

***Возможность алгебраических операций над музыкальным текстом:
ритм, интонация, тембр***

***The possibility of algebraic operations on a musical text:
rhythm, intonation, timbre***

音樂文本代數運算的可能性：節奏、語調、音色

Амосов Г.Г.

*доктор физико-математических наук, профессор,
ведущий научный сотрудник Математического института им. В.А. Стеклова РАН
Москва, Россия*

*Doctor of Physical and Mathematical Sciences, Professor,
Leading Scientific Researcher, Steklov Mathematical Institute of RAS
Moscow, Russia*

*物理與數學科學博士、教授、
俄羅斯科學院斯特克洛夫數學研究所首席研究員
俄羅斯莫斯科
gramos@mi-ras.ru*

Комиссаренко А.В.

*доцент кафедры композиции Научно-композиторского факультета
Московской государственной консерватории им. П.И. Чайковского
Москва, Россия*

*Associate Professor of Composition Department, Faculty of Science and Composition,
Moscow Tchaikovsky Conservatory
Moscow, Russia*

*科學與作曲學院作曲系副教授，莫斯科柴可夫斯基音樂學院
俄羅斯莫斯科
anjelikamiss@mail.ru*

Аннотация. Запись музыки в виде некоторого текста превращает музыкальный материал в структурированное множество, над которым возможно выполнять определенные математические операции. В зависимости от того, какие именно структурные части мы выделяем, можно говорить о записи ритма, звуковысотности, тембра. Данное выделение и обеспечивает качество математических операций, среди которых могут быть такие как сложение и умножение, по отношению к тексту. Мы анализируем алгебраические операции, встречающиеся в практике современных композиторов. В качестве примеров будут приведены сочинения Т. Мюрая и Н. Хруста, а также теория ритмов А. Грамши. Используется объектно-ориентированный язык программирования OpenMusic.

Ключевые слова: запись музыки, алгебраические операции над музыкальным текстом, объектно-ориентированный язык OpenMusic.

Annotation. Recording music in the form of some text turns the musical material into a structured set over which it is possible to perform certain mathematical operations. Depending on which structural parts we select, we can talk about recording rhythm, pitch, timbre. This selection ensures the quality of mathematical operations, among which may be such as addition and multiplication, in relation to the text. We analyze algebraic operations encountered in the practice of modern composers. The works of T. Murail and N. Khrust, as well as the theory of rhythms by A. Gramsci will be given as examples. The object-oriented programming language OpenMusic is used.

Keywords: music recording, algebraic operations on a musical text, object-oriented language OpenMusic.

註解。 以某種文字形式錄製音樂會將音樂材料轉變為結構化的集合，可以在其上執行某些數學運算。根據我們選擇的結構部分，我們可以談論錄製節奏、音調、音色。這種選擇確保了數學運算的質量，其中可能包括與文字相關的加法和乘法。我們分析現代作曲家實踐中所遇到的代數運算。T. Murail 和 N. Khrust 的作品以及 A. Gramsci 的節奏理論將作為例子。使用物件導向的程式語言 OpenMusic。

關鍵字：音樂錄音、音樂文本的代數運算、物件導向語言 OpenMusic。

1. Введение

Проблема хранения и обработки информации, представляющей из себя некоторый набор интеллектуального знания, особенно остро проявляет себя в разных областях науки, техники и искусства в последние годы в связи с появлением больших массивов данных и появлением таких новых методов обработки как квантовые [1]. В 2019 году была принята национальная стратегия развития искусственного интеллекта [2], что предполагает проникновение новых методов во все области знания.

Музыка (как явление искусства) обладает целым рядом параметров, включая звуковысотность, ритм и тембр. Конечно, все эти параметры имеют как физическую так и психоакустическую составляющие [3]. Для более скрупулезных сведений о принципах создания современной музыки, использующих электронные устройства, мы рекомендуем ознакомиться с учебником [4]. При традиционной нотации наибольшее значение играет запись звуковысотности, а проблема указания тембра вообще сводится к названию инструмента, на котором должно быть осуществлено исполнение. Что же касается ритма, то о нём делаются самые общие указания с надеждой на интерпретацию профессиональными музыкантами. Часто такой подход не годится для записи современных сочинений, в которых ритм и тембр играют ничуть не меньшую роль, чем интонационные мотивы.

Проявление тембрального тематизма в музыке Д.Д. Шостаковича было отмечено ещё в [5]. Далеко идущее развитие таких идей привело к возникновению работы [6], где было указано на связь между переходом от интонационного тематизма к тембральному и появлением в XX веке квантовых методов обработки информации. Использование электронных инструментов для

создания звука определенного тембра имеет богатую историю. Россия тут стоит на самых впечатляющих позициях [7]. Следует заметить, что попытки исполнения музыкальных композиций на компьютере, оказывается требуют серьезных дополнений к существующим формальным алгоритмам, что ярко продемонстрировал прославленный патриарх исследований в области компьютерной музыки Max Mathews [8].

Для решения создавшейся проблемы можно использовать как новые способы нотной записи, так и старые, придавая им некоторый новый смысл. В своих исследованиях мы опираемся на объектно-ориентированный язык программирования OpenMusic, созданный в парижском IRCAM [9]. Данный язык программирования, написанный на традиционном языке Lisp, позволяет визуализировать информацию, содержащуюся в строках с помощью удобных “иконок”, часто имеющих вид традиционной музыкальной нотации, но с несколько другим смыслом, позволяющим четко задавать не только звуковысотность, но и структуру ритма и тембра.

Мы показываем возможности такого подхода для реализации трех совершенно различных видов музыки. Это, так называемая, спектральная музыка Т. Murail, теории ритмов Антонио Грамши, а также одного сочинения Н. Хруста, построенного сходно с теорией А. Грамши, но только с заменой интерпретации параметров, задающих ритм, на параметры звуковысотности. Во всех трех случаях мы будем использовать возможности языка программирования OpenMusic.

Анализ возможностей использования OpenMusic для изучения принципов строения спектральной музыки для части пьесы Tristan Murail “Désintégrations” был предпринят нами в [10]. С основами теории ритмов Антонио Грамши можно ознакомиться по видео [11-12]. Сочинение Николая Хруста “Евгеника III. Feedback door man”, при написании которого применялся язык OpenMusic, было исполнено на фестивале “Пять вечеров+1” в Москве в 2021 г. [13]. Это сочинение проливает свет на связь развития музыкального произведения с математической теорией сложности, что также можно извлечь из давней музыкально-теоретической работы Ю.М. Буцко, воспроизведенной в [14].

