

УДК 004.4'2

ПРЕОБРАЗОВАНИЕ ХАРАКТЕРНЫХ ОСОБЕННОСТЕЙ ОФОРМЛЕНИЯ ПРОГРАММНОГО КОДА В МЕЛОДИЮ

Сорокин М.В., Слоцкий А.В.

к.т.н., доцент кафедры ИС Болтенков В.А.

Одесский национальный политехнический университет, УКРАИНА

АННОТАЦИЯ. В работе предложен метод преобразования особенностей оформления программных кодов в мелодическую последовательность звуков.

Введение. В процессе рассмотрения различных примеров программного кода были выделены характерные особенности, которые присущи большему количеству легко читаемых и понятных программных кодов. Эти особенности описаны в виде большого набора формальных правил [1] и было решено, что код можно описать в виде последовательности звуков, таким образом, что если код соблюдает правила «хорошего» кода, то мелодия будет звучать гармонично, а если нет – мелодия будет негармоничной.

Цель работы. Целью работы стала разработка программного обеспечения, которое преобразовывает произвольный фрагмент программного кода в набор звуков, по которым при наличии слуха можно проследить, соблюдены ли в данном фрагменте «правила хорошего кода».

Основная часть работы. Работа предложенной системы описывается следующим образом: пользователь передает в специальное поле интересующий его участок кода. Система отслеживает характерные особенности этого участка. Затем на основе выделенных особенностей и теории музыкальной гармонии строится мелодия, которая будет звучать гармонично, или негармонично в зависимости от выделенных особенностей.

Основными характеристиками, присущими любому коду вне зависимости от языка и среды программирования, были выбраны форматирование и длина строки.

Задача, решаемая системой, состоит из следующих подзадач: 1) определение длины табуляции, присущей конкретному программному коду, и устранение табуляции с помощью не пробельных символов; 2) составление музыкальной последовательности на основе длины табуляции и длины строк.

В процессе определения длины табуляции возникли проблемы, так как все примеры программного кода можно разделить на 3 группы: 1) табулированный с помощью символа табуляции; 2) табулированный с помощью пробельного символа; 3) табулированный и с помощью символов табуляции и пробельных символов. В первом случае все символы табуляции заменялись на одинаковую последовательность пробельных символов (U). Во втором случае никаких действий по замене символов не производилось, но нужно подсчитать самую часто встречающуюся разницу количества пробелов между соседними строками (U). В третьем случае нужно заменить символы табуляции на некоторое количество пробельных символов, при этом не нарушив табуляцию, которая возможно уже содержится в данном участке кода. Решением было находить последовательность из как минимум 2-х строк, не содержащих символов табуляции между началом строки и первым непробельным символом. Сохраняются все разницы и находится среди них самое часто встречающееся значение (U). Далее - заменяются все символы табуляции на это значение.

В результате были получены формализованные наборы признаков по табуляции и длине строки, которые и использовались для создания мелодии.

Мелодия состоит из звуков, характеристиками которых служит высота и длительность; в качестве составляющих композиции могут использоваться ноты (один звук), аккорды (три или более звука, звучащих гармонично), интервалы (два звука разной высоты). Для создания мелодий из программного кода в качестве составляющей единицы было решено использовать арпеджио – способ воспроизведения аккорда, заключающийся в последовательном извлечении

звуків аккорда. Аккорды имеют основной тон (тонику) – звук, на основе которого строятся интервалы, составляющие аккорд. Полутон – наименьший интервал (разница высоты между двумя звуками).

С целью создать музыкальную последовательность, которая гармонично звучала бы для корректно оформленного программного кода, и дисгармонично для некорректно оформленного программного кода, были выбраны критерии, определяющие верную табуляцию и длительность аккорда.

На основе глубины табуляции выбирается основной тон проигрываемого аккорда : $I U$ (длина табуляции в пробелах) = I ступень. Так, аккорды корректно оформленного программного кода образуют мажорную гармоническую последовательность: каждый уровень табуляции – следующая ступень мажорной гаммы. Если размер табуляции в пробелах не кратен U , тоника построенного аккорда смещается на количество полутонов, равное остатку от деления на U . Это значит, что если в программном коде присутствует неравномерный размер табуляции, то аккордовая последовательность будет звучать в диссонанс (негармонично).

Пример работы программы приведен на рис.1.

```

13   var pack = function(c,arg){ return [new Uint8Array([arg, arg >> 8]), new
14   var setPrivateVar = function(n,v,w,e){Object.defineProperty(this,n,{value
15   var setPublicVar = function(n,v,w){setPrivateVar.call(this,n,v,w,true)};
16   AudioSynthInstrument = function AudioSynthInstrument(){this.__init__.appl
17   var setPriv = setPrivateVar.bind(AudioSynthInstrument.prototype);
18   var setPub = setPublicVar.bind(AudioSynthInstrument.prototype);
19   setPriv('__init__', function(a,b,c) {
20   if(!encapsulated) { throw new Error('AudioSynthInstrument can only be
21   setPrivateVar.call(this, '_parent', a);
22   setPublicVar.call(this, 'name', b);
23   setPrivateVar.call(this, '_soundID', c);
24   });

```

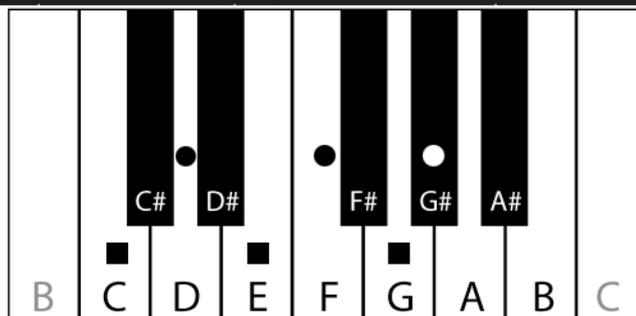


Рис 1. – Аккорды корректной и некорректной табуляции. 1). квадратная метка – пример правильной табуляции. 2). Круглая метка – пример неправильной табуляции.

Для задания длительности нот арпеджио аккорда, используется рекомендация по оформлению кода, которая гласит, что строка кода должна иметь длину не более 80 символов. Если количество символов в строке кода от 0 до 40 и от 40 до 80 – используется последовательность длительностей в сумме образующая целый такт (цельная структурная единица музыкального произведения), а если больше 80 – переполненный такт.

Таким образом, если соблюдены вышестоящие правила, то из программного кода получается ритмическая мажорная мелодия. Если же в коде присутствуют некоторые нарушения правил, то они будут слышны в виде «рваного» ритма и «фальшивых» нот.

Выводы. Разработана программа, преобразующая фрагменты программного кода в мелодию, позволяющую судить о правильности написания кода.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. С. Макконнел. Совершенный код. Практическое руководство по разработке программного обеспечения. // — М. : Издательско-торговый дом «Русская Редакция» – 2010 – С.713-747.