

Trio für Sopran, Violine und Klavier Nr. 2

Ergon 87, Musikwerknummer 2146 (2022)

Auf 5 Gedichte des Komponisten

Kompositionsprotokoll

Durchgesehen am 21.3.2022.

Inhalt

• Ausdruck	2
• Format	2
• Klang / Aufnahme	2
• Dynamik	2
• Daten / Tagesprotokoll	2
• Arbeiten	6
• Die Programmcodes	7
• Programmcode für Teil 2b	7
• Programmcode für Teil 4a	9
• Programmcode für Teil 4b	10
• Programmcode für Teil 6a	12
• Programmcode für Teil 6b	13
• Programmcode für Teil 8a	15
• Programmcode für Teil 8b (und c)	17
• Vorgehen	18
• Brainstorming / Planung / Texte	18
• Form	19
• Kontrollen	20
• Konzept / Werkkommentar	23
• Vorwort	24
• Text	24
• Technik	24
• Fassungen	24
• Kritik / Fragen	24
• Titel	24

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:

- 27.02.2022, Sonntag, 10.30-11.00 Uhr: Die erste Idee definieren:
 - Gruppen von 3, 5 oder 7 Sechzehntelnoten, aleatorisch.
 - Pro Gruppe in kleinen Sekunden entweder nur steigend oder nur fallend, aleatorisch.
 - Zwischen den Gruppen entweder eine Achtelpause oder eine punktierte Achtelpause, aleatorisch.
 - Zwischen den Gruppen entweder kleine Sekunde, eine Quarten oder ein Tritonus, aleatorisch.

Das Protokoll einrichten.
Das Finale-Dokument einrichten.
- 28.02.2022, Montag, 10.30-11.50 Uhr: Die Takte 1-13 komponieren (wie immer direkt ins Finale-Notationsprogramm).
16.40-16.55 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“). Dokument: Dokument-Optionen: Musikausrichtung: Manuelle Positionierung: einarbeiten. X-tolen-Werkzeug: Horizontales Ziehen erlauben, immer horizontal, Klammer über die gesamte Dauer.)
- 01.03.2022, Dienstag, 10.15-11.45 Uhr: Editieren. Ich speicherte das Dokument als **2. Fassung** unter **Trio für Sop, Vn u. Pfte Nr. 2-2.musx** ab und ließ im 2. Takt den letzten Klang in der linken Hand einen Takt lang liegen. Die Geige in Takt 5 komponieren und in Takt 6 ändern. Den Sopran ab Takt 9 und ab Takt 14 alle Stimmen bis Takt 19 weiterführen.
14.00-14.40 Uhr: Bis Takt 31 weiterkomponieren.
- 02.03.2022, Mittwoch, 8.20-8.55 / 9.36-10.50 Uhr: Die Definition vom 27.02.2022 präzisieren und ausführen.
 - 3 Sechzehntel & Achtelpause, 5 Sechzehntel & punktierte Achtelpause, 7 Sechzehntel & Viertelpause. Ergibt 9 Takte.

- Reihenfolge: 3-5-3-5-7-5-7-7-5-7-5-3 (zunehmend-abnehmend) für Violine, Klavier im Krebs
 - Aufwärts-abwärts: /-/-\--\--\--\--\--\--\ für Violine, Klavier im Krebs
 - Register: Violine in der zwei- und dreigestrichenen Oktave, Klavier in der kleinen und eingestrichenen Oktave
 - Vorgehen: In der Violine nur aufwärts vom a2: 3-5 schreiben & copy-paste, 7 schreiben, Rest copy-paste.
Gemacht in I. von „Trio SopVnKlav-Hilfsdatei 2. Teil 2a.musx“.
In II. die Spiegelungen machen, damit die Aufwärts-Abwärts-Reihenfolge stimmt.
In III. die Singstimme komponieren. Die Instrumentalstimmen durch Einschoben von halben Pausen (jeweils nach 2-3-2-2 usw. Figuren) und durch die Wiederholung des Schlußteils so weit auseinanderlegen, daß die Singstimme gut atmen kann und Singstimme sowie Instrumente gleichzeitig aufhören.
 - Der Teil III. der Hilfsdatei könnte zum 2. Teil des Stückes ab Takt 33 werden.
 - Editieren.
- 18.30-19.10 Uhr: Zusammen mit Stéphane Boussuge zwei Opusmodus-Versionen der Definitionen / Vorgaben für den 2. Teil des Stückes programmieren.
- 03.03.2022, Donnerstag, 10.00-11.30 Uhr: In Opusmodus verschiedene Versionen für den 2. Teil des Stückes programmieren. Schließlich entschied ich mich für die 4. Version. Ich exportierte diese Version als **Trio SopVnKlav Teil 2b OM.xml** aus Opusmodus und importierte sie al **Trio SopVnKlav Teil 2b OM.musx** ins Finale. Grob editieren.
Die Definition / Vorgaben sind leicht abgewandelt:
 - 3 Sechzehntel & Achtelpause, 5 Sechzehntel & punktierte Achtelpause, 7 Sechzehntel & Viertelpause. Stets kleine Sekunden pro Gruppe entweder steigend oder fallend. Zwischen den Gruppen auch kleine Sekunden, was aber durch Hinzufügung einer Umfangsbeschränkung teilweise verfälscht wird.
 - Sopran: Viertel, punktierte Viertel, Halbe, punktierte Halbe, dazwischen ab und zu Achtelpausen. Intervalle: Tritonus, Quarte, kleine Sekunde.
 - 04.03.2022, Freitag, 10.40-11.05 Uhr: Die Opusmodus-Version des 2. Teils in die Gesamtpartitur einfügen: ergibt den Teil 2b bzw. die Takte 51-76 (alt: 48-73). Dadurch wird der 2. Teil zweiteilig: der Teil 2a bzw. die Takte 35-50 (alt: 35-48) entstand aus der manuell komponierten Version, der Teil 2b bzw. die Takte 51-76 (alt: 50-75) entstand aus der Opusmodus-Version. Die Definition für die Opusmodus-Version in den gestrigen Protokoll-Eintrag einfügen.
17.08-18.15 Uhr: Die eingefügte Opusmodus-Version Teil 2b editieren.
 - 07.03.2022, Montag, 11.30-12.45 Uhr: Das Bisherige durchhören. In Takt 49 (alt: 47) den Schluß von Takt 48 (alt: 46) wiederholen, um den Teil 2a besser abzuschließen. In den Takten 16-32 kleine Veränderungen und Ergänzungen vornehmen und den Teil bis Takt 34 erweitern. Dann komponierte ich den Teil, der die Takte 77-104 (alt: 76-103) beinhaltet (Teil 3).
13.30-13.50 Uhr: Editieren (ab T. 33 die Legatobögen).
 - 08.03.2022, Dienstag, 10.50-12.30 / 12.50-13.05 /13.30-14.00 Uhr: Das Bisherige durchhören. In Opusmodus die 5. Version programmieren, die den Teil 4a ergibt. Diese Version als **Trio SopVnKlav Teil 4a OM.xml** exportieren und als **Trio SopVnKlav Teil 4a OM.musx** ins Finale importieren, grob editieren und ins Hauptdokument kopieren. Die Programmcodes für die Teile 2b und 4a ins Protokoll kopieren.
In Opusmodus die 6. Version programmieren, die den Teil 4b ergibt. Diese Version als **Trio SopVnKlav Teil 4b OM.xml** exportieren und als **Trio SopVnKlav Teil 4b OM.musx** ins Finale importieren, grob editieren und ins Hauptdokument kopieren. Den Programmcode für den Teil 4b ins Protokoll kopieren.
Editieren.
 - 09.03.2022, Mittwoch, 11.35-12.50 Uhr: Das Bisherige durchhören. Kleinigkeiten ergänzen, editieren. Die Dynamik gestalten. Ab Takt 136 (alt: 135) weiterkomponieren, indem der Teil 1 auseinander gedehnt wird (Takte 136-159, alt: 135-158 bzw. Teil 5).