2. Объектно-ориентированный язык программирования OpenMusic

Любой язык программирования содержит два класса объектов: для хранения информации и осуществляющих операции над информацией. Самая общая классическая схема вычислений представляет последовательность действий над информацией, хрянящейся в битах [15].

Среда OpenMusic была написана на языке программирования Lisp, в котором как объекты, предназначенные для хранения информации, так и функции над этими объектами задаются при помощи строк, разделенных открывающимися и закрывающимися скобками. Основным достоинством

OpenMusic, по сравнению с обычным языком программирования, является удобный графический интерфейс, вид которого часто отвечает традиционной музыкальной нотации.

Ниже мы приведем несколько основных типов объектов, которые будем использовать. Во-первых, это, конечно, просто строка, состоящая из цифр и скобок. Такой способ хранения данных характерен для Lisp. Обыкновенно такая строка наделяется структурой дерева при помощи скобок:

```
((6100) (6000) (6000) (6100) (6000) (6000) (6000) (6000))
```

Рис. 1. Основной способ хранения данных

Fig. 1. Main method of data storage

圖 1. 儲存資料的主要方法

На Рис. 1 изображена последовательность четырехзначных цифр, разделенных скобками. Звуковысотность нот в OpenMusic принято измерять в мидицентах. В отличие от единиц звуковысотности в миди, измеряемых в единицах, приписываемых белым и черным клавишам рояля с тем расчетом, что “ля” первой октавы имеет номер 69, на один мидицент приходится в сто раз большая величина. Это делается для того, чтобы можно было записывать ноты с более сложной звуковысотностью, например, четвертьтона. Получается, что мы видим последовательность нот первой октавы, состоящую из до и до#.



Рис. 2. Объект для хранения данных Voice

Fig. 2. Voice data storage object

圖 2. 語音資料儲存對象

Другой типичный пример объекта, предназначенного для хранения данных, приведён на Рис. 2. Тут, тем не менее, следует сразу оговориться. Хотя то, что мы видим, очень напоминает обычную музыкальную нотацию, в эту запись может вкладываться принципиально другой смысл. Например, данная

последовательность может определять ритм. В этом случае доли длительности нот задают структуру ритма, а звуковысотность на нотном стане означает выбор инструмента на котором будет происходить извлечение звука (высокие или низкие бонги, конги и т.п.)

Нотные записи первого и второго типа, изображенные на Рис. 1 и Рис. 2, могут быть преобразованы друг в друга. При этом наглядное графическое изображение Рис. 2 превращается в сложное дерево Рис. 1 с большим количеством скобок. Отметим, что строками вида, изображенного на Рис. 1 можно определять не только последовательность нот или аккордов, но и ритмические деревья, простейший пример которых изображен на Рис. 3. Цифры на рисунке означают последовательно: количество тактов (один), размер такта (четыре четверти) и деление такта на отдельные доли (2+1+1).

(1 (((4 4) (2 1 1))))

Рис. 3. Простейший пример записи ритмического дерева

Fig. 3. The simplest example of recording a rhythm tree

圖 3. 錄製節奏樹最簡單的範例

Как уже было указано выше, ко второму классу объектов относятся функции, позволяющие изменять информационные данные. Мы не ставим для себя целью описать все подобные функции. Это невозможно, да и не нужно. Поскольку далее речь пойдёт об использовании алгоритмов при создании музыкальных сочинений, нам потребуется, прежде всего, создание циклов, при котором будут последовательно перебираться некоторые данные.

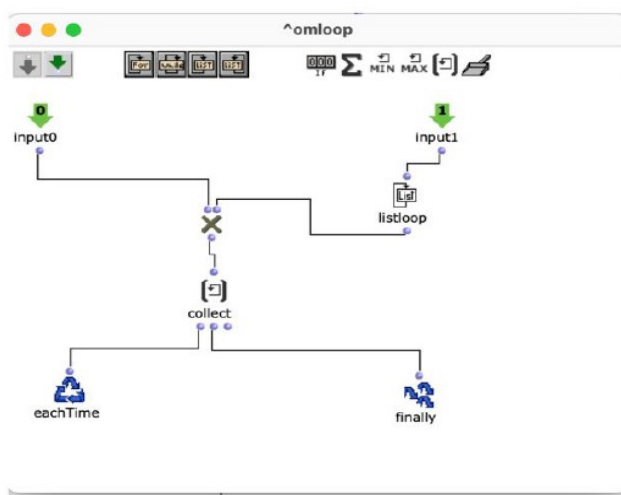


Рис. 4. Организация циклов

Fig. 4. Cycle organization

圖 4. 循環組織

Основным объектом, осуществляющим данную операцию в OpenMusic является *omloop*, внешний вид которого приведен на Рис. 4. На Рис. 4 мы видим, что числа, содержащиеся на входах *input0* и *input1* перемножаются друг с другом. Программа создаёт список, содержащий результаты такого перемножения. Здесь мы сталкиваемся с ещё одной операцией над данными, которая является их обычным умножением друг с другом. Конечно, определено и сложение. Эти операции мы не будем обсуждать подробно.

3. Анализ одной части сочинения Тристана Мюра "Désintégrations"

Пьеса "Désintégrations", написанная в стиле спектральной музыки Тристаном Мюраем в 1982-83 гг., состоит из 11 частей, для каждой из которых характерна своя работа со спектром, ритмом и текстурой. Мы остановимся на разъяснении идеи, лежащей в основе 10-ой части [10]. В основе данной части пьесы лежит искажение тембра тромбона, который играет ноту ми большой октавы. В качестве точки отсчета берется 12-ый обертон ноты. С каждым новым аккордом он повышается на четвертьтона. Оставшиеся 11 гармоник растягиваются пропорционально. Кроме того, такой динамический процесс, воспринимающийся на слух как увеличение диссонантности, ускоряется во времени.

