)))))
(setf Int-Pf-lH (gen-repeat count-Pf-lH (mclist (gen-weight 50 '((-1 9)(1 9)))))
```

; gen-weight gewichtet die Intervall-Häufigkeit: (-1 9) bedeutet: fallende kleine Sekunde 9x gewichtet.
 ; mclist = mapcar list, damit nicht eine lange Liste mit (1 -1 -1 1 usw.) entsteht, sondern damit jeder ausgegebene Wert in einer eigenen Liste / Klammer steht ((1)(-1)(-1) usw.). Somit können diese Klammern den Rhythmus-Klammern zugeordnet werden. Z.B. gilt dann (1) für (s s s -e), und es werden nur steigende kl. Sekunden erklingen (vorerst aber nur 1).
 ; Vergleiche gen-weight ohne und mit mclist
 ; Mit "gen-repeat count-Vn" wird die Intervallfortschreitung (z.B. kl2 aufwärts) so oft wiederholt, wie in der Variablen "count-Vn" in "Rhy-Vn" rhythmische Ereignisse gezählt wurden. Also bei (s s s -e) 4x, wobei das 4. Mal als Pause nicht erklingt.

```
(setf Pitch-Vn (pitch-transpose 24 (interval-to-pitch Int-Vn)))
(setf Pitch-Vc (pitch-transpose 24 (interval-to-pitch Int-Vc)))
(setf Pitch-rH (pitch-transpose 24 (interval-to-pitch Int-Pf-rH)))
(setf Pitch-lH (pitch-transpose 24 (interval-to-pitch Int-Pf-lH)))
```

; interval-to-pitch verwandelt die Zahlenfolgen von Int-Vn in Intervalle.
 ; pitch transpose betrifft nur den Ausgangston. Ohne nähere Definition starten die Tonhöhen immer beim eingestrichenen "c" (c4).
 ; (setf Pitch-Sop (pitch-transpose 12 (interval-to-pitch Int-Sop)))
 ; (setf Pitch-Klav (interval-to-pitch Int-Klav))

```
(setf Vn-Part (make-omn
  :pitch Pitch-Vn
  :length Rhy-Tutti
  :velocity '(mf)
```

```

))
(setf Vc-Part (make-omn
               :pitch Pitch-Vc
               :length Rhy-Tutti
               :velocity '(mf)
))
(setf Pf-rH-Part (make-omn
                  :pitch Pitch-rH
                  :length Rhy-Tutti
                  :velocity '(mf)
))
(setf Pf-lH-Part (make-omn
                  :pitch Pitch-lH
                  :length Rhy-Pf-lH
                  :velocity '(mf)
))
; Tonhöhen und Dauern werden pro Stimme zusammengefügt.

(setf violin (ambitus '(g3 c7) Vn-Part))
(setf cello (ambitus '(c2 c5) Vc-Part))
(setf piano-rH (ambitus '(c4 c6) Pf-rH-Part))
(setf piano-lH (ambitus '(fs1 b3) Pf-lH-Part))

(ps 'gm :vn* (list violin) :vc (list cello) :p (list piano-rH piano-lH) :tempo 120 :time-signature '(4 4))
(init-seed nil)
)
; Mit 'gm :vn* erklingt der Midi-Klang für Solo-Violine. Sie Utilities: L: GM Instrument Set.lisp
; Mit (list Vn-Part ... werden die einzelnen Stimmen zu einer Partitur zusammengefügt.

; Evaluieren: cmd-e
; [36"]

```

Vorgehen

Brainstorming / Planung / Texte

Idee: Violine, Cello und Klavier rH spielen die gleichen Rhythmen, aber verschiedene Töne. Klavier l.H. spielt einzelne Achtel. Wie 1. Version, aber mit gleichzeitigen Viertel-Pausen in Vn-Vc-Klavier rH (23.6.2022)

Weitere Möglichkeit: einfache Tonrepetition, d.h. pro Phrase eine Tonhöhe

Zusammengesetzte Tonrepetitionen, d.h. pro Phrase verschiedene Tonhöhen 2er, 3er oder 4er Repetitionen in 5er, 6er oder 7er Phrasen (25.6.2022)

3. Teil: Nur fallende Chromatik. Die Instrumente nacheinander. 5/7/13

4. Teil: Akkorde aller Instrumente, einzelne Linien gehen noch weiter

Brainstorming-Ende [bre]

Form

Siehe auch " Fassungen".