13.10-13.20 / 13.50-14.05 Uhr: In Opusmodus die 7. Version für den Teil 6a programmieren. Diese Version als **Trio SopVnKlav Teil 6a OM.xml** exportieren.

18.00-18.35 Uhr: In Opusmodus die 8. Version für den Teil 6b programmieren. Verschiedene Versionen ausprobieren, bis ich das bekam, das ich suchte. Diese Version als **Trio SopVnKlav Teil 6b OM.xml** exportieren.

- 10.03.2022, Donnerstag, 12.00-12.30 Uhr: Das **Trio SopVnKlav Teil 6a OM.xml** als **Trio SopVnKlav Teil 6a OM.musx** ins Finale importieren. Das **Trio SopVnKlav Teil 6b OM.xml** als **Trio SopVnKlav Teil 6b OM.musx** ins Finale importieren. Von den Opusmodus-Versionen 4, 5, 6, 7 und 8 bzw. den Teilen 2b, 4a, 4b, 6a und 6b direkt aus Opusmodus ein Pdf herstellen, um allfällige Abweichungen zu erkennen bzw. zu verdeutlichen.

13.45-14.00 Uhr: **Trio SopVnKlav Teil 6a OM.musx** editieren.

15:35-16:30 Uhr: Die Teile **Trio SopVnKlav Teil 6a OM.musx** und **Trio SopVnKlav Teil 6b OM.musx** editieren und in das Hauptdokument **Trio für Sop, Vn u. Pfte Nr. 2-2.musx** kopieren. Unter „Form“ die Form-Übersicht nachtragen.

- 11.03.2022, Freitag, 17.55-18.22 Uhr: Editieren.
- 13.03.2022, Sonntag, 11.10-12.40 / 13.00-13.10 Uhr: Die Programmcodes in Opusmodus mit Erläuterungen ergänzen und erneut in das Protokoll kopieren.

Am Keyboard die Takte 1-103 kontrollierend durchspielen. In den Takten 35-36, 42-43 und 47-48 das Unisono Violine-Klavier in eine Spiegelung umwandeln. Die Unisoni in den Takten 54, 70-71, 71, 74 und 160 (alt: 53, 69-70, 70, 73 und 159) sind nur kurz und können stehen bleiben. Zudem soll das Halbton-Connect-System zwischen den Tongruppen nicht gestört werden. Halbton-Connect-System: Nicht nur innerhalb der Tongruppen von Violine und Klavier gibt es nur ausschließlich Halbtöne, sondern auch von Tongruppe zur nächsten Tongruppe (in Opusmodus: bind-to-interval).

Oktaven dürfen vorkommen (z.B. 2 Oktaven in Takt 64, weitere Oktaven in Takt 65 usw. (alt: 63, in Takt 64 usw.).

In Takt 76 (alt: 75) hat das System in der Klavierstimme zweimal einen Fehler produziert, indem statt durchgehender Halbton-Bewegung zweimal eine Terz erklingt. Aber dadurch, daß es zweimal erscheint (zudem in Umkehrung und einmal als große und einmal als kleine Terz, also in Analogie-Bildungen) wirkt es plausibel. (Also lassen.)

Form-Angaben unter „Form“ verzeichnen.

16.00-16.50 / 18.40-18.55 Uhr:

Etwas unschön ist die (im Klavier im Sprung erreichte) verdeckte Oktave zum „fis“ in Takt 111 (alt: 110) auf den 8. Sechzehntel (Violine f1-fis1, Klavier G-fis). Ich korrigierte es, indem ich das „fis“ am Ende des 2. Viertels im Klavier oktavierte (und machte damit eine Ausnahme von der Ambitus-Beschränkung in diesem Teil, indem das Klavier nicht unter das große G gehen sollte).

Am Keyboard die Takte 104-209 kontrollierend durchspielen.

In der Partitur die Teile anschreiben („Teil 1“, „Teil 2a“ usw.).