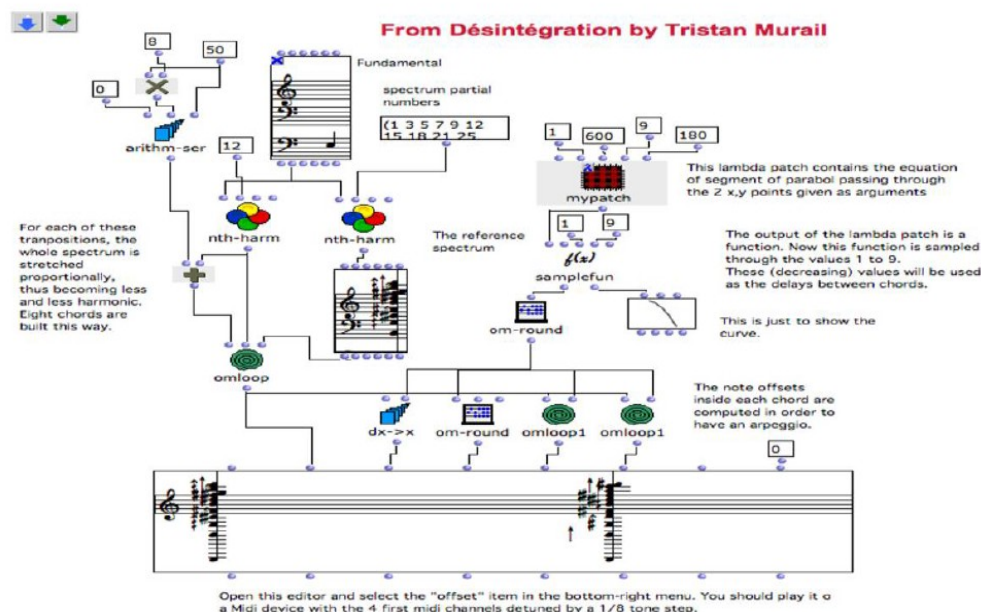


Рис. 5. Патч, моделирующий последовательность аккордов

Fig. 5. Patch modeling a chord progression

圖 5. 和弦進行的 Patch 建模

Для генерации последовательности аккордов, лежащей в основе 10-ой части

пьесы можно воспользоваться патчем, написанным в OpenMusic. На Рис 5 ниже изображен патч, позволяющий построить последовательность всё более расстраиваемых аккордов.

Программа работает сверху вниз. Сначала из звука ми большой октавы извлекаются обертоны с номерами, указанными в окошке, справа от ноты. Получается звук ми с тембром как у тромбона. Его нотная запись можно увидеть ниже. Далее, строится последовательность аккордов со всё более искаженным тембром. Итоговая последовательность аккордов помещена в объект Voice внизу Рис. 5.

4. Алгебраические операции в теории ритмов Антонио Грамши

Под ритмом обыкновенно понимается некоторая повторяющаяся структура. Именно в таком виде она появилась у Антонио Грамши - музыканта-мультиинструменталиста и преподавателя школы восточных барабанов. Его интересовало строение существующих ритмов, которые можно выстукивать на барабанах, а также, что для нас более интересно, операции, которые можно осуществлять над ритмами. Оказалось, что такие операции, как достаточно естественным образом определяемые сложение и умножение, превращают множество ритмов (точное определение дадим ниже) в алгебраическую структуру. Взяв за основную операцию возведение ритма в степень, мы обнаружили, что изначально прозрачную ритмическую структуру можно значительно усложнять во времени.

Простой ритмический рисунок можно изобразить последовательностью из единиц и нулей, где единица означает наличие удара, а ноль — паузу [11-12]. Сложив количество последовательных нулей со стоящей слева от них единицей, получим запись вида |422111|. Конечно, умножая каждое из стоящих в записи чисел на одно и то же число, получим тот же самый ритм, поскольку единица времени не имеет абсолютного значения. Например, запись |8442222| или |16884444| задаёт один ритм. Над ритмами, записанными в таком виде, можно определить значительное число различных алгебраических операций. Для нас их важнейшими примерами являются сложение (конкатенация) и умножение. Результатом сложения двух ритмов является их последовательное исполнение. Более сложную структуру имеет умножение, результатом которого является исполнение одного ритма в ритме другого. Формально математически это означает покомпонентное перемножение цифр, задающих ритмический рисунок. Так |422111| умножить на |211| даёт |844422422211211211211|.

Чистый ритм (без звуковысотности) может воспроизводиться на различных ударных инструментах, имеющих нумерацию миди. Для них зарезервирован канал 10 в компьютерных программах для воспроизведения звука. При этом

можно возводить определенные базовые ритмы в различные степени и воспроизводить сразу нескольких дорожек, содержащих степени ритмов, разными инструментами. Наиболее интересный результат получается при возведении ритма в достаточно высокую (3 или 5) степень. Мы воспользовались для получения таких компьютерных композиций программой OpenMusic. Ниже на Рис. 6 приведен патч, осуществляющий возведение в 5, 6 и 3 степень различных ритмов. После чего они объединяются в объекте Poly.

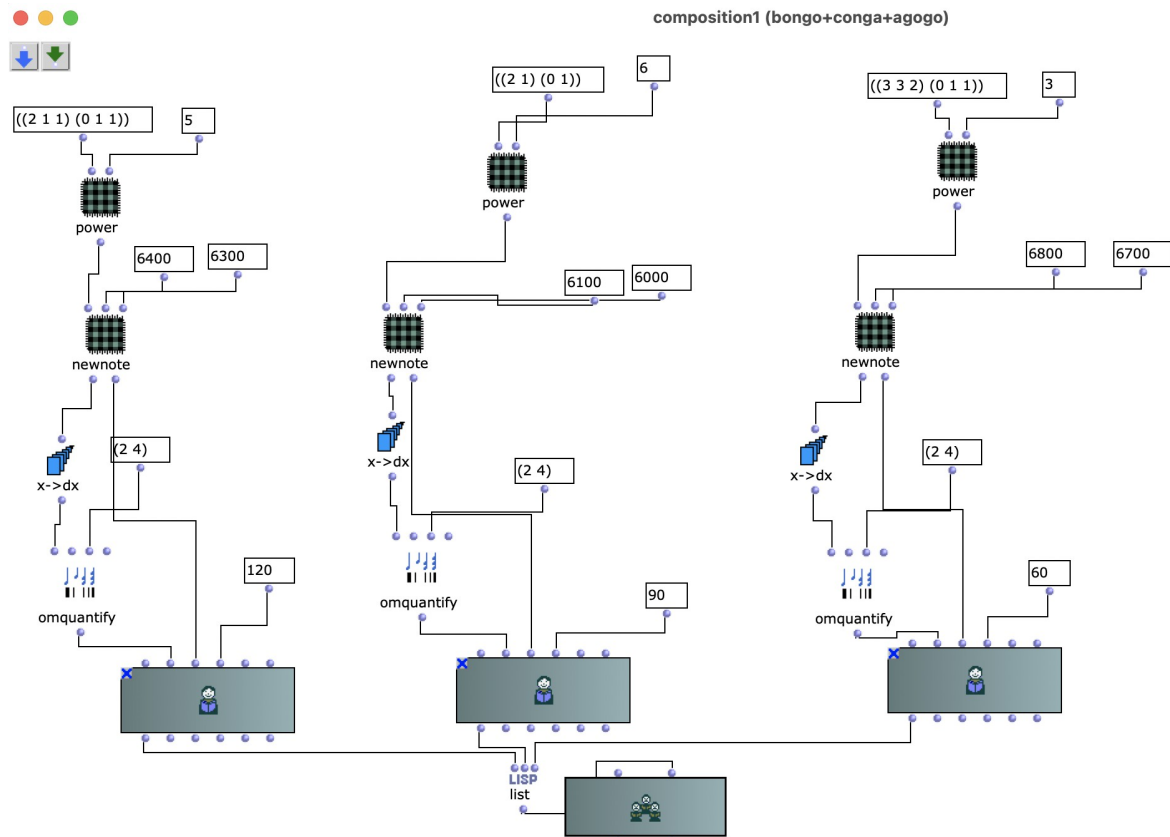


Рис. 6. Патч, строящий одновременное исполнение трех разных ритмов
 Fig. 6. A patch that builds the simultaneous performance of three different rhythms
 圖 6. 建構三種不同節奏同時演奏的補丁