- 1. Teil (Takte 1-27): Tonrepetitionen mit größeren Pausen zwischen Phrasengruppen. Violine, Cello und rechte Hand Klavier rhythmisch synchron. Linke Hand Klavier dazu Achtel. (Opusmodus-Version 4a.) Takt 14: Zwischentakt. Takte 22-27: Auslaufen.
- 2. Teil (Takte 28-47): Wie 1. Teil, aber keine größeren Pausen zwischen den Phrasengruppen. (Opusmodus-Version 4c.) Takte 46-47: Auslaufen.
- 3. Teil (Takte 48-72): Duo Violine-Cello: Violine zuerst nur Abwärtsbewegungen, dann nur Aufwärtsbewegungen, dann wieder Abwärtsbewegungen, schließlich beim Auslaufen nur Aufwärtsbewegungen, Cello dazu Pizzicato-Achtel. (Opusmodus-Version 5a.)
- 4. Teil (Takte 73-98): Violine und Cello Gegenbewegungen, Klavier Akkorde. (Ohne OM.)
- ~~5. Teil (Takte 91-106): Violine und Cello gehen alleine weiter. (Takt 91: Phrasen-Verschränkung.)~~
- 5. Teil (Takte 99-108): Es geht ohne Klavierakkorde imitatorisch weiter und läuft bald aus. (Ohne OM.)
- 6. Teil (Takte 109-127): Ähnlich wie 1. Teil, aber nicht Tonrepetitionen, sondern chromatische Tonleitern. (Opusmodus-Version 3a.)
- 7. Teil (Takte 128-147) Wie 6. Teil, aber kürzere Phrasen und längere Pausen. (Opusmodus-Version 3b.)
- 8. Teil (Takte 148-188): Wie Version 4a, aber längere Tonrepetitionen, die manchmal ohne Pausen aneinander anschließen. Längeres Nachspiel. Die Wiederaufnahme der Tonrepetitionen markiert die Mitte des Stückes. (Opusmodus-Version 4d.)
- 9. Teil (Takte 189-229): Wie 5a, aber mit Stimmtausch: Duo Vn-Vc ohne Klavier (in OM der Einfachheit halber als Klaviersystem zusammengefaßt). Vc spielt zuerst fallende, dann steigende, dann wieder fallende Chromatik, 4-5-6-13 Töne-Phrasen, Vn spielt (wie vorher in der l.H. des Klaviers) einzelne Achtel. Keine Tonrepetitionen. Längeres Cello-Nachspiel. Am Ende (ab Takt 224) rutscht das Cello ins tiefste Register hinab. (Opusmodus-Version 5b.)
- 10. Teil (Takte 230-250): Wie Version 4a, aber keine Tonrepetitionen, sondern Blöcke mit längeren Pausen zwischen jedem Block. Nur einzelne Punkte. Manchmal Zweiergruppen. Dreiklänge im Klavier geplant, aber nur Tritoni ausgeführt, Zweiklänge in Violine und Cello (Tritoni). (Opusmodus-Version 4e.)
- 11. Teil (Takte 251-279): Wie Version 3a bzw. wie 3b bzw. wie 7. Teil, aber noch kürzere Phrasen und noch längere Pausen. (Opusmodus-Version 3c.)
- 12. Teil (Takte 280-298): Wie Version 3a, aber jetzt erstmals eine Mischung von Tonwiederholungen und Läufen. (Am Schluß: Kombination chrom. Tonleitern und Tonrepetitionen.) Viele Gegenbewegungen, aber auch Unisoni. (Opusmodus-Version 3d.) Ab Takt 299 folgt die Coda.

Kontrollen:

- **xml-Import-Kontrollen (Grobeditierung):**
 - Die Wiedergabe des Tempos einstellen.
 - Die Wiedergabe der dynamischen Angaben einstellen.
 - Die Wiedergabe der Instrumente einstellen.
 - Die Marginalien editieren und einstellen.

- Titel (24, fett), Seitenbeschriftung (Größe: 12, Stil: normal, Seitenbereich 2 bis ...), Seitenzahlen (Links H: 0, V: 0.15278 **bzw. gleich hoch wie die Seitenbeschriftung** (meistens 0.22222). Verwende Positionierung für rechte Seiten: Rechts: dito, Seitenbereich 2 bis ...).
- Layout grob editieren.