- 14.03.2022, Montag, 11.40-12.45 Uhr: Das Bisherige durchhören. Das Dokument als **3. Fassung** unter **Trio für Sop, Vn u. Pfte Nr. 2-3.musx** abspeichern und nach Takt 49 einen Pausentakt einschieben. Dadurch verschieben sich die folgenden Taktnummern um eine Zahl nach oben. (Im Protokoll korrigiert, indem nun die neuen Taktzahlen stehen und die bisherigen unter „alt:“ in Klammern beigefügt sind.)

Aus Schnipseln der Takte 6-13 den Teil 7 (Takte 212-228) zusammensetzen (komponieren).

Die Opusmodus-Codes der Versionen 9 und 10 für die Teile 8a und 8b programmieren.

13.45-15.25 Uhr: Diese Versionen 9 und 10 einerseits als **Trio SopVnKlav Teil 8a, Pdf aus OM.pdf** und **Trio SopVnKlav Teile 8a&8b, Pdf aus OM.pdf** und andererseits als **Trio SopVnKlav Teil 8a OM.xml** und **Trio SopVnKlav Teile 8a&8b OM.xml** aus Opusmodus exportieren und nur die Version 10 als **Trio SopVnKlav Teile 8a&8b OM.musx** ins Finale importieren und in der Violine die Register kontrollieren und editieren. (Manchmal verschieben sich beim Import die Violintöne um eine Oktave nach unten.) Dann kopierte ich die Takte 1-86 aus **Trio SopVnKlav Teile 8a,8b,8c**

OM.musx ins Hauptdokument **Trio für Sop, Vn u. Pfte Nr. 2-3.musx**. Dies ergab dort die Takte 229-318 (alt: 229-314) bzw. die Teile 8a-8c. Durch das Kopieren wurde die Violinstimme seltsamerweise um 2 Oktaven nach unten versetzt, so daß ich dies korrigieren und nochmals den ganzen Teil kontrollieren mußte.

Von den Takten 1-86 von **Trio SopVnKlav Teile 8a,8b,8c OM.musx** (bzw. von den Takten 229-314 des Hauptdokuments **Trio für Sop, Vn u. Pfte Nr. 2-3.musx**) wurden die Takte 1-42 von **Trio SopVnKlav Teile 8a,8b,8c OM.musx** (bzw. die Takte 229-270 des Hauptdokuments **Trio für Sop, Vn u. Pfte Nr. 2-3.musx**) zum Teil 8a. Dann fügte ich zwei Pausentakte ein. Die folgenden Takte 273-320 (alt: 273-317) des Hauptdokuments wurden zum Teil 8b (der vielleicht nochmals unterteilt werden wird). In den 9 Takten 34-42 von **Trio SopVnKlav Teile 8a,8b,8c OM.musx** (bzw. in den Takten 262-270 des Hauptdokuments **Trio für Sop, Vn u. Pfte Nr. 2-3.musx**) entfernte ich die Violinstimme aus der Partitur. In Spiegelung dazu ebenso in den 9 Takten 43-51 von **Trio SopVnKlav Teile 8a,8b,8c OM.musx** (bzw. in den Takten 273-281 des Hauptdokuments **Trio für Sop, Vn u. Pfte Nr. 2-3.musx**).

In den 11 Takten 32-42 von **Trio SopVnKlav Teile 8a,8b,8c OM.musx** (bzw. in den Takten 260-270 des Hauptdokuments **Trio für Sop, Vn u. Pfte Nr. 2-3.musx**) entfernte ich die Sopranstimme aus der Partitur. In Spiegelung dazu ebenso in den 11 Takten 43-53 von **Trio SopVnKlav Teile 8a,8b,8c OM.musx** (bzw. in den Takte 273-283 des Hauptdokuments **Trio für Sop, Vn u. Pfte Nr. 2-3.musx**), in denen der Sopran sowieso über weite Strecken pausierte.

Dann teilte ich den Teil 8b nochmals auf. Die 44 Takte 273-316 (ohne die beiden Pausentakte 300 und 301) im Goldenen Schnitt aufgeteilt, ergeben 27+17 Takte. Folglich werden die Takte 273-299 zum Teil 8b, Takte 300 und 301 sind Pause, und die Takte 302-318 werden zum Teil 8c. Dann werden wieder 2 Pausentakte eingefügt.

Durch das Einfügen der Pausentakte verschob sich die Taktnumerierung. Dies ist im Protokoll korrigiert. Dies bisherigen Taktzahlen sind mit „alt:“ angefügt.

- 15.03.2022, Dienstag, 12.20-13.05 Uhr: Das Bisherige durchhören. Ideen zum Werkkommentar notieren. Aus dem Material der Takte 13-16 die Takte 321-334 komponieren.
13.55-14.25 Uhr: Nach dem Schnipsel- und Dehnungsverfahren aus dem Material der Takte 17-33 die Takte 335-371 komponieren (plus ein Pausentakt). Damit ist **das Ende der Musikkomposition** erreicht. Nun folgt das Textieren, Editieren und Kontrollieren.
- 16.03.2022, Mittwoch, 18.00-18.20 Uhr: Ich begann mich mit der Textierung zu befassen. Benötigte Silben:
 - Teil 1, Takte 1-34: 17 Töne.
 - Teil 2a, Takte 35-50: 23 Töne (ev. repetierend)
 - Teil 2b, Takte 51-76: 67 Töne
 - Teil 3, Takte 77-104: 5 Töne
 - Teil 4a, Takte 105-115: 12 Töne
 - Teil 4b, Takte 116-135: 16 Töne
 - Teil 5, Takte 136-159: 3 Töne
 - Teil 6a, Takte 160-175: 19 Töne
 - Teil 6b, Takte 176-211: 31 Töne
 - Teil 7, Takte 212-228: 5 Töne
 - Teil 8a, Takte 229-272: 26 Töne
 - Teil 8b, Takte 273-301: 13 Töne
 - Teil 8c, Takte 302-320: 7 Töne
 - Teil 9, Takte 321-372: 12 Töne

Ev. in den geraden Teilen algorithmisch erstellte Kombinationen gegebener Silben bzw. 12 einsilbiger Wörter:

21.30-21.45 Uhr: Holz Stolz Wurf Schlurf Wurm Turm kalt Spalt macht lacht Schund Schwund

23.15-23.30 Uhr: Test mit Rnd-Sample.