Приведем напоследок картинку с нотным станом, демонстрирующую результат действия программы OpenMusic. Взглянув на Рис. 7 можно подумать, что мы видим обычную нотную запись. На самом деле, звуковысотность тут означает номер инструмента миди, на котором исполняется данный ритм, а длительность звучания ноты как раз и передает ритмический рисунок.

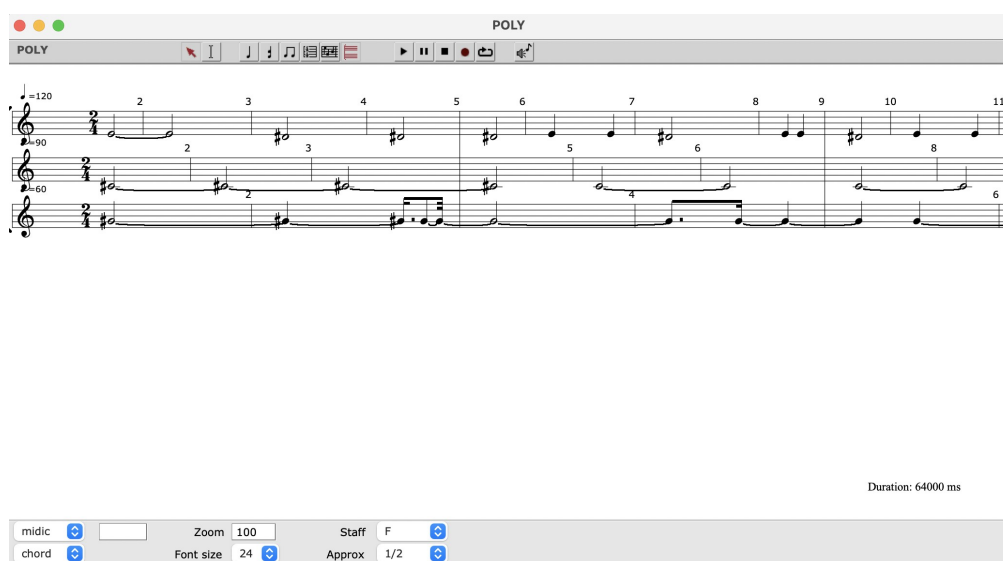


Рис. 7. Результат действия программы, выводящей композицию ритмов

Fig. 7. The result of a program that displays a composition of rhythms

圖 7. 顯示節奏組合的程式結果

5. Возведение интонации в степень в сочинении “Евгеника III. Feedback door man” Николая Хруста

Ознакомившись с теорией алгебраических операций над ритмами композитор Николай Хруст предложил похожие операции для звуковысотного нотного текста. В качестве параметров ноты были выбраны длительность и высота. При умножении двух нот их длительности складываются, а высоты перемножаются по модулю хроматической гаммы. Такую операцию можно сопоставить сложному умножению двух ритмов, где параметров у каждого из компонентов два. Один из них отвечает за непосредственно величину доли, а второй принимает два значения, одно из которых может быть названо условно “глухой”, а второе “звонкий”. Для ритмов при таком умножении принимается правило, согласно которому умножение двух глухих и двух звонких даёт глухой, а в противном случае, получается звонкий.

Усложнение ритмического рисунка при возведении в степень удивительным образом проявилось и при новом определении умножения. Построение музыкальной пьесы Н. Хруста эксплуатирует принцип повторяемости с целью обогащения на каждом новом витке. Именно это свойство определяет смысл названия “Евгеника”. Ниже, на Рис. 8 приведен простейший патч, написанный в программе OpenMusic, который осуществляет такую операцию.

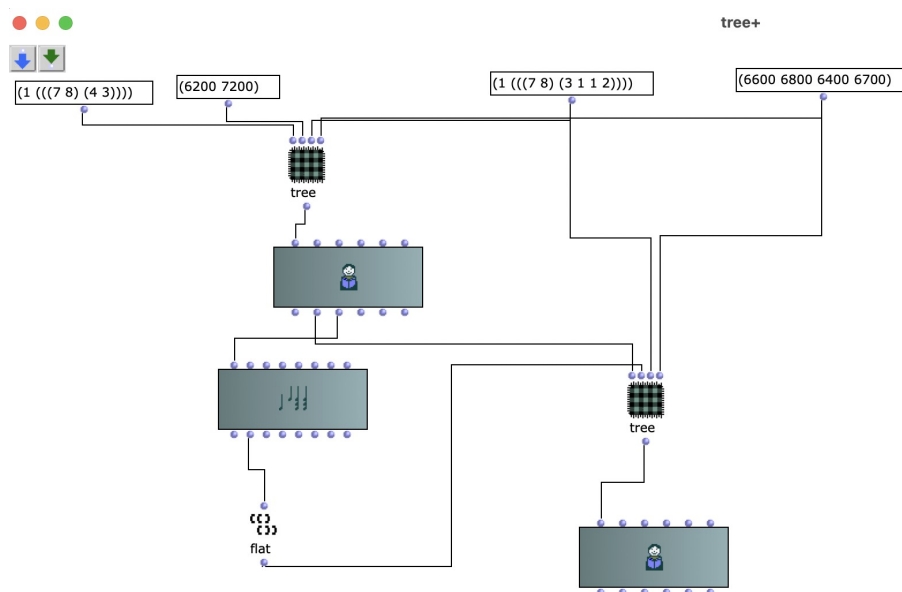


Рис. 8. Результат умножения двух интонационных рисунков
 Fig. 8. Result of multiplying two intonation patterns
 圖 8. 兩種語調模式相乘的結果

В окошках сверху патча на Рис. 8 мы видим две простейшие мелодии. Тут четырехзначные цифры задают последовательность высот нот, а цифры со сложной системой скобок определяют тактовый размер. Сочинение Николая Хруста, которое, конечно, значительно сложнее приведенных математических схем, было исполнено на фестивале современной музыки в Рахманиновском зале Московской консерватории [13].