- Das Finale-Dokument einrichten (Hilfswerkzeuge programmieren, Olen einrichten, Halslänge korrigieren: Fin12: Dokument: Dokument-Optionen: Notenhäse. Normale Halslänge 0.35167 statt 0.29167 Zoll, verkürzte Halslänge 0.27611 statt 0.23611 Zoll (Maßeinheiten: Einstellungen unter „Finale 2012“)). -> Erst bei der definitiven Version machen.
- Takte neu einteilen, ev. mehr Nachteile.

Normale Kontrollen:

- ✓ Die Taktgruppen fixieren. (Bei Solostücken nach dem Festlegen der Wendestellen.): Gemacht. Seitenwendestellen in diesem Stück:
- Hinweis: Entweder „Partitur klingend notiert / Partitur in C“ oder bei den transponierenden Instrumenten: „Klingend notiert“.
- Wo Klarinette statt Baßklarinette, wo Flöte statt Baßflöte?
- Den Wechsel von Klarinette zu Baßklarinette (bzw. Flöte zu Baßflöte usw.) in der Partitur mit dem Instrumentenwechsel von Finale machen (Extras: Instrumentenwechsel).
- Vorzeichen vor jeden Ton? Nur in extrem chromatisierter Musik (siehe Aph. 10.10.2015).
- Vorzeichenkontrolle: Kommt ein alterierter Ton im Takt nochmals vor, alteriert oder unalteriert, dann müssen Versetzungszeichen gesetzt werden. (Kontrollieren, daß einmal alterierte Töne aufgelöst werden, wenn sie im gleichen Takt in unalterierter Form wiederkommen.) Alles durchlesen. Gemacht bis Schluß in Takt 307.
- Bzw. sind alle Vorzeichen sichtbar (Vorzeichen-Wiederholungen im gleichen Takt), besonders bei den Akkordballungen?
- Anfangs des folgenden Taktes ein Sicherheits-Auflösungszeichen, wenn kurz davor eine Alteration stattfindet.
- Haltetöne am Anfang der Zeile: Vorzeichen in Klammern. Nur am Seitenanfang.
- Balken durchbrechen und Pausen zusammenfassen. Werden zwischen zwei Teilen eines Achtels (wobei jeder Teil einen Gesamtwert von einem **Sechzehntel** hat) die Zweitbalken durchbrochen, so muß der **Sechzehntelbalken** stehen bleiben. Gemacht bis Takt 306 (Schluß).
- Balken-Richtung (steigende Melodie = steigende Balkenrichtung). Gemacht bis Takt 147 (7. Teil).
- ✓ Kontrapunkt-Kontrolle. Gelesen.
- ✓ Kontrapunkt-Kontrolle. Am Klavier gespielt.
- Dynamik ausdifferenzieren. Gemacht Takte ...
- ✓ Bei Vc etc. den richtigen Schlüssel (Tenorschlüssel statt Violinschlüssel). Beim Schlüsselwechsel automatische Musikausrichtung.
Beim direkten Wechsel vom Baß- zum Violinschlüssel im Cello „(loco)“ schreiben, damit im Violinschlüssel nicht eine Oktave tiefer gespielt wird (Casella S. 176).
- Détaché-Kontrolle (bzw. Artikulationskontrolle): Steht überall „détaché“, wo nicht legato gespielt werden soll? (Stehen überall die richtigen Artikulationszeichen?) Ist überall klar, wie gespielt werden soll?
- Nach „pont.“: pos. norm. oder ord.
- Nach „col legno battuto“: ord.
- Nach „pizz.“: „arco“
- Silbenverlängerungsstriche bearbeiten.
- Sind die Vokaltexsilben richtig unter den Noten? -> Nacheditieren. Gemacht.
- ✓ Taktinhalte kontrollieren.
- ✓ Instrumentenumfänge kontrollieren.
- Tempo-Vorankündigung am Ende des Systems (ohne hinterlegte Tempoänderung), wenn auf dem nächsten System das Tempo wechselt. (Bei zweistelligen Zahlen: Notenkopf über dem Ende der Notenzeile. Bei dreistelligen Zahlen: Noch 6 Klicks nach links.)

- Leere Notensysteme ausblenden?
- Tempo-Wechsel kontrollieren.
- Dirigierzeichen bei Taktwechseln einsetzen.
- Doppelstriche / Abschnitte im Bezug zu den Tempi kontrollieren.