Resultat 1 (seed 23): ((turm) (schlurf) (schlurf) (schwund) (schlurf) (turm) (holz) (holz) (kalt) (turm) (schund) (turm) (wurf) (stolz) (holz) (wurm) (schund) (schwund) (schwund) (wurm) (holz) (wurm) (wurf))

Resultat 2 (seed 24): ((spalt) (turm) (turm) (schund) (kalt) (schwund) (turm) (wurm) (wurf) (lacht) (schund) (lacht) (lacht) (kalt) (stolz) (schwund) (schwund) (turm) (wurf) (kalt) (schwund) (stolz) (wurf))

Die Resultate finde ich nicht zufriedenstellend.

23.30-23.55 Uhr: Das Gedicht "Aberlein quanimog" für die Teile 2a und 2b schreiben bzw. erfinden.

23.55-00.05 Uhr: Das Gedicht "Malfroni" für die Teile 4a und 4b schreiben.

00.05-00.19 Uhr: Das Gedicht "Siffaz" für die Teile 6a und 6b schreiben.

00.19-00.30 Uhr: Das Gedicht "Drangu Sehu" für die ungeraden Teile geschrieben.

00.30-00.45 Uhr: Die Gedichte verzeichnen.

00.45-00.55 Uhr: Das Gedicht "Drangu Sehu" in die Partitur einsetzen.

- 17.03.2022, Donnerstag, 3.55-5.25 Uhr: Vokalttexte einsetzen bis und mit Teil 7.

5.25-6.13 Uhr: Das Gedicht "Kurz-Holz" geschrieben, 2 Fassungen.

6.13-6.30 Uhr: Die restlichen Vokalttexte einsetzen. Damit ist auch das **Ende der Vokalttextkomposition** erreicht.

6.30-6.40 Uhr: Die Musikwerknummer zuordnen und das Werk im Werkverzeichnis verzeichnen.

18.10-19.10 Uhr: Die Teile 8a-c editieren.

21.15-21.45 Uhr: Die Takte 210-371 am Keyboard durchspielen.

23.00-00.45 Uhr: Details editieren. Die Vokalttexte kontrollieren und ausrichten. Unter "Form" die Angaben ergänzen und erläutern. Die Codes der Versionen 9 und 10 ins Protokoll kopieren.

- 18.03.2022, Freitag, 9.50-11.50 / 14.00-15.55 Uhr: Kontrollen. Die Violinstimme ziehen und editieren. Systemtrennstriche setzen. Das Werk auf der Homepage und in der Word-Werkliste verzeichnen.

18.35-19.10 Uhr: Den Werkkommentar schreiben.

21.00-21.15 / 23.00-1.45 Uhr: Titelblatt, Innenblatt, Vorwort, Biographie herstellen. Titelblatt, Innenblatt, Vorwort, Werkkommentar und Bio für die Kopiervorlage ausdrucken. Gesamt-Pdf-Fassung und Druckfassung herstellen. Die Dateien für den Entwicklungsgang editieren, zusammensetzen und das Inhaltsverzeichnis dazu herstellen.

- 19.03.2022, Samstag, 11.00-12.45 Uhr: Die Partitur editieren, vorallem die Handverteilung im Klavier.

14.15-15.00 Uhr: Das Stück durchhören. Editieren.

16.45-18.00 18.20-19.05 Uhr: Auf der Homepage zu diesem Stück den Artikel in "Downloads" einrichten und die Dateien hochladen. Werkanmeldungen bei der GEMA und im AdS-Lexikon (Autoren der Schweiz).

- 20.03.2022, Sonntag, 18.45-19.10 Uhr: Gedichte-Anmeldungen bei der Pro Litteris.

21.45-22.30 Uhr: Das Werk bei der SME (Schweizer Musikedition) anmelden. Die Anmeldung auf dem „SME-Infos“-Dokument unter den verschiedenen Kategorien verzeichnen. Auf der Homepage Verlinkungen machen.

- 21.03.2022, Montag, 11.00-12.40 / 14.00-15.15 Uhr: Das Protokoll durchlesen und ergänzen. Alle Taktzahlen in den Partitur-Dateien kontrollieren und korrigieren, was sich nachträglich verschoben hat. Das Protokoll auf die Homepage setzen.

Arbeiten:

- ✓ Die Opusmodus-Version des 2. Teils (Takte 48-73) bearbeiten, um die durch die Umfangbeschränkung erfolgten Verfälschungen zu korrigieren. Am 4.3.2022 gemacht.
- ✓ Ab T. 33 Legatobögen. Gemacht am 7.3.2022.
- ✓ Die Teile anschreiben. Gemacht am 13.3.2022.
- ✓ Vokalttexte einsetzen. Gemacht am 16. und 17.3.2022
- ✓ Vokalttexte kontrollieren und ausrichten. Gemacht am 17.3.2022
- ✓ Die Form ergänzen und erläutern. Gemacht am 17.3.2022

- ✓ Die Codes der Versionen 9 und 10 ins Protokoll kopieren. Gemacht am 17.3.2022
- ✓ S. 1 u. 2 ausdr. und mit der Schnipsel-Rekomposition der Teile 5 und 7 vergleichen. Gemacht am 17.3.2022
- ✓ Teile 8a-c: Legatobögen, Klavier ins richtige System. Pausen zusammenfassen. Gemacht am 17.3.2022.
- ✓ Textieren. Gemacht am 16. und 17.3.2022.
- ✓ T. 332 Handverteilung. 17.3.2022.
- ✓ S. 34 editieren. Gemacht am 17.3.2022.
- ✓ Am Keyboard durchspielen. Am 13.3.2022 bis Takt 209 gemacht. Am 17.2.2022 von 210-371 gemacht.
- Während dem Üben:
- Koordinationspfeile in den B-Teilen. Ev. nur in der Spielpartitur.
- ✓ Handverteilung in den B-Teilen. Gemacht am 19.3.2022.
-
- **ArbeitenAktuellerPunkt [ar]**

Die Programmcodes (Opusmodus):

Trio SopVnKlav2022.opmo

; 4. Version -> für **Teil 2b**

; 3.3.2022

; (Wie 2. Version, aber mit anderen Intervallen nur zwischen den Sopran-Gruppen, sonst stets kleine Sekunden. Mit anderem Ambitus.)