6. Заключение

Мы продемонстрировали возможности объектно-ориентированного языка программирования OpenMusic, написанного специально для нужд композиторов, в трех различных случаях. Это спектральная музыка, позволяющая моделировать различные динамически меняющиеся тембры, алгебраические операции над ритмами и перенос алгебраических операций с ритмов на звуковысотные тексты. Безусловно, этим не исчерпываются все возможности использования алгоритмов при написании музыкального сочинения. Показано, что такой психоакустический результат как ощущение постепенного нарастания сложности может быть получен с помощью использования алгоритмов.

1. Introduction

The problem of storing and processing information representing a certain set of intellectual knowledge has been particularly acute in various fields of science, technology and art in recent years due to the emergence of large data arrays and the emergence of such new processing methods as quantum [1]. In 2019, the national strategy for the development of artificial intelligence was adopted [2], which implies the penetration of new methods into all areas of knowledge.

Music (as an art phenomenon) has a number of parameters, including pitch, rhythm and timbre. Of course, all these parameters have both physical and psychoacoustic components [3]. For more scrupulous information about the principles of creating modern music using electronic devices, we recommend reading the textbook [4]. With traditional notation, the recording of pitch plays the greatest importance, and the problem of specifying the timbre in general is reduced to the name of the instrument on which the performance should be fulfilled. As for the rhythm, the most general instructions are made with the hope of interpretation by professional musicians. Often this approach is not suitable for recording modern compositions in which rhythm and timbre play no less a role than intonation motives.

The manifestation of timbral thematism in the music of D.D. Shostakovich was noted in [5]. The far-reaching development of such ideas led to the emergence of work [6], where it was pointed out the connection between the transition from intonational thematism to timbral and the appearance of quantum information processing methods in the XX century. The use of electronic instruments to create a sound of a certain timbre has a rich history. Russia is in the most impressive positions here [7]. It should be noted that attempts to perform musical compositions on a computer, it turns out, require serious additions to existing formal algorithms, which was vividly demonstrated by the renowned patriarch of research in the field of computer music Max Mathews [8].

To solve this problem, you can use both new methods of musical notation and old ones, giving them some new meaning. In our research, we rely on the object-oriented programming language OpenMusic, created in Paris IRCAM [9]. This programming language, written in the traditional Lisp language, allows you to visualize the information contained in strings using convenient “icons”, often having the form of traditional musical notation, but with a slightly different meaning, allowing you to clearly set not only the pitch, but also the structure of rhythm and timbre.

We show the possibilities of this approach for the implementation of three completely different types of music. This is the so-called spectral music of T. Murail, the theory of rhythms by Antonio Gramsci, as well as one composition by N. Khrust, constructed similarly to the theory of A. Gramsci, but only with the replacement of the interpretation of the parameters that set the rhythm with the parameters of pitch. In all three cases, we will use the capabilities of the Open Music programming

language.

The analysis of the possibilities of using OpenMusic to study the principles of the structure of spectral music for a part of Tristan Murail's piece "Désintégrations" was undertaken by us in [10]. The basics of Antonio Gramsci's theory of rhythms can be found in the video [11-12]. The work of Nikolai Khrust "Eugenics III. Feedback door man", which was written using the OpenMusic language, was performed at the festival "Five Evenings + 1" in Moscow in 2021 [13]. This work sheds light on the connection between the development of a musical composition and the mathematical theory of complexity, which can also be extracted from the long-standing musical-theoretical work of Yu.M. Butsko, reproduced in [14].

2. Object-oriented programming language OpenMusic

Any programming language contains two classes of objects: for storing information and performing operations on information. The most general classical scheme of calculations represents a sequence of actions on information stored in bits [15].

The OpenMusic environment was written in the programming language Lisp, in which both objects intended for storing information and functions over these objects are specified using strings separated by opening and closing brackets. The main advantage of OpenMusic, in comparison with a conventional programming language, is a user-friendly graphical interface, the appearance of which often corresponds to traditional musical notation.

Below we will list a few basic types of objects that we will use. Firstly, it is, of course, just a string consisting of numbers and brackets. This way of storing data is typical for Lisp. Usually such a string is endowed with a tree structure using brackets:

Fig. 1. The main way to store data

Fig. 1 shows a sequence of four-digit numbers separated by parentheses. The pitch of notes in OpenMusic is usually measured in midicents. Unlike the units of pitch in midi, measured in units attributed to the white and black keys of the piano supposing that middle La (A-440) has the number 69, one midicent accounts for a hundred times more. This is done so that it is possible to record notes with a more complex pitch, for example, a quarter tone. It turns out that we see a sequence of notes consisting of middle C and C#.

Fig. 2. Object for storing data Voice

Another typical example of an object intended for data storage is shown in Fig. 2. Here, however, you should immediately make a reservation. Although what we see is very similar to ordinary musical notation, a fundamentally different meaning can

be put into this recording. For example, this sequence can determine the rhythm. In this case, the fractions of the duration of the notes set the rhythm structure, and the pitch on the sheet music means the choice of the instrument on which the sound will be extracted (high or low bongos, congas, etc.)

Musical notes of the first and second types, shown in Fig. 1 and Fig. 2, can be converted into each other. At the same time, the visual graphic image of Fig. 2 turns into a complex tree of Fig. 1 with a large number of brackets. Note that the lines of the type shown in Fig. 1 can determine not only the sequence of notes or chords, but also rhythmic trees, the simplest example of which is shown in Fig. 3. The numbers in the figure mean sequentially: the number of bars (one), the size of the bar (four quarters) and the division of the bar into separate fractions (2+1+1).

Fig. 3. The simplest example of recording a rhythmic tree

As already mentioned above, the second class of objects includes functions that allow you to change information data. We do not aim to describe all such functions. This is impossible, and it is not necessary. Since next we will talk about the use of algorithms when creating musical compositions, we will need, first of all, the creation of cycles in which some data will be sequentially sorted through. The main object performing this operation in OpenMusic is omloop, the appearance of which is shown in Fig. 4.

In Fig. 4, we see that the numbers contained in the inputs input 0 and input 1 are multiplied with each other. The program creates a list containing the results of such multiplication. Here we are faced with another operation on the data, which is their usual multiplication with each other. Of course, addition is also defined. We will not discuss these operations in detail.