- Die ausgedruckte Partitur mit der Fortlaufenden Ansicht vergleichen und kontrollieren, ob nichts verschluckt wurde. Ergibt sich automatisch beim Vergleich Einzelstimmen-Partitur.
- Alle beweglichen Schlüssel kontrollieren. Problemorte auflisten: Takte ...
- Die ganze Partitur durchgehen, inwieweit man noch mehr in Richtung korrekte proportionale Darstellung gehen kann (Abstände enger bzw. weiter machen).
- Den Rhythmus der beiden Stimmen synchronisieren.
- Mikrotöne-Kontrollen: 1) das Versetzungszeichen, 2) die Angabe / Definition, ob Viertel- oder Drittelton hoch oder tief, 3) Legatobogen, 4) Gliss.-Strich mit „gliss.“-Angabe.
Die mikrotonalen Abweichungen betragen ca. einen Drittelton ($3\uparrow, 3\downarrow$) bzw. ca. einen Viertelton ($4\uparrow, 4\downarrow$).
- Auflisten, was sich **beim Drucken** ab und zu verschiebt: -
- ✓ Musikwerknummer:
 - In: Dokumente: 1/Texte/Word: BAC Bio & Interview: Werklisten: Werkverz.Übersicht: Aktuelle Musikwerke-Verzeichnisse: Musikwerknummern 1152-, Infos.doc
 - In: Dokumente: 1/Texte/Word: BAC Bio & Interview: Werklisten: Werkverz.Übersicht: Aktuelle Musikwerke-Verzeichnisse: WerkverzMusikÜbersichtAktuell✓ (Excel)
 - In das Dokument „Musikwerknummern 1- Die komplette Musikwerkliste✓.xlsx“
 - An den Anfang dieses Protokoll-Dokuments.
- ✓ Auf die 1. Seite der Partitur: Den Titel fett setzen.
- ✓ Auf der 1. Seite: Untertitel (auf ein Gedicht von, Besetzung), Ergon, Jahr, Musikwerknummer, ev. Widmung
- Auf die 1. Seite unten links: Copyright (© Copyright 2019 by René Wohlhauser-Eigenverlag, CH-Basel, Edition Wohlhauser Nr. 1909 -> Musikwerknummer)
- ✓ Seitenbeschriftung: Titel, Duofassung, Ensemblefassung.
- ✓ Stimmen ziehen
 - Zuerst die Notengröße einstellen! (Prozentwerkzeug, Größe der Seite und Größe des Notensystems auf 100%).
Größe der Akkolade (Akkolade anklicken, Seitenlayout-Werkzeug: Menü Seitenlayout: Größe der Akkolade ändern: Notensystemhöhe: 0.28472 Zoll, Akkolade skalieren 100%, resultierende Akkoladenskalierung: 85%).
 - Die Stimmen anschreiben: Auf der 1. Seite links oben (24 Punkt, fett) und auf jeder Seite Mitte oben.
 - Die transponierenden Stimmen transponieren.
Um bei der transponierenden Notation statt der Tonartenvorzeichen die Vorzeichen vor jedem betreffenden Ton zu haben: Rufen Sie die Partiturverwaltung auf. Im Einblendmenü "Transposition" finden Sie die Option "Andere", die Sie auswählen. Daraufhin öffnet sich ein Fenster. Dort wählen Sie „Chromatisch“.
 - Bei den transponierenden Instrumenten die richtigen Schlüssel einstellen. (Klarinette hat nur den Violinschlüssel.)
 - Bei den transponierenden Instrumenten die Oktavlagen kontrollieren.
 - Bei den transponierenden Instrumenten: „Transponierend notiert“.
 - Die Seitenwende-Stellen einrichten. Gemacht.
 - Teile-Bezeichnungen (Teil 1-7) in die Stimmen
 - Taktgruppen fixieren und Sicherheits-Pdf erstellen.
 - "Sektion" einsetzen. Gemacht.
 - Layout-Abstände zwischen den Systemen.

- Stichnoten nach längeren Pausen setzen. (Bei transponierenden Instrumenten die Stichnoten auf klingend transponieren.)
- Nach mehrtaktigen Pausen die Taktzahl zeigen.
- G.P. (nicht tacet) nur in allen Stimmen: angeben. (18 fett)
- Tempo-Vorankündigung am Ende des Systems (ohne hinterlegte Tempoänderung), wenn auf dem nächsten System das Tempo wechselt.