(progn

(init-seed 23)

(setf Rhy-Vn (rnd-sample 50 '((s s s -e) (s s s s s -e.) (s s s s s s -q))))

(setf Rhy-Sop (rnd-sample 70 '((q) (q.) (h) (h.) (-e))))

(setf Rhy-Klav (rnd-sample 52 '((s s s -e) (s s s s s -e.) (s s s s s s -q))))

(setf count-Vn (mclength Rhy-Vn))

(setf count-Sop (mclength Rhy-Sop))

(setf count-Klav (mclength Rhy-Klav))

; mclength zählt die Anzahl Ereignisse einer definierten Variablen, z.B. von Rhy-Vn (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-Sop (rnd-sample 50 '(q q. h h. -e))), sondern

; (setf Rhy-Sop (rnd-sample 50 '((q) (q.) (h) (h.) (-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-Vn (mclist (gen-weight 50 '((-1 9)(1 9)))))

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

(setf Int-Klav (gen-repeat count-Klav (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 12 (interval-to-pitch Int-Vn)))
(setf Pitch-Sop (pitch-transpose 12 (interval-to-pitch Int-Sop)))
(setf Pitch-Klav (interval-to-pitch Int-Klav))
  ; 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 Vn-Part (make-omn
  :pitch Pitch-Vn
  :length Rhy-Vn
))
(setf Sop-Part (make-omn
  :pitch Pitch-Sop
  :length Rhy-Sop
))
(setf Klav-Part (make-omn
  :pitch Pitch-Klav
  :length Rhy-Klav
))
  ; Tonhöhen und Dauern werden pro Stimme zusammengefügt.
```

```
(setf list-int1 (rnd-sample 10 '(1 -1)))
(setf list-int2 (rnd-sample 10 '(6 -6 5 -5 1 -1)))
(setf list-int3 (rnd-sample 10 '(1 -1)))
  ; Eine Liste von Intervallen, die zwischen den Gruppen zur Anwendung kommen sollen.
```

```
(setf violin (ambitus '(c5 fs6)(bind-to-interval list-int1 Vn-Part)))
(setf soprano (ambitus '(fs4 c6)(bind-to-interval list-int2 Sop-Part)))
(setf piano (ambitus '(c2 c5) (bind-to-interval list-int3 Klav-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) :soprano-voice (list soprano) :pg (list piano) :tempo 138 :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
; [47"]
```

--

; 5. Version -> für **Teil 4a**
 ; 8.3.2022

```
(progn
  (init-seed 27)
  (setf Rhy-Vn (rnd-sample 17 '((s s s -e) (s s s s s -e.) (s s s s s s s -q))))
  (setf Rhy-Sop (rnd-sample 25 '((q) (q.) (h) (h.) (-e) (-q) (-q.))))
  (setf Rhy-Klav (rnd-sample 21 '((s s s -e) (s s s s s -e.) (s s s s s s s -q))))
```

```
(setf count-Vn (mclength Rhy-Vn))
(setf count-Sop (mclength Rhy-Sop))
(setf count-Klav (mclength Rhy-Klav))
```

; mclength zählt die Anzahl Ereignisse einer definierten Variablen, z.B. von Rhy-Vn (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-Sop (rnd-sample 50 '(q q. h h. -e))), sondern
 ; (setf Rhy-Sop (rnd-sample 50 '((q) (q.) (h) (h.) (-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-Vn (mclist (gen-weight 50 '((-1 9)(1 9)))))
(setf Int-Sop (gen-repeat count-Sop (mclist (gen-weight 50 '((-1 9)(1 9)))))
(setf Int-Klav (gen-repeat count-Klav (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 12 (interval-to-pitch Int-Vn)))
; 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-Vn
))
(setf Sop-Part (make-omn
  :pitch Pitch-Sop
```

```

        :length Rhy-Sop
    ))
    (setf Klav-Part (make-omn
        :pitch Pitch-Klav
        :length Rhy-Klav
    ))
    ; Tonhöhen und Dauern werden pro Stimme zusammengefügt.

```

```

(setf list-int1 (rnd-sample 10 '(1 -1)))
(setf list-int2 (rnd-sample 10 '(6 -6 5 -5 1 -1)))
(setf list-int3 (rnd-sample 10 '(1 -1)))
    ; Eine Liste von Intervallen, die zwischen den Gruppen zur Anwendung kommen sollen.

```

```

(setf violin (ambitus '(g3 fs4)(bind-to-interval list-int1 Vn-Part)))
(setf soprano (ambitus '(fs4 c6)(bind-to-interval list-int2 Sop-Part)))
(setf piano (ambitus '(c2 fs3) (bind-to-interval list-int3 Klav-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) :soprano-voice (list soprano) :pg (list piano) :tempo 138 :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
    ; [17"]

```

--

```

; 6. Version -> für Teil 4b
; 8.3.2022

```

```

(progn
    (init-seed 29)
    (setf Rhy-Vn (rnd-sample 17 '((s s s -h) (s s s s -h.) (s s s s s -h..))))
    (setf Rhy-Sop (rnd-sample 28 '((q) (q.) (h) (h.) (-e) (-q) (-q..)))
    (setf Rhy-Klav (rnd-sample 19 '((s s s -h) (s s s s -h.) (s s s s s -h..))))