Fig. 4. Organization of cycles

3. Analysis of one part of Tristan Murail's composition "Désintégrations"

The piece "Désintégrations", written in the style of spectral music by Tristan Murail in 1982-83, consists of 11 parts, each of which is characterized by its own work with spectrum, rhythm and texture. We will focus on explaining the idea underlying the 10th part [10]. The basis of this part of the piece is the distortion of the timbre of the trombone, which plays E of the great octave. The 12th overtone of the note is taken as a reference point. With each new chord, it rises by a quarter of a tone. The remaining 11 harmonics are stretched proportionally. In addition, such a dynamic process, perceived by ear as an increase in dissonance, accelerates over time.

To generate the chord sequence underlying the 10th part of the piece, you can use a patch written in OpenMusic. Fig. 5 below shows a patch that allows you to build a sequence of increasingly frustrated chords.

Fig. 5. A patch that simulates a chord sequence

The program works from top to bottom. First, overtones with the numbers indicated in the window to the right of the note are extracted from the sound of E of the great octave. It turns out the sound of E with the timbre like a trombone. His musical notation can be seen below. Next, a chord sequence is constructed with an increasingly distorted timbre. The final chord sequence is placed in the Voice object at the bottom of Fig. 5.

4. Algebraic operations in the theory of rhythms by Antonio Gramsci

Rhythm is usually understood as a kind of repetitive structure. It was in this form that it appeared in Antonio Gramsci, a multi-instrumentalist musician and teacher of the school of oriental drums. He was interested in the structure of existing rhythms that can be tapped on the drum, and also, what is more interesting for us, the operations that can be performed on rhythms. It turned out that operations such as addition and multiplication, which are quite naturally defined, turn a set of rhythms (we will give an exact definition below) into an algebraic structure. Taking the exponentiation of rhythm as the main operation, we found that the initially transparent rhythmic structure can be significantly complicated in time.

A simple rhythmic pattern can be represented by a sequence of ones and zeros, where one means the presence of a beat, and zero means a pause [11-12]. Adding up the number of consecutive zeros with one standing to the left of them, we get an entry of the form |4221111|. Of course, multiplying each of the numbers in the record by the same number, we get the same rhythm, since the unit of time has no absolute value. For example, the entry |8442222| or |16884444| sets one rhythm. A significant number of different algebraic operations can be defined over rhythms written in this form. For us, their most important examples are addition (concatenation) and multiplication. The result of the addition of two rhythms is their consistent performance. Multiplication has a more complex structure, the result of which is the performance of one rhythm in the rhythm of another. Formally, mathematically, this means a component-by-component multiplication of numbers that define a rhythmic pattern. So |4221111| multiplied by |211| gives |844422422211211211211|.

Fig. 6. A patch that builds simultaneous performance of three different rhythms

A pure rhythm (without pitch) can be played on various percussion instruments that have midi numbering. Channel 10 is reserved for them in computer programs for sound reproduction. At the same time, it is possible to build certain basic rhythms to various degrees and play several tracks containing degrees of rhythms at once with different instruments. The most interesting result is obtained when the rhythm is raised to a sufficiently high (3 or 5) degree. We used the Open

Music program to obtain such computer compositions. Below in Fig. 6 shows a patch that performs the construction of 5, 6 and 3 degrees of different rhythms. After that, they are combined in the object Poly.

Fig. 7. The result of the program that outputs the composition of rhythms

Finally, here is a picture with a sheet music mill, demonstrating the result of the Open Music program. Looking at Fig. 7, you might think that we are seeing an ordinary musical notation. In fact, the pitch here means the number of the midi instrument on which this rhythm is performed, and the duration of the sound of the note just conveys the rhythmic pattern.

5. Raising intonation to a degree in the composition “Eugenica III. Feedback door man” by Nikolai Khrust

Having familiarized himself with the theory of algebraic operations on rhythms, composer Nikolai Khrust proposed similar operations for a high-pitched musical text. Duration and pitch were chosen as the parameters of the note. When two notes are multiplied, their durations are added, and the heights are multiplied modulo the chromatic scale. Such an operation can be compared to a complex multiplication of two rhythms, where each of the components has two parameters. One of them is directly responsible for the value of the fraction, and the second takes two values, one of which can be called conditionally “deaf”, and the second “sonorous”. For rhythms with such multiplication, a rule is adopted according to which multiplication of two deaf and two sonorous gives a deaf, and otherwise, a sonorous one is obtained.

The complication of the rhythmic pattern during exponentiation was surprisingly manifested in the new definition of multiplication. The construction of a musical piece by N. Khrust exploits the principle of repeatability in order to enrich at each new turn. It is this property that determines the meaning of the name “Eugenica”. Below, Figure 8 shows the simplest patch written in the Open Music program, which performs such an operation.

Fig. 8. The result of multiplying two intonation patterns

In the windows at the top of the patch in Figure 8, we see two simple melodies. Here, four-digit numbers set the sequence of note heights, and numbers with a complex system of brackets determine the clock size. The composition of Nikolai Khrust, which, of course, is much more complicated than the mathematical schemes given, was performed at the festival of contemporary music in the Rachmaninoff Hall of the Moscow Conservatory [13].

6. Conclusion

We demonstrated the capabilities of the object-oriented programming language

OpenMusic, written specifically for the needs of composers, in three different cases. This is spectral music that allows you to simulate various dynamically changing timbres, algebraic operations on rhythms and the transfer of algebraic operations from rhythms to sound pitch texts. Of course, this does not exhaust all the possibilities of using algorithms when writing a musical composition. It is shown that such a psychoacoustic result as a feeling of a gradual increase in complexity can be obtained by using algorithms.

1、簡介

近年來，由於大數據陣列的出現以及量子等新處理方法的出現，存儲和處理代表某一組智力知識的信息的問題在科學、技術和藝術的各個領域變得尤為尖銳。]。 2019 年，國家人工智慧發展策略的推出[2]，意味著新方法滲透到各個知識領域。

音樂（作為藝術現象）有許多參數，包括音高、節奏和音色。當然，所有這些參數都具有物理和心理聲學成分 [3]。有關使用電子設備創作現代音樂的原理的更多詳細信息，我們建議閱讀教科書 [4]。對於傳統的記譜法，音高的記錄起著最重要的作用，而一般指定音色的問題則被簡化為要完成演奏的樂器的名稱。至於節奏，最籠統的說明是希望專業音樂家的詮釋。通常，這種方法不適合錄製節奏和音色與語調動機同樣重要的現代作品。

D.D. 音樂中音色主題主義的表現 蕭斯塔科維奇在[5]中被指出。這些思想的深遠發展導致了著作[6]的出現，其中指出了從語調民族主題主義到音色的轉變與二十世紀量子資訊處理方法的出現之間的聯繫。使用電子樂器來創造某種音色的聲音有著悠久的歷史。俄羅斯在這方面處於最令人印象深刻的位置 [7]。值得注意的是，事實證明，嘗試在電腦上演奏音樂作品需要對現有的形式演算法進行認真的補充，電腦音樂領域的著名研究元老 Max Mathews 生動地證明了這一點 [8]。

為了解決這個問題，你可以同時使用新的樂譜方法和舊的樂譜方法，賦予它們一些新的意義。在我們的研究中，我們依賴巴黎 IRCAM 創建的物件導向程式語言 OpenMusic [9]。這種以傳統 Lisp 語言編寫的程式語言可讓您使用方便的「圖示」視覺化字串中包含的訊息，通常具有傳統樂譜的形式，但含義略有不同，使您不僅可以清楚地設定音高，還包括節奏和音色的結構。

我們展示了這種方法實現三種完全不同類型音樂的可能性。這就是 T. Murail 所謂的光譜音樂，Antonio Gramsci 的節奏理論，以及 N. Khrust 的一首作品，其構建與 A. Gramsci 的理論類似，但只是替換了對用音高參數設置節奏的參數。在所有這三種情況下，我們都將使用 Open Music 程式語言的功能。

我們在 [10] 中對使用 OpenMusic 研究 Tristan Murail 作品

「Désintégrations」的頻譜音樂結構原理的可能性進行了分析。安東尼奧·葛蘭西節奏理論的基礎知識可以在影片中找到[11-12]。尼古拉·赫魯斯特的著作《優生學 III. 使用 OpenMusic 語言編寫的《反饋門人》曾在 2021 年莫斯科「五晚+1」音樂節上演出[13]。這部作品揭示了音樂作品的發展與複雜性數學理論之間的聯繫，這也可以從 Yu.M. 的長期音樂理論著作中提取出來。Butsko，轉載於[14]。

2. 物件導向的程式語言 OpenMusic

任何程式語言都包含兩類物件：用於儲存資訊和對資訊執行操作。最通用的經典計算方案表示對儲存在位元中的資訊執行一系列操作[15]。

OpenMusic 環境是用程式語言 Lisp 編寫的，其中用於儲存資訊的物件和這些物件上的函數都使用由左括號和右括號分隔的字串指定。與傳統的程式語言相比，OpenMusic 的主要優點是使用者友善的圖形介面，其外觀通常與傳統的樂譜相對應。

下面我們將列出我們將使用的一些基本類型的物件。首先，它當然只是一個由數字和括號組成的字串。這種儲存資料的方式是 Lisp 的典型方式。通常這樣的字串使用括號賦予樹狀結構：

圖 1. 儲存資料的主要方式

圖 1 顯示了由括號分隔的四位數字序列。OpenMusic 中音符的音高通常以中音來衡量。與 midi 中的音高單位不同，假設中音 La (A-440) 的數字為 69，則以鋼琴的白鍵和黑鍵的單位來測量，而 1 midicent 的音高則多出一百倍。這樣做是為了可以錄製具有更複雜音高的音符，例如四分音。事實證明，我們看到了由中間的 C 和 C# 組成的一系列音符。

圖 2. 儲存資料 Voice 的對象

用於資料儲存的物件的另一個典型範例如圖 2 所示。但是，您應該立即進行預訂。儘管我們所看到的與普通的樂譜非常相似，但這段錄音可以表達出根本不同的意義。例如，這個順序可以決定節奏。在這種情況下，音符持續時間的分數決定了節奏結構，樂譜上的音高意味著提取聲音的樂器的選擇（高音或低音煙槍、康加鼓等）

圖 1 和圖 2 所示的第一類型和第二類型的音符可以相互轉換。同時，圖 2 的視覺圖形影像變成了圖 1 的具有大量括號的複雜樹。請注意，圖 1 所示類型的線條不僅可以確定音符或和弦的順序，還可以確定節奏樹，最簡單的例子如圖 3 所示。圖中的數字表示順序：數字條形的數量（一）、條形的大小（四分之四）以及將長條分成單獨的分數(2+1+1)。

圖 3. 錄製節奏樹最簡單的範例

如上所述，第二類物件包括允許您更改資訊資料的函數。我們的目的並不是要描述所有這些功能。這是不可能的，也沒有必要。由於接下來我們將討論創建音樂作品時演算法的使用，因此我們首先需要建立循環，在循環中一些資料將按順序排序。OpenMusic 中執行此操作的主要物件是 `omloop`，其外觀如圖 4 所示。

在圖 4 中，我們看到輸入 0 和輸入 1 中包含的數字相互相乘。該程式建立一個包含此類乘法結果的清單。這裡我們面臨著對資料的另一種操作，即它們通常的相互乘法。當然，加法也是有定義的。我們不會詳細討論這些操作。

圖 4. 循環的組織

3. Tristan Murail 作品 «Désintégrations» 的一部分分析

特里斯坦·穆拉伊(Tristan Murail) 在 1982-83 年以頻譜音樂風格創作的作品《Désintégrations》由 11 個部分組成，每個部分都有自己的頻譜、節奏和織體作品的特點。我們將重點解釋背後的想法第 10 部分[10]。該部分的基礎是長號音色的失真，演奏大八度的 E。音符的第 12 泛音被視為參考點。每個新和弦，它上升四分之一個音調。其餘 11 個和聲按比例拉伸。此外，這種動態過程（被耳朵感知為不和諧音的增加）會隨著時間的推移而加速。

要產生該樂曲第 10 部分的和弦序列，您可以使用以 OpenMusic 編寫的補丁。下面的圖 5 顯示了一個補丁，它允許您建立一系列逐漸受挫的和弦。

圖 5. 模擬和弦序列的 Patch

該程式從上到下運行。首先，從大八度的 E 音中提取音符右側視窗中指示的數字的泛音。事實證明，E 的音色像長號一樣。他的樂譜如下。接下來，用越來越扭曲的音色建構和弦序列。最終的和弦序列放置在圖 5 底部的 Voice 物件中。

4. 安東尼奧葛蘭西節奏理論中的代數運算

節奏通常被理解為一種重複結構。多樂器演奏家音樂家、東方鼓學院教師安東尼奧·葛蘭西正是以這種形式出現的。他對可以在鼓上敲擊的現有節奏的結構感興趣，而且對我們來說更有趣的是可以對節奏執行的操作。事實證明，諸如加法和乘法之類的運算是很自然地定義的，它們將一組節奏（我們將在下面給出精確的定義）轉變為代數結構。以節奏求冪為主要運算，我們發

現最初透明的節奏結構在時間上會變得顯著複雜。

一個簡單的節奏模式可以用一系列 1 和 0 表示，其中 1 表示節拍的存在，0 表示暫停 [11-12]。將連續零的數量（其中左邊有一個）相加，我們得到 |4221111| 形式的條目。當然，將記錄中的每個數字乘以相同的數字，我們會得到相同的節奏，因為時間單位沒有絕對值。例如，條目 |8442222| 或 |16884444| 設定一種節奏。可以在以這種形式編寫的節奏上定義大量不同的代數運算。對我們來說，它們最重要的例子是加法（串聯）和乘法。兩種節奏相加的結果就是他們一致的表現。乘法具有更複雜的結構，其結果是一種節奏在另一種節奏中的表現。從形式上來說，從數學上來說，這意味著定義節奏模式的數字的逐個分量相乘。所以 |4221111| 乘以 |211| 給出 |844422422211211211211|。

圖 6. 建構三種不同節奏同時演奏的補丁

純節奏（無音高）可以在具有 midi 號碼的各種打擊樂器上演奏。電腦程式中為他們保留了第 10 頻道，用於聲音再現。同時，可以建立不同程度的某些基本節奏，並用不同的樂器同時演奏多個包含節奏程度的曲目。當節奏提升到足夠高（3 或 5）度時，會得到最有趣的結果。我們使用 Open Music 程式來獲取此類電腦作品。下面的圖 6 顯示了一個執行 5、6 和 3 度不同節奏構建的補丁。之後，它們被組合在 Poly 物件中。

圖 7. 輸出節奏構成的程式結果

最後，這是一張樂譜工廠的圖片，展示了開放音樂計畫的結果。看圖 7，您可能認為我們看到的是普通的樂譜。事實上，這裡的音高是指演奏該節奏的 MIDI 樂器的編號，而音符聲音的持續時間只是傳達了節奏模式。

5. 將樂曲 «Eugenica III. 中的語調提升到一定程度。回饋門人» 作者：Nikolai Khrust

作曲家尼古拉·赫魯斯特熟悉了節奏代數運算理論後，提出了針對高音音樂文本的類似運算。選擇持續時間和音高作為音符的參數。當兩個音符相乘時它們的持續時間相加，高度乘以半音階的模數。這樣的運算可以與兩個節奏的複雜乘法進行比較，其中每個分量都有兩個參數。其中一個直接負責分數的值，第二個取兩個值，其中一個可以稱為有條件“聾”，第二個可以稱為“響亮”。對於具有這種乘法的節奏，採用一種規則，根據該規則，兩個聾子和兩個響亮的相乘得到聾子，否則獲得響亮的。

求冪過程中節奏模式的複雜性在乘法的新定義中令人驚訝地表現出來。

N. Khrust 的音樂作品的建構利用了可重複性原則，以便在每個新轉折處豐富內容。正是這個屬性決定了「Eugenica」這個名字的意義。下面的圖 8 顯示了在 Open Music 程式中編寫的最簡單的補丁，它執行這樣的操作。

圖 8. 兩種語調模式相乘的結果

在圖 8 中補丁頂部的視窗中，我們看到兩個簡單的旋律。在這裡，四位數字設定音符高度的順序，帶有複雜括號系統的數字確定時鐘尺寸。尼可拉·赫魯斯特的作品當然比給出的數學方案複雜得多，是在莫斯科音樂學院拉赫曼尼諾夫音樂廳的當代音樂節上演奏的[13]。

6. 結論

我們在三種不同的情況下演示了專門為滿足作曲家的需求而編寫的物件導向程式語言 OpenMusic 的功能。這是頻譜音樂，可讓您模擬各種動態變化的音色、節奏的代數運算以及從節奏到音高文字的代數運算的轉換。當然，這並沒有窮盡在創作音樂作品時使用演算法的所有可能性。結果表明，利用演算法可以獲得複雜度逐漸增加的感覺的心理聲學結果。

Литература

References

參考書目

1. Nielsen M., Chuang I. Quantum Computation and Quantum Information. 2011.
2. National Strategy for the development of Artificial Intelligence for the period up to 2030, Electronic resource.
URL: http://static.kremlin.ru/media/events/files/ru/AH4x6_HgKWANwVtMOFPDhcbRpvd1HCCsv.pdf
3. Aldoshina I., Pritts R. Musical Acoustics. 2014 (in Russian).
4. Amosov G.G., Nadzharov A.S., Khrust N.Yu. Lectures on electroacoustic music. 2017 (in Russian).
5. Denisov E.V. Modern music and the problem of the evolution of compositional technique. 1986 (in Russian).
6. Amosov G.G., Komissarenko A.V. The history of European music as a passage from classical physics to quantum mechanics // 9th International conference on music perception and cognition, Bologna, August 22-26 2006
7. Barkalaya N., Komissarenko A. Electric musical instruments in Russia in the 20th century // Academic Forum “Integral Music Theory” 2018. URL: http://integral-music-theory.com/archiv%D0%B5/academic_forum_2018.pdf (in Russian)
8. Max Mathews Radio Baton Demonstration // Electronic resource. URL:

<https://www.youtube.com/watch?v=3ZOzUVD4oLg>

9. Agon C. OpenMusic: Un langage visuel pour la composition musicale assistée par ordinateur. PhD Thesis. Paris VI. 1998.

10. Amosov G.G., Nadzharov A.S. Object-oriented programming language OpenMusic and its use for analyzing the concept of creating musical works // Neurocomputers: development, application. 2015. № 9. С. 67-76 (in Russian).

11. A. Gramsci, G. Amosov. Algebra of rhythms // Electronic resource. URL: <https://www.youtube.com/watch?v=6yMdT3LN9pI>

12. Amosov G.G., Gramsci A. Algebra of rhythms and its computer implementation in the program OpenMusic // Biomashsystems. 2018. T. 2. № 3. С. 171-183 (in Russian).

13. Five Evenings+1. Chamber Festival of the Union of Composers of Russia // Electronic resource.

URL: <https://www.unioncomposers.ru/public/upload/1/9dcae13935-booklet-2021.pdf>

14. Butsko Yu.M. The phenomenon of the mass of sound (to the question of the analysis of thematism) // Scientific Bulletin of the Moscow Conservatory. 2016. T. 7. В. 3. С. 32-45 (in Russian).

15. Turing A On Computable Numbers, with an Application to the Entscheidungsproblem // Proceedings of the London Mathematical Society. Series 2. 1937. V. 42. № 1. P. 230–265.