- Am Ende (oder unten / oben auf) der 1. Seite:
Anmerkung: Vorzeichen gelten jeweils nur für eine einzige Note in der entsprechenden Oktavlage. Unmittelbare Tonwiederholungen (auch durch Pausen getrennte) behalten die gleiche Tonhöhe bei. Töne ohne Vorzeichen gelten immer als nicht alteriert.
- Durch die Transposition verursacht:
 - Bei der Klarinetten-Einzelstimme keine eis, his etc.: T.3,
 - Nochmals Vorzeichenkontrolle: Kommt ein alterierter Ton im Takt nochmals vor, alteriert oder unalteriert, dann müssen Versetzungszeichen gesetzt werden.
 - Bei der transponierenden Klarinettenstimme die überflüssigen Auflösungszeichen löschen. Kontrolliert: ...
- Grob editieren.
- Die Stimmen durchgehen und u.a. überflüssige Vorzeichen (in Klammern) löschen. Gemacht:
- ✓ Systemtrennstriche setzen: Gemacht.
 - ✓ Die Stimmen ausdrucken und mit der Partitur vergleichen. Ev. noch bearbeiten. (Tempowechsel, Attacca). (~~Am Bildschirm gemacht.~~) Der korrigierte Ausdruck ergibt die Kopiervorlage.
- ✓ **Das Werk verzeichnen:**
Homepage:
 - ✓ Chronologisch,
 - ✓ nach Besetzungen (ev. Duofassung, wenn Bariton und Klavier von der gleichen Person gespielt werden / Triofassung, wenn Bariton und Klavier nicht von der gleichen Person gespielt werden),
 - ✓ Duos
 - Für Stimme und Klavier
 - Mit Stimme
 - ✓ nach Instrumenten,
 - (Zyklen)
 Werkliste (Word): ✓ Chronologisch, ✓ nach Besetzungen, ✓ Duos, Für Stimme und Klavier, Mit Stimme, (Zyklen)
- ✓ Den Werkkommentar schreiben,
 - in ein eigenes Word-Dokument setzen,
 - auf die Homepage setzen und dort mit dem Inhaltsverzeichnis (der Werkkommentare) und mit der Werkliste verlinken.
- ✓ Titelblatt, Innenblatt, Vorwort, Biographie herstellen.
- ✓ Titelblatt, Innenblatt, Vorwort, Werkkommentar und Bio für die Kopiervorlage ausdrucken.
- ✓ Gesamt-Pdf-Fassung und Druckfassung herstellen.
- ✓ Den Entwicklungsgang / History herstellen.
 - Im Titel jeder Fassung die jeweilige Fassung angeben („Titel-3“) und in den Seitenbeschriftungen die verschiedenen Fassungen durchnummerieren.
 - Alle Fassungen grob editieren.
 - linker Akkoladenrand 0.1:
 - Dynamik:
- • Alle Fassungen in das gleiche Dokument setzen.
- Kontrollieren, ob die ungeraden Partiturseiten mit den ungeraden Pdf-Seiten übereinstimmen, falls

dies wichtig ist.

- Das Inhaltsverzeichnis des Entwicklungsgangs erstellen, siehe „Miramsobale“.
- ✓ Auf der Homepage "Downloads" einrichten (alle Angaben aus der Werkliste kopieren und damit einen neuen Artikel auf "Downloads" einrichten).
- Die Gesamt-Pdf-Fassung, die Druckfassung, die Einzelstimmen, den Entwicklungsgang, die Skizzen ggf. den Vortrag auf die Homepage (Downloads) setzen, und mit dem Inhaltsverzeichnis und der Werkliste verlinken.

René Wohlhauser:

Partitur **mira schinak** Score pdf download, sheet music (-> überall)

- Modell: Kasamarówa:
- René Wohlhauser Partitur **Kasamarówa** score pdf download, sheet music
 - Kasamarówa-Partitur
 - Kasamarówa-Druckvorlage
- Analyse von Kasamarówa:
 - Kasamarówa-Analyse
- Entwicklungsgang / History:
 - Kasamarówa-Entwicklungsgang/History
- Kasamarówa-Skizzen / Sketches:
 - Kasamarówa-Skizzen/sketches
- Einzelstimmen / Parts
- Kompositionsprotokoll / Composition report:
 - Kasamarówa- Kompositionsprotokoll
- (✓) Das Gedicht mit der Fassung in der Gedichtsammlung vergleichen.