```

```

(setf count-Vn (mclength Rhy-Vn))
(setf count-Sop (mclength Rhy-Sop))
(setf count-Klav (mclength Rhy-Klav))
    ; mclength zählt die Anzahl Ereignisse einer definierten Variablen, z.B. von Rhy-Vn (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-Sop (rnd-sample 50 '(q q. h h. -e))), sondern
    ; (setf Rhy-Sop (rnd-sample 50 '((q) (q.) (h) (h.) (-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-Vn (mclist (gen-weight 50 '((-1 9)(1 9)))))
(setf Int-Sop (gen-repeat count-Sop (mclist (gen-weight 50 '((-1 9)(1 9)))))
(setf Int-Klav (gen-repeat count-Klav (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 12 (interval-to-pitch Int-Vn)))
  ; 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-Vn
  :velocity '(p)
))
```

```
(setf Sop-Part (make-omn
  :pitch Pitch-Sop
  :length Rhy-Sop
  :velocity '(mp)
))
```

```
(setf Klav-Part (make-omn
  :pitch Pitch-Klav
  :length Rhy-Klav
  :velocity '(p)
))
```

; Tonhöhen und Dauern werden pro Stimme zusammengefügt.

```
(setf list-int1 (rnd-sample 10 '(1 -1)))
(setf list-int2 (rnd-sample 10 '(6 -6 5 -5 1 -1)))
(setf list-int3 (rnd-sample 10 '(1 -1)))
```

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

```
(setf violin (ambitus '(cs6 c7)(bind-to-interval list-int1 Vn-Part)))
(setf soprano (ambitus '(cs4 c5)(bind-to-interval list-int2 Sop-Part)))
(setf piano (ambitus '(cs5 c6) (bind-to-interval list-int3 Klav-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) :soprano-voice (list soprano) :pg (list piano) :tempo 138 :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
; [33"]
```

--

```
; 7. Version -> für Teil 6a
; 9.3.2022
```

```
(progn
(init-seed 33)
(setf Rhy-Vn (rnd-sample 39 '((s s s -s) (s s s s -e) (s s s s s -e))))
(setf Rhy-Sop (rnd-sample 39 '((q) (q.) (h) (h.) (-e) (-q) (-q.)))
(setf Rhy-Klav (rnd-sample 39 '((s s s -s) (s s s s -e) (s s s s s -e))))
```

```
(setf count-Vn (mclength Rhy-Vn))
(setf count-Sop (mclength Rhy-Sop))
(setf count-Klav (mclength Rhy-Klav))
```

; mclength zählt die Anzahl Ereignisse einer definierten Variablen, z.B. von Rhy-Vn (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-Sop (rnd-sample 50 '(q q. h h. -e))), sondern
; (setf Rhy-Sop (rnd-sample 50 '((q) (q.) (h) (h.) (-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-Vn (mclist (gen-weight 50 '((-1 9)(1 4)))))
(setf Int-Sop (gen-repeat count-Sop (mclist (gen-weight 50 '((-1 9)(1 9)))))
(setf Int-Klav (gen-repeat count-Klav (mclist (gen-weight 50 '((-1 9)(1 4)))))
; 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)))
; 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-Vn
  :velocity '(f)
))
(setf Sop-Part (make-omn
  :pitch Pitch-Sop
  :length Rhy-Sop
  :velocity '(f)
))
(setf Klav-Part (make-omn
  :pitch Pitch-Klav
  :length Rhy-Klav
  :velocity '(f)
))

```

; Tonhöhen und Dauern werden pro Stimme zusammengefügt.

```

(setf list-int1 (rnd-sample 10 '(1 -1)))
(setf list-int2 (rnd-sample 10 '(6 -6 5 -5 1 -1)))
(setf list-int3 (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-int1 Vn-Part)))
(setf soprano (ambitus '(b4 c6)(bind-to-interval list-int2 Sop-Part)))
(setf piano (ambitus '(fs1 c6) (bind-to-interval list-int3 Klav-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) :soprano-voice (list soprano) :pg (list piano) :tempo 138 :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"]

```

--

```

; 8. Version -> für Teil 6b
; 9.3.2022

```

```

(progn
  (init-seed 34)
  (setf Rhy-Vn (rnd-sample 39 '((s -h.) (s s -h.) (s s s -h.) (s s s s -h.))))
  (setf Rhy-Sop (rnd-sample 69 '((q) (q.) (h) (h.) (-e) (-q) (-q.) (-h) (-h.))))

```

```
(setf Rhy-Klav (rnd-sample 39 '((s -h.) (s s -h.) (s s s -h.) (s s s s -h.))))
```

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

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

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

; mclength zählt die Anzahl Ereignisse einer definierten Variablen, z.B. von Rhy-Vn (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-Sop (rnd-sample 50 '(q q. h h. -e))), sondern
```

```
; (setf Rhy-Sop (rnd-sample 50 '((q) (q.) (h) (h.) (-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-Vn (mclist (gen-weight 50 '((-1 9)(1 4)))))
```

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

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

; 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)))
```

; 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-Vn
```

```
  :velocity '(pp)
```

```
))
```

```
(setf Sop-Part (make-omn
```

```
  :pitch Pitch-Sop
```

```
  :length Rhy-Sop
```

```
  :velocity '(p)
```

```
))
```

```
(setf Klav-Part (make-omn
```

```
  :pitch Pitch-Klav
```

```
  :length Rhy-Klav
```

```
  :velocity '(pp)
```

```

))
; Tonhöhen und Dauern werden pro Stimme zusammengefügt.

(setf list-int1 (rnd-sample 10 '(1 -1)))
(setf list-int2 (rnd-sample 10 '(6 -6 5 -5 1 -1)))
(setf list-int3 (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-int1 Vn-Part)))
(setf soprano (ambitus '(e4 a5)(bind-to-interval list-int2 Sop-Part)))
(setf piano (ambitus '(fs1 c6) (bind-to-interval list-int3 Klav-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) :soprano-voice (list soprano) :pg (list piano) :tempo 138 :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
; [1:02]

```

--

```

; 9. Version -> für Teil 8a
; 14.3.2022

```

```

(progn
(init-seed 35)
(setf Rhy-Vn (rnd-sample 39 '((s -h.) (s -h.) (s s -h.) (s s -h.) (s s s -h.))))
(setf Rhy-Sop (rnd-sample 69 '((q) (q.) (h) (h.) (-e) (-q) (-q.) (-h) (-h.) (-q.) (-h) (-h.))))
(setf Rhy-Klav (rnd-sample 49 '((s -h.) (s -h.) (s s -h.) (s s -h.) (s s s -h.))))

(setf count-Vn (mclength Rhy-Vn))
(setf count-Sop (mclength Rhy-Sop))
(setf count-Klav (mclength Rhy-Klav))
; mclength zählt die Anzahl Ereignisse einer definierten Variablen, z.B. von Rhy-Vn (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-Sop (rnd-sample 50 '(q q. h h. -e))), sondern
; (setf Rhy-Sop (rnd-sample 50 '((q) (q.) (h) (h.) (-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-Vn (mclist (gen-weight 50 '((-1 9)(1 4)))))
(setf Int-Sop (gen-repeat count-Sop (mclist (gen-weight 50 '((-1 9)(1 9)))))

```

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

; 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 36 (interval-to-pitch Int-Vn)))
```

; 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 24 (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-Vn
  :velocity '(pp)
))
```

```
(setf Sop-Part (make-omn
  :pitch Pitch-Sop
  :length Rhy-Sop
  :velocity '(p)
))
```

```
(setf Klav-Part (make-omn
  :pitch Pitch-Klav
  :length Rhy-Klav
  :velocity '(pp)
))
```

; Tonhöhen und Dauern werden pro Stimme zusammengefügt.

```
(setf list-int1 (rnd-sample 10 '(1 -1)))
(setf list-int2 (rnd-sample 10 '(6 -6 5 -5 1 -1)))
(setf list-int3 (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-int1 Vn-Part)))
```

```
(setf soprano (ambitus '(d4 fs5)(bind-to-interval list-int2 Sop-Part)))
```

```
(setf piano (ambitus '(fs1 c6) (bind-to-interval list-int3 Klav-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) :soprano-voice (list soprano) :pg (list piano) :tempo 138 :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

; [1:12]

--

; 10. Version -> für **Teil 8b** bzw. Teile 8a-8c, 2 Takte Pause dazwischen schieben, ab Takt 43 (Violine 9 Takte Pause, Sopran 11 Takte Pause). Bis Takt 86.

; Ev. Teil 8b nochmals aufteilen in Teile 8b und 8c.

; 14.3.2022

(progn

(init-seed 35)

(setf Rhy-Vn (rnd-sample 1000 '((s -h.) (s -h.) (s s -h.) (s s -h.) (s s s -h.))))

(setf Rhy-Sop (rnd-sample 180 '((q) (q.) (h) (h.) (-e) (-q) (-q.) (-h) (-h.) (-q.) (-h) (-h.))))

(setf Rhy-Klav (rnd-sample 100 '((s -h.) (s -h.) (s s -h.) (s s -h.) (s s s -h.))))

(setf count-Vn (mclength Rhy-Vn))

(setf count-Sop (mclength Rhy-Sop))

(setf count-Klav (mclength Rhy-Klav))

; mclength zählt die Anzahl Ereignisse einer definierten Variablen, z.B. von Rhy-Vn (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-Sop (rnd-sample 50 '(q q. h h. -e))), sondern

; (setf Rhy-Sop (rnd-sample 50 '((q) (q.) (h) (h.) (-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-Vn (mclist (gen-weight 50 '((-1 9)(1 4)))))

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

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

; 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 36 (interval-to-pitch Int-Vn)))

; 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 24 (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-Vn
  :velocity '(pp)
))
```

```
(setf Sop-Part (make-omn
  :pitch Pitch-Sop
  :length Rhy-Sop
  :velocity '(p)
))
```

```
(setf Klav-Part (make-omn
  :pitch Pitch-Klav
  :length Rhy-Klav
  :velocity '(pp)
))
```

; Tonhöhen und Dauern werden pro Stimme zusammengefügt.

```
(setf list-int1 (rnd-sample 10 '(1 -1)))
(setf list-int2 (rnd-sample 10 '(6 -6 5 -5 1 -1)))
(setf list-int3 (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-int1 Vn-Part)))
(setf soprano (ambitus '(d4 fs5)(bind-to-interval list-int2 Sop-Part)))
(setf piano (ambitus '(fs1 c6) (bind-to-interval list-int3 Klav-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) :soprano-voice (list soprano) :pg (list piano) :tempo 138 :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
; [8a & 8b 2:30]

--

Vorgehen

Brainstorming / Planung / Texte

Bearbeitungen, Formplan

Brainstorming-Ende [bre]

Form

Siehe auch " Fassungen".

Form: Immer zwischen variierten A- und B-Teilen abwechseln. (Varianten in den B-Teilen: Ambitus, Gruppenlänge (-> sehr lang bis sehr kurz), Connect-Intervalle

Form-Entwurf:

Teil 5: Extrem gedehnt: Schnipsel aus Teil 1.

Teil 6a: extrem dicht, kurz und laut

Teil 6b: mehr Pausen, länger und leiser

Teil 7: Weitere Schnipsel aus Teil 1. Verschiedene Dehn- und Kompressionsgrade.

Teil 8: Viele und lange Pausen, sehr lang, leise. Nur 1-, 2- und 3-Tongruppen. Vn transp. 36, Klav. transp. 24, Sop d-fis2

Form-Realisation:

Die ungeraden Teile sind die A-Teile, die geraden Teile sind die B-Teile.

- Teil 1: Takte 1-34 [0:59] Eine erste Setzung.
 - Teil 2a: Takte 35-50 [0:28] In Violine und Klavier stets vom „a“ ausgehend.
 - Teil 2b: Takte 51-76 [0:45] Halbton-Connect-System zwischen den einzelnen Gruppen in Violine und Klavier.
 - Teil 3: Takte 77-104 [0:48] Nachklang, Ausklang.
 - Teil 4a: Takte 105-115 [0:19] Tieferes Register in Violine und Klavier. Das Halbton-Connect-System in Violine und Klavier wird (durch die Umfangbeschränkung) oft in eine große Septime umgekehrt.
 - Teil 4b: Takte 116-135 [0:35] Violine und Klavier in hoher Lage und leise (pp). Sopran in relativ tiefer Lage.
 - Teil 5: Takte 136-159 [0:42] Rekomposition und Dehnung der Takte 1-6 (9.3.2022).
 - Teil 6a: Takte 160-175 [0:28] Laut (f) und untere Mittellage für alle. Das Klavier ist oft über der Violine.
 - Teil 6b: Takte 176-211 [1:03] In Violine und Klavier vollziehen über den ganzen Teil eine langsame, große Bewegung, die zuerst nach unten und dann wieder nach oben geht. Leise (pp).
 - ___[Total bis hier: [6:05]
 - Teil 7: Takte 212-228 [0:30] Rekomposition und Dehnung der Takte 6-13 (14.3.2022).
 - Teil 8a: Takte 229-272 [1:17] Violine hoch und leise. Sopran Mittellage. Klavier startet in der Mittellage und geht allmählich nach unten. Besetzung: Der Sopran pausiert die letzten 11 Takte bis zur Pause in Takt 271. Die Violine pausiert die letzten 9 Takte bis zur Pause in Takt 271.
 - Teil 8b: Takte 273-301 [0:50] Die Violine bleibt hoch. Der Sopran bleibt in der Mittellage. Das Klavier bleibt tief. Der Sopran pausiert die ersten 11 Takte. Die Violine pausiert die ersten 9 Takte. Der Sopran bleibt in der Mittellage. (Pausen am Anfang des Teils spiegelbildlich zum Ende von Teil 8a.)
 - Teil 8c: Takte 302-320 [0:33] Die Violine geht tendenziell noch höher hinauf. Das Klavier geht noch tiefer hinab. Der Sopran pausiert in den letzten 6 Takten bis zur Pause in Takt 319.
 - Teil 9: Takte 321-372 [1:30] Rekomposition und Dehnung der Takte 13-34 (15.3.2022).
-

Kontrollen:

- ✓ **xml-Import-Kontrollen (Grobeditierung):**
 - Die Wiedergabe des Tempos einstellen.
 - Die Wiedergabe der dynamischen Angaben einstellen.
 - Die Wiedergabe der Instrumente einstellen.
 - Marginalien
 - 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.)
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 durchgespielt.
- 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.
- Kontrapunkt-Kontrolle. Gelesen Takte ...
- Kontrapunkt-Kontrolle. Am Klavier gespielt Takte ...
- 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[↑], 3[↓]) bzw. ca. einen Viertelton (4[↑], 4[↓]).
- 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 19.3.2022.
 - ✓ AdS-Lexikon: Gemacht am 19.3.2022.
 - Pro Litteris: Gemacht am 20.3.2022.
 - SME: Gemacht am 20.3.2022.
 - 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 21.3.2022.

Konzept / Werkkommentar

Werkkommentar:

Dehnung, Kompression und Entwicklungsfluß. Verschiedene Formen des Atmens. Gedehte Momente. Bögen.

Die Entwicklung der liegenbleibenden Töne.

Verschiedene Momente, Grade und Zwischenbereiche von Dehnung und Kompression der Ereignisdichte prägen die Wahrnehmung dieses Stückes. Große Entwicklungsbögen mit maximal ausdifferenziertem minimalem Material gestalten den formalen Ablauf. Es ging darum, den musikalischen Atem bis in die

Detailstruktur hineinzutragen und Momente der gefärbten Ruhe durch die bewußte Gestaltung des Kontextes in ihren Wahrnehmungsmöglichkeiten vieldeutig zu belassen. (15.3.2022)

Gegenläufige Bewegungen kennzeichnen die formale Entwicklung dieses Stückes im Spannungsfeld zwischen Dehnung und Kompression. Zu Beginn erklingt eine dichte und zupackende Anordnung, die im weiteren Verlauf mehrmals aufgenommen und immer mehr gedehnt wird. Dazwischen entfalten sich große Entwicklungsbögen mit maximal ausdifferenziertem minimalem Material in verschiedenen Dichtegraden, aber immer in einem stetigen Entwicklungsfluß und in verschiedene Richtungen der Register-Gestaltung tendierend. Der musikalische, vibrierende Atem wird bis in die Detailstruktur hineingetragen. Momente der gefärbten Ruhe kontrastieren mit virtuosem Fließen und prägen die Wahrnehmung dieses Stückes, die in der Unausdeutbarkeit verbleibt. (18.3.2022)

Vorwort

Die Haltebögen, die (bei den Einschüben) in die Pausen führen, bedeuten keine Tonverlängerung, sondern zeigen an, daß die Linie nach dem Einschub weitergeht.

Text

Meine Gedichte

- 1) Drangu Sehu
- 2) Aberlein quanimog
- 3) Malfroni
- 4) Siffaz
- 5) Kurz-Holz

Technik

Siehe Protokoll 27.02.2022, 02.03.2022, 03.03.2022, 13.03.2022, 14.03.2022, 15.03.2022.

Halbton-Connect-System: Nicht nur innerhalb der Tongruppen von Violine und Klavier gibt es nur ausschließlich Halbtonschritte, sondern auch von Tongruppe zur nächsten Tongruppe (in Opusmodus: bind-to-interval). Siehe 13.03.2022.

Fassungen:

Siehe auch "Form / Tempi".

Chronologisch:

- 1. Gesamtfassung: 28.02.2022
- 2. Gesamtfassung: 01.03.2022
- 3. Gesamtfassung: 14.03.2022: **3. Fassung** unter **Trio für Sop, Vn u. Pfte Nr. 2-3.musx** abspeichern und nach Takt 49 einen Pausentakt einschieben. Dadurch verschieben sich die folgenden Taktnummern um eine Zahl nach oben.

Kritik / Fragen:

Titel:

Trio für Sopran, Violine und Klavier Nr. 2