- ✓ Das Werk im Excel-Dokument „Werke-Standorte.xlsx“ verzeichnen.
- Anmelden bei
 - ✓ GEMA. Gemacht am 04.8.2022.
 - AdS-Lexikon: Gemacht am
 - Pro Litteris: Gemacht am
 - ✓ SME: Gemacht am 4.8.2022.
 - ✓ Die Anmeldungen im Excel-Dokument „Werke-Standorte.xlsx“ (mit Kommentar!) verzeichnen.
- ✓ In das Dokument „Partitur-Nachbereitung“ den Titel mit dem Vermerk setzen: „Nach der UA und der Studioaufnahme aktualisieren und auf der Homepage und bei der SME ersetzen.“
-
-
- Das Kompositionsprotokoll durchlesen und auf die Homepage (Downloads) setzen. Gemacht am 5.8.2022.

Konzept / Werkkommentar

Werkkommentar:

Es wird zwar „nur“ die chromatische Tonleiter verwendet. Was aber mit dieser chromatischen Tonleiter gemacht wird, ist entscheidend. Hier ist eine große Phantasie innerhalb enger Grenzen gefragt. Dies ist größte und schönste Herausforderung für einen Komponisten.

Manchmal ist eine Reduktion des Materials notwendig, um zur Essenz der Aussage vordringen zu können. Dies wußte schon Beethoven, z.B. mit seiner 5. Symphonie.

Die Reduktion des Tonmaterials und die Synchronität des Rhythmus lenken das Ohr auf die anderen Parameter-Prozesse: Phrasendauern, Phrasenentwicklung, Phrasentendenzen, Phrasenkombinationen.

Geringste Abweichungen und ihre Folgen werden wahrgenommen. Sehr gut nachzuvollziehen beispielsweise in der Quasi-Cellokadenz (Takte 189-228). (16.7.2022)

Schroffe, Prägnante und zupackende Rhythmizität prägt den Charakter dieser Komposition. Die strenge Reduktion der Melodik auf Halbton-Fortschreitungen führt dazu, daß es keine prägnanten melodischen Wendungen gibt, die von der rigide durchgehaltenen Grundkonzeption ablenken und daß daher auch dieser lineare Parameter die rhythmische Ausrichtung der Komposition unterstützt.

Manchmal ist eine Reduktion des Materials notwendig, um zur Essenz der Aussage vordringen zu können. Die Reduktion des Tonmaterials und die Synchronität des Rhythmus lenken das Ohr auf die anderen Parameter-Prozesse: Phrasendauern, Phrasenentwicklung, Phrasentendenzen, Phrasenkombinationen. Geringste Abweichungen und ihre Folgen werden wahrgenommen. Eine möglichst reiche Entwicklung der Phantasie innerhalb eng abgesteckter Grenzen stellt seit jeher eine der größten und reizvollsten Herausforderungen an Komponierende dar.

Die ganze Komposition läßt sich sozusagen auf wenige konstitutive Elemente zurückführen, die aber in immer wieder anderer Weise variiert, kombiniert und von verschiedenen Seiten her beleuchtet werden. Dies schafft eine starke Homogenität und Geschlossenheit in der Wahrnehmung, eine Objektivierung im Ausdruck, eine differenzierte Vielfalt im Kleinen und eine Einheit des Denkens im Großen. (2.8.2022, 2:21 Uhr)

Vorwort

Text

Technik

Fassungen:

Chronologisch:

- **1. Fassung** des Hauptdokuments **Klaviertrio Nr. 3-1 (2022).musx:**
 - **2. Fassung** des Hauptdokuments **Klaviertrio Nr. 3-2 (2022).musx:** Den Takt 72 ändern, nach Takt 76 vier Takte einschieben, um die Idee von Takt 77 weiter auszuarbeiten. Einen Takt später wieder zwei Takte einschieben, um die dortige Idee mehr auszuarbeiten (9.7.2022).
 - **3. Fassung** des Hauptdokuments **Klaviertrio Nr. 3-3 (2022).musx:** Nach Takt 26 einen weiteren Wartetakt einschieben (14.7.2022).
 - **4. Fassung** des Hauptdokuments **Klaviertrio Nr. 3-4 (2022).musx:** In den Teilen 1-11 bzw. in den Takten 1-279 die Zusammenklänge / Akkordstrukturen bearbeiten (16.7.2022).
-

Kritik / Fragen:

Titel: