

Stefan Sokołowski

PROGRAMOWANIE OBIEKTOWE II

wykład 15

**Inst. Informatyki Stosowanej, ANS
Elbląg, 2024/2025**

Muzyka w Jawie

program grający
plik dźwiękowy .wav

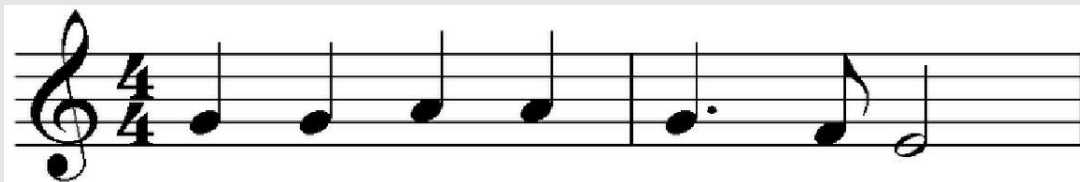


Muzyka w Jawie

program grający
plik dźwiękowy .wav



program grający z nut

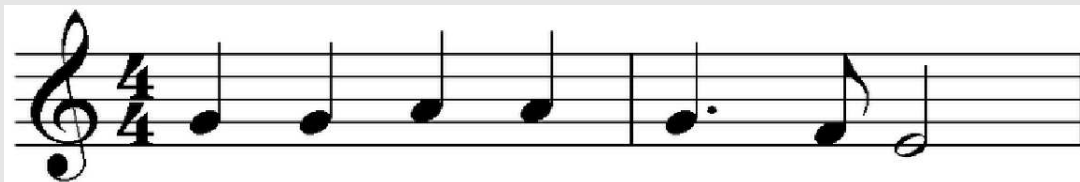


Muzyka w Jawie

program grający
plik dźwiękowy .wav



program grający z nut



program realizujący
„organy elektroniczne”

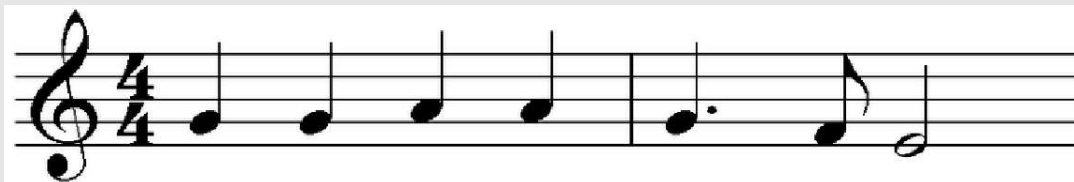


Muzyka w Jawie

program grający
plik dźwiękowy .wav



program grający z nut



program realizujący
„organy elektroniczne”



akordy i efekty



Dane z pliku

import pakietów	dostęp do pliku	użycie danych

Dane z pliku

	import pakietów	dostęp do pliku	użycie danych
tekst	<pre>java.io.* java.util.Scanner</pre>	<pre>File f = new File("tekst.txt"); Scanner czyt = new Scanner(f);</pre>	<pre>while (czyt.hasNextLine()) System.out.println(" " + czyt.nextLine());</pre>

Dane z pliku

	import pakietów	dostęp do pliku	użycie danych
tekst	<pre>java.io.* java.util.Scanner</pre>	<pre>File f = new File("tekst.txt"); Scanner czyt = new Scanner(f);</pre>	<pre>while (czyt.hasNextLine()) System.out.println(" " + czyt.nextLine());</pre>
obrazek	<pre>java.io.* javax.imageio.* javax.swing.* java.awt.* java.awt.image.*</pre>	<pre>File f = new File("obraz.png"); BufferedImage obr = ImageIO.read(f);</pre>	<pre>JFrame okno = new JFrame(); okno.add(new JComponent() { public void paintComponent(Graphics g) { g.drawImage(obr, 0, 0, null); } });</pre>

Dane z pliku

	import pakietów	dostęp do pliku	użycie danych
tekst	<pre>java.io.* java.util.Scanner</pre>	<pre>File f = new File("tekst.txt"); Scanner czyt = new Scanner(f);</pre>	<pre>while (czyt.hasNextLine()) System.out.println(" " + czyt.nextLine());</pre>
obrazek	<pre>java.io.* javax.imageio.* javax.swing.* java.awt.* java.awt.image.*</pre>	<pre>File f = new File("obraz.png"); BufferedImage obr = ImageIO.read(f);</pre>	<pre>JFrame okno = new JFrame(); okno.add(new JComponent() { public void paintComponent(Graphics g) { g.drawImage(obr, 0, 0, null); } });</pre>
muzyka	<pre>java.io.* javax.sound.sampled.*</pre>	<pre>File f = new File("muzyka.wav"); Clip clip = AudioSystem.getClip(); clip.open(AudioSystem. getAudioInputStream(f));</pre>	<pre>clip.start(); clip.drain();</pre>

Dane z pliku

	import pakietów	dostęp do pliku	użycie danych
tekst	<pre>java.io.* java.util.Scanner</pre>	<pre>File f = new File("tekst.txt"); Scanner czyt = new Scanner(f);</pre>	<pre>while (czyt.hasNextLine()) System.out.println(" " + czyt.nextLine());</pre>
obrazek	<pre>java.io.* javax.imageio.* javax.swing.* java.awt.* java.awt.image.*</pre>	<pre>File f = new File("obraz.png"); BufferedImage obr = ImageIO.read(f);</pre>	<pre>JFrame okno = new JFrame(); okno.add(new JComponent() { public void paintComponent(Graphics g) { g.drawImage(obr, 0, 0, null); } });</pre>
muzyka	<pre>java.io.* javax.sound.sampled.*</pre>	<pre>File f = new File("muzyka.wav"); Clip clip = AudioSystem.getClip(); clip.open(AudioSystem. getAudioInputStream(f));</pre>	<pre>clip.start(); clip.drain();</pre>

Z pliku dźwiękowego



```
import java.io.*;
import javax.sound.sampled.*;

class C3_Muzyka {
    public static void main(String[] args) throws Exception {
        File f = new File("muzyka.wav");
        Clip clip = AudioSystem.getClip();
        clip.open(AudioSystem.getAudioInputStream(f));
        clip.start();  clip.drain();
    }
}
```

Z pliku dźwiękowego



```
import java.io.*; ← obsługa plików
import javax.sound.sampled.*;
```

```
class C3_Muzyka {
    public static void main(String[] args) throws Exception {
        File f = new File("muzyka.wav");
        Clip clip = AudioSystem.getClip();
        clip.open(AudioSystem.getAudioInputStream(f));
        clip.start(); clip.drain();
    }
}
```

Z pliku dźwiękowego



```
import java.io.*; ← obsługa plików  
import javax.sound.sampled.*; ← pakiet dźwiękowy
```

```
class C3_Muzyka {  
    public static void main(String[] args) throws Exception {  
        File f = new File("muzyka.wav");  
        Clip clip = AudioSystem.getClip();  
        clip.open(AudioSystem.getAudioInputStream(f));  
        clip.start();    clip.drain();  
    }  
}
```

Z pliku dźwiękowego



```
import java.io.*; ← obsługa plików
import javax.sound.sampled.*; ← pakiet dźwiękowy

class C3_Muzyka {
    public static void main(String[] args) throws Exception {
        File f = new File("muzyka.wav"); ← plik dźwiękowy
        Clip clip = AudioSystem.getClip();
        clip.open(AudioSystem.getAudioInputStream(f));
        clip.start(); clip.drain();
    }
}
```

Z pliku dźwiękowego



```
import java.io.*; ← obsługa plików
import javax.sound.sampled.*; ← pakiet dźwiękowy

class C3_Muzyka {
    public static void main(String[] args) throws Exception {
        File f = new File("muzyka.wav"); ← plik dźwiękowy
        Clip clip = AudioSystem.getClip(); ← „kawałek”
        clip.open(AudioSystem.getAudioInputStream(f));
        clip.start(); clip.drain();
    }
}
```

Z pliku dźwiękowego



```
import java.io.*; ← obsługa plików
import javax.sound.sampled.*; ← pakiet dźwiękowy

class C3_Muzyka {
    public static void main(String[] args) throws Exception {
        File f = new File("muzyka.wav"); ← plik dźwiękowy
        Clip clip = AudioSystem.getClip(); ← „kawałek”
        clip.open(AudioSystem.getAudioInputStream(f));
        clip.start(); clip.drain();
    }
}
```

połączyć plik z systemem grającym

Z pliku dźwiękowego



```
import java.io.*; ← obsługa plików
import javax.sound.sampled.*; ← pakiet dźwiękowy

class C3_Muzyka {
    public static void main(String[] args) throws Exception {
        File f = new File("muzyka.wav"); ← plik dźwiękowy
        Clip clip = AudioSystem.getClip(); ← „kawałek”
        clip.open(AudioSystem.getAudioInputStream(f));
        clip.start(); clip.drain();
    }
} ← grać
```

połączyć plik z systemem grającym

Z pliku dźwiękowego



```
import java.io.*; ← obsługa plików
import javax.sound.sampled.*; ← pakiet dźwiękowy

class C3_Muzyka {
    public static void main(String[] args) throws Exception {
        File f = new File("muzyka.wav"); ← plik dźwiękowy
        Clip clip = AudioSystem.getClip(); ← „kawałek”
        clip.open(AudioSystem.getAudioInputStream(f));
        clip.start(); clip.drain();
    }
} ← grać
```

połączyć plik z systemem grającym

Uwaga:

Tak można grać pliki .wav . Dla np. .mp3 potrzebne są inne sposoby.

Z pliku dźwiękowego



```
import java.io.*;      ← obsługa plików
import javax.sound.sampled.*; ← pakiet dźwiękowy

class C3_Muzyka {
    public static void main(String[] args) throws Exception {
        File f = new File("muzyka.wav"); ← plik dźwiękowy
        Clip clip = AudioSystem.getClip(); ← „kawałek”
        clip.open(AudioSystem.getAudioInputStream(f));
        clip.start(); clip.drain();
    }
}

      ← grać
      ← połączyć plik z systemem grającym
```

Uwaga:

Tak można grać pliki .wav . Dla np. .mp3 potrzebne są inne sposoby.

Ale muzyka z gotowego pliku to w ogóle najprostszy sposób grania...

Z pliku dźwiękowego



```
import java.io.*;      ← obsługa plików
import javax.sound.sampled.*; ← pakiet dźwiękowy

class C3_Muzyka {
    public static void main(String[] args) throws Exception {
        File f = new File("muzyka.wav"); ← plik dźwiękowy
        Clip clip = AudioSystem.getClip(); ← „kawałek”
        clip.open(AudioSystem.getAudioInputStream(f));
        clip.start(); clip.drain();
    }
}
      ← grać
      ← połączyć plik z systemem grającym
```

Uwaga:

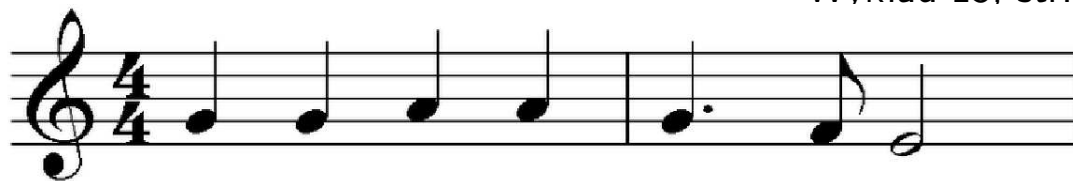
Tak można grać pliki .wav . Dla np. .mp3 potrzebne są inne sposoby.

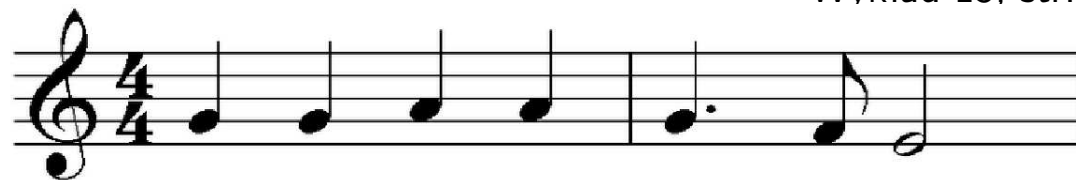
Ale muzyka z gotowego pliku to w ogóle najprostszy sposób grania...

Teraz zajmiemy się bardziej skomplikowanymi.

Z nut

Wykład 15, str. 4

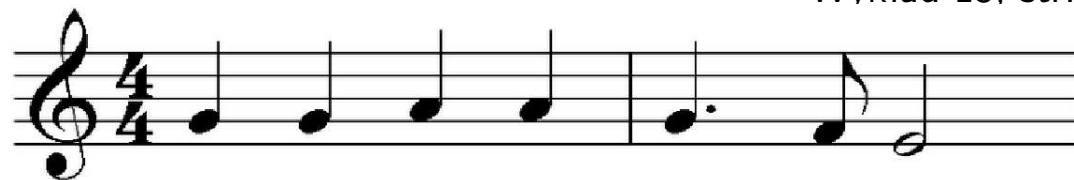


Z nut

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Dzwiek.java

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Nuta.java

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/WeShallOvercome.java

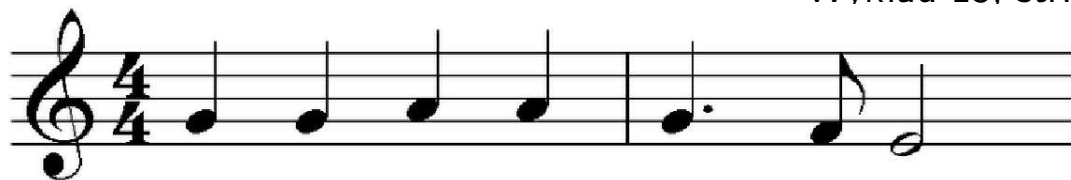
Z nut

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Dzwiek.java

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Nuta.java

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/WeShallOvercome.java

Z nut



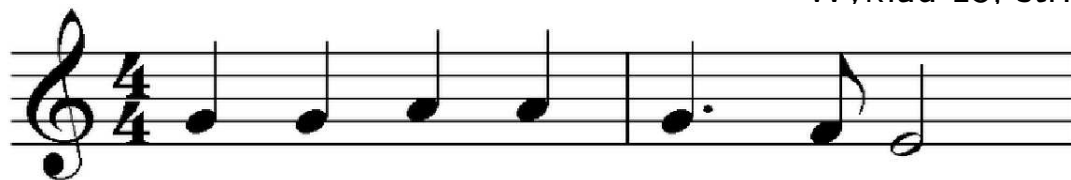
iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Dzwiek.java

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Nuta.java

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/WeShallOvercome.java

```
class WeShallOvercome {  
    public static void main(String[] args) {  
        .....  
        Nuta.G.graj(0.75);  
        Nuta.F.graj(0.25);  
        Nuta.E.graj(1.0);  
        .....  
    }  
}
```

Z nut



iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Dzwiek.java

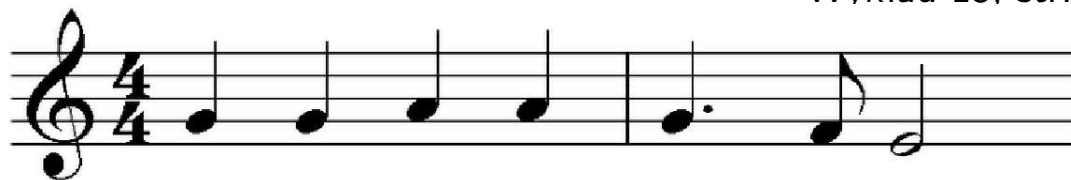
iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Nuta.java

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/WeShallOvercome.java

```
class WeShallOvercome {  
    public static void main(String[] args) {  
        .....  
        Nuta.G.graj(0.75);  
        Nuta.F.graj(0.25);  
        Nuta.E.graj(1.0);  
        .....  
    }  
}
```

nazwa nuty

Z nut



iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biIII/Prog15/Dzwiek.java

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biIII/Prog15/Nuta.java

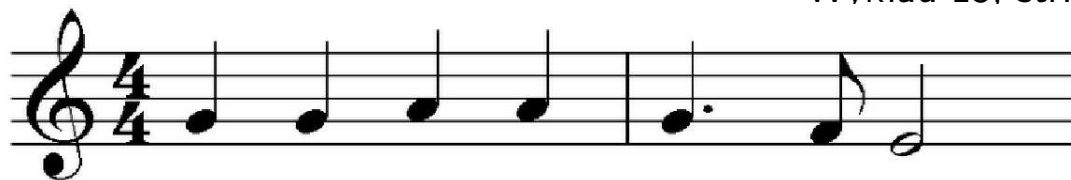
iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biIII/Prog15/WeShallOvercome.java

```
class WeShallOvercome {
    public static void main(String[] args) {
        .....
        Nuta.G.graj(0.75);
        Nuta.F.graj(0.25);
        Nuta.E.graj(1.0);
        .....
    }
}
```

Diagram illustrating the mapping of musical notes to code:

- Red arrows point from the note names **G**, **F**, and **E** in the code to the label **długość trwania** (duration).
- A red arrow points from the note name **E** in the code to the label **nazwa nuty** (note name).

Z nut



iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Dzwiek.java

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Nuta.java

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/WeShallOvercome.java

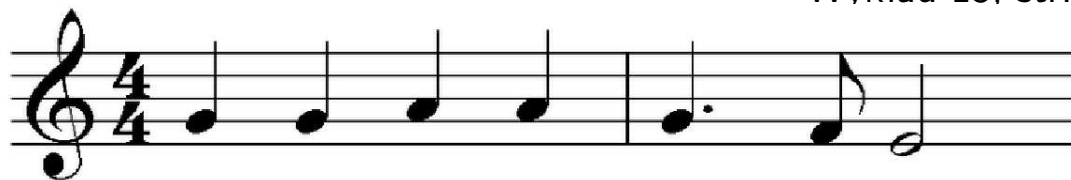
```
class WeShallOvercome {
    public static void main(String[] args) {
        .....
        Nuta.G.graj(0.75);
        Nuta.F.graj(0.25);
        Nuta.E.graj(1.0);
        .....
    }
}
```

Diagram illustrating the mapping of musical notes to code execution:

- Red arrows point from the notes G, F, and E in the code to the label "długość trwania" (duration), indicating that the value passed to `graj()` represents the duration of the note.
- A red arrow points from the letter 'E' in `Nuta.E` to the label "nazwa nuty" (note name), indicating that the letter represents the pitch of the note.

Jak zbudowana jest klasa `Nuta` ?

Z nut



iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Dzwiek.java
iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Nuta.java
iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/WeShallOvercome.java

```

class WeShallOvercome {
    public static void main(String[] args) {
        .....
        Nuta.G.graj(0.75);
        Nuta.F.graj(0.25);
        Nuta.E.graj(1.0);
        .....
    }
}

```

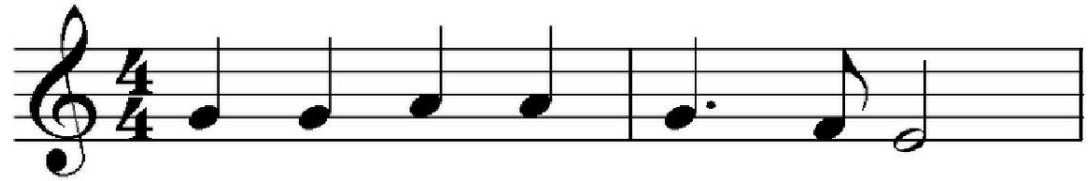
Diagram illustrating the mapping of musical notes to code:

- Red arrows point from the notes G, F, and E in the code to the label "nazwa nuty" (note name).
- Red arrows point from the numerical values 0.75, 0.25, and 1.0 in the code to the label "długość trwania" (duration).

Jak zbudowana jest klasa **Nuta** ?

Co robi metoda **graj** ?

Z nut

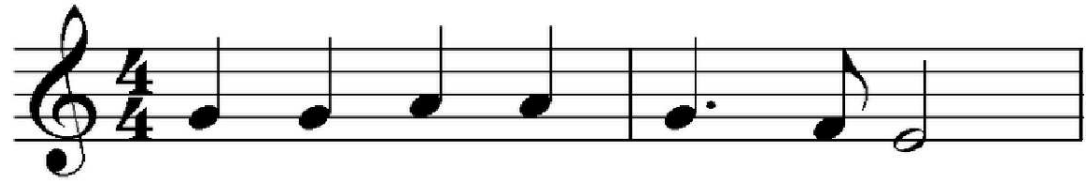


```
enum Nuta {
    C, Cis, D, Dis, E, F, Fis, G, Gis, A, Ais, H, C2;

    private Fcja naDzwiek() {
        double herce;
        switch (this) {
            case C      : herce = 261.6; break;
            case Cis    : herce = 277.2; break;
            .....
            case C2     : herce = 523.2; break;
            default    : herce = 0; break;
        }
        return x -> { return Math.sin(2*Math.PI * herce * x); };
    }

    public void graj(double czas) {
        Dzwiek.graj(naDzwiek());
        try { Thread.sleep((int)(czas*1000)); } catch(Exception e) {}
        Dzwiek.nieGraj();
    }
}
```

Z nut

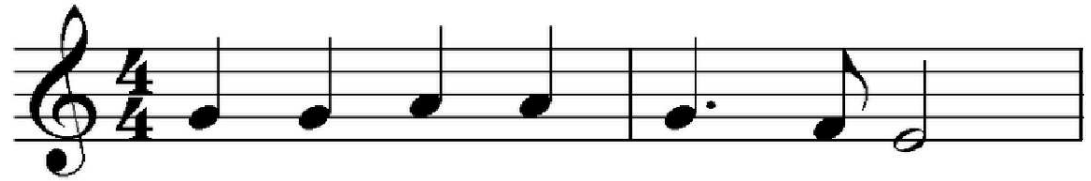


```
enum Nuta {
    C, Cis, D, Dis, E, F, Fis, G, Gis, A, Ais, H, C2;

    private Fcja naDzwiek() {
        double herce;
        switch (this) {
            case C      : herce = 261.6; break;
            case Cis    : herce = 277.2; break;
            .....
            case C2     : herce = 523.2; break;
            default    : herce = 0; break;
        }
        return x -> { return Math.sin(2*Math.PI * herce * x); };
    }

    public void graj(double czas) {
        Dzwiek.graj(naDzwiek());
        try { Thread.sleep((int)(czas*1000)); } catch(Exception e) {}
        Dzwiek.nieGraj();
    }
}
```

Z nut



```
enum Nuta {
    C, Cis, D, Dis, E, F, Fis, G, Gis, A, Ais, H, C2;

    private Fcja naDzwiek() {
        double herce;
        switch (this) {
            case C      : herce = 261.6; break;
            case Cis    : herce = 277.2; break;
            .....
            case C2     : herce = 523.2; break;
            default     : herce = 0; break;
        }
        return x -> { return Math.sin(2*Math.PI * herce * x); };
    }

    public void graj(double czas) {
        Dzwiek.graj(naDzwiek());
        try { Thread.sleep((int)(czas*1000)); } catch(Exception e) {}
        Dzwiek.nieGraj();
    }
}
```

Z nut

Dzwiek.graj(*funkcja*) — gra wg tej funkcji

Z nut

Dźwięk.graj(*funkcja*) — gra wg tej funkcji

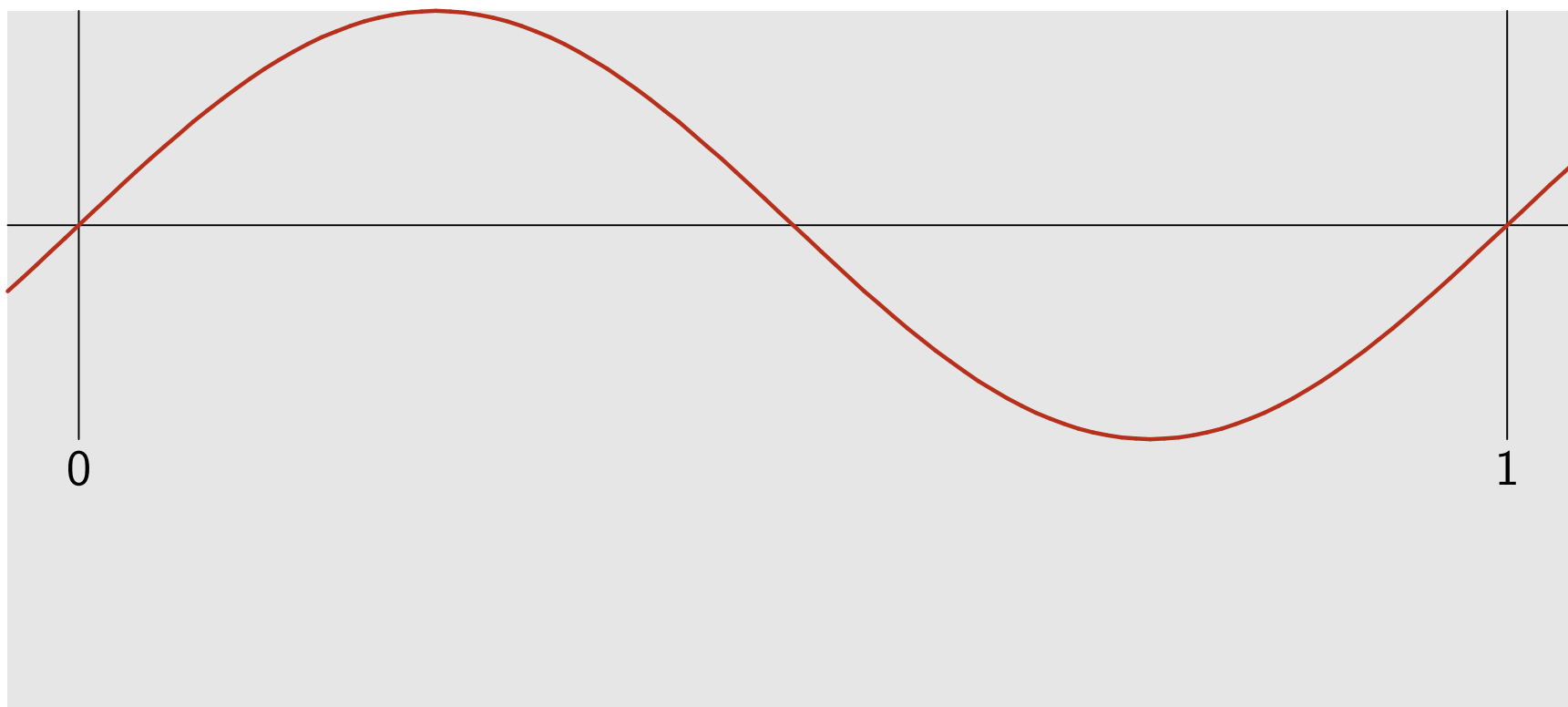
Dźwięk.nieGraj() — kończy granie

Z nut

Dzwiek.graj(*funkcja*) — gra wg tej funkcji

Dzwiek.nieGraj() — kończy granie

Granie wg funkcji:

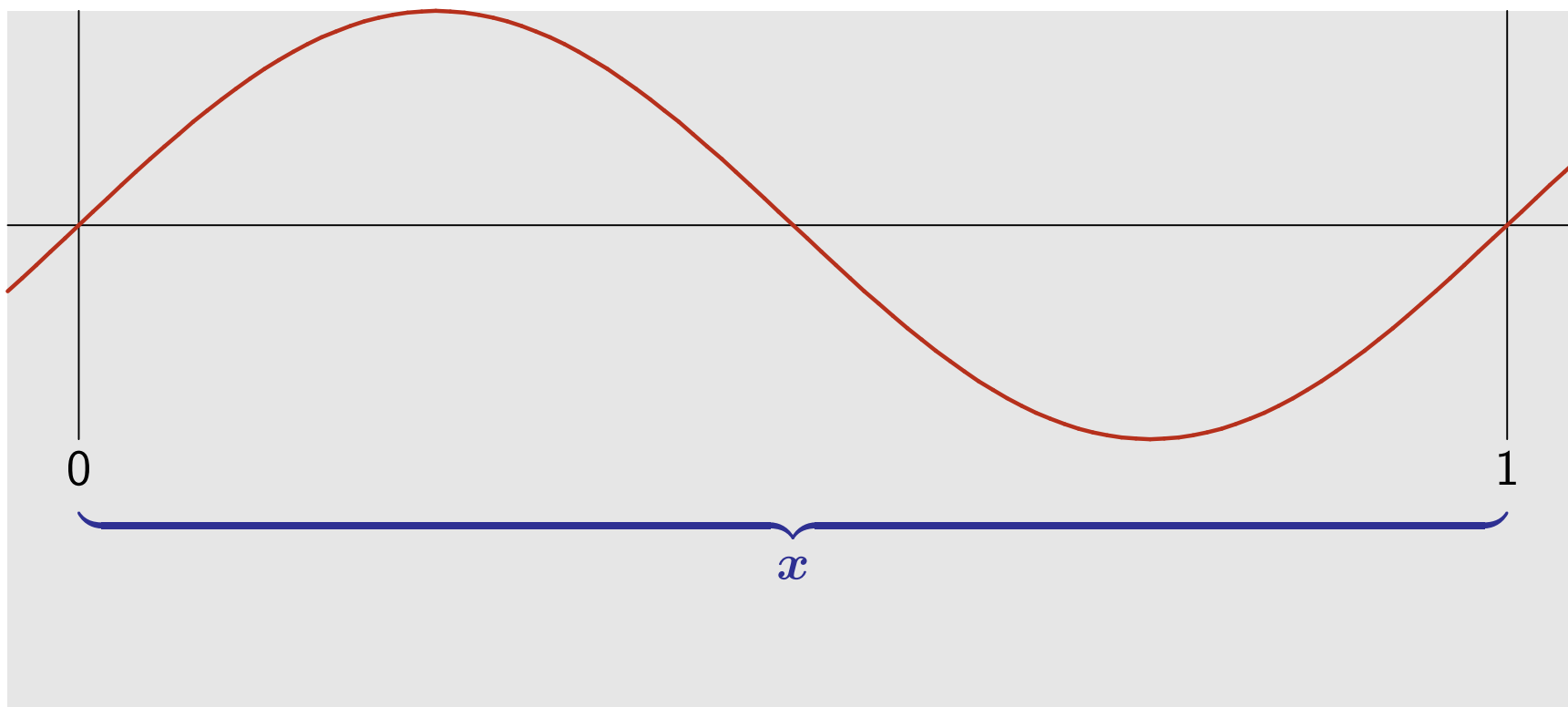


Z nut

Dzwiek.graj(*funkcja*) — gra wg tej funkcji

Dzwiek.nieGraj() — kończy granie

Granie wg funkcji:

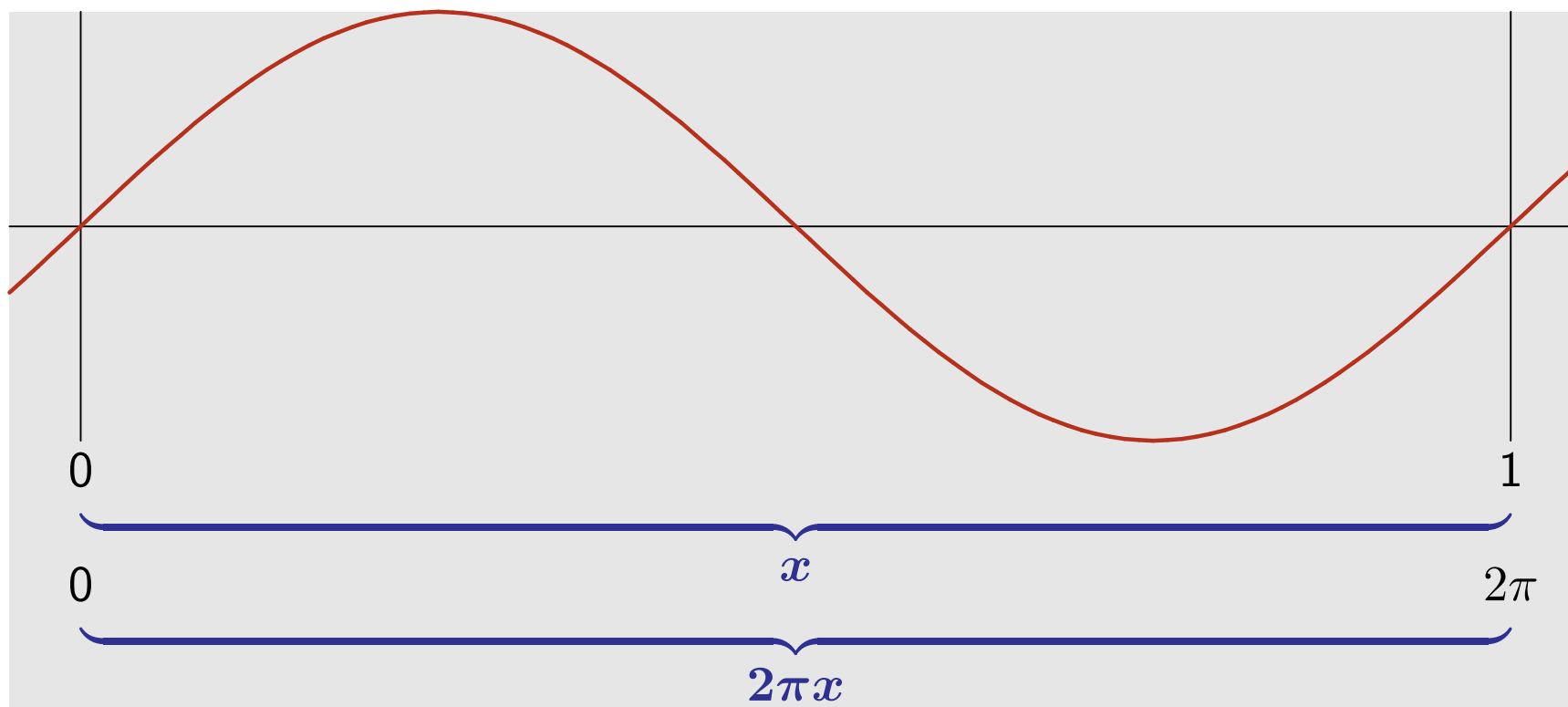


Z nut

Dźwięk.graj(*funkcja*) — gra wg tej funkcji

Dźwięk.nieGraj() — kończy granie

Granie wg funkcji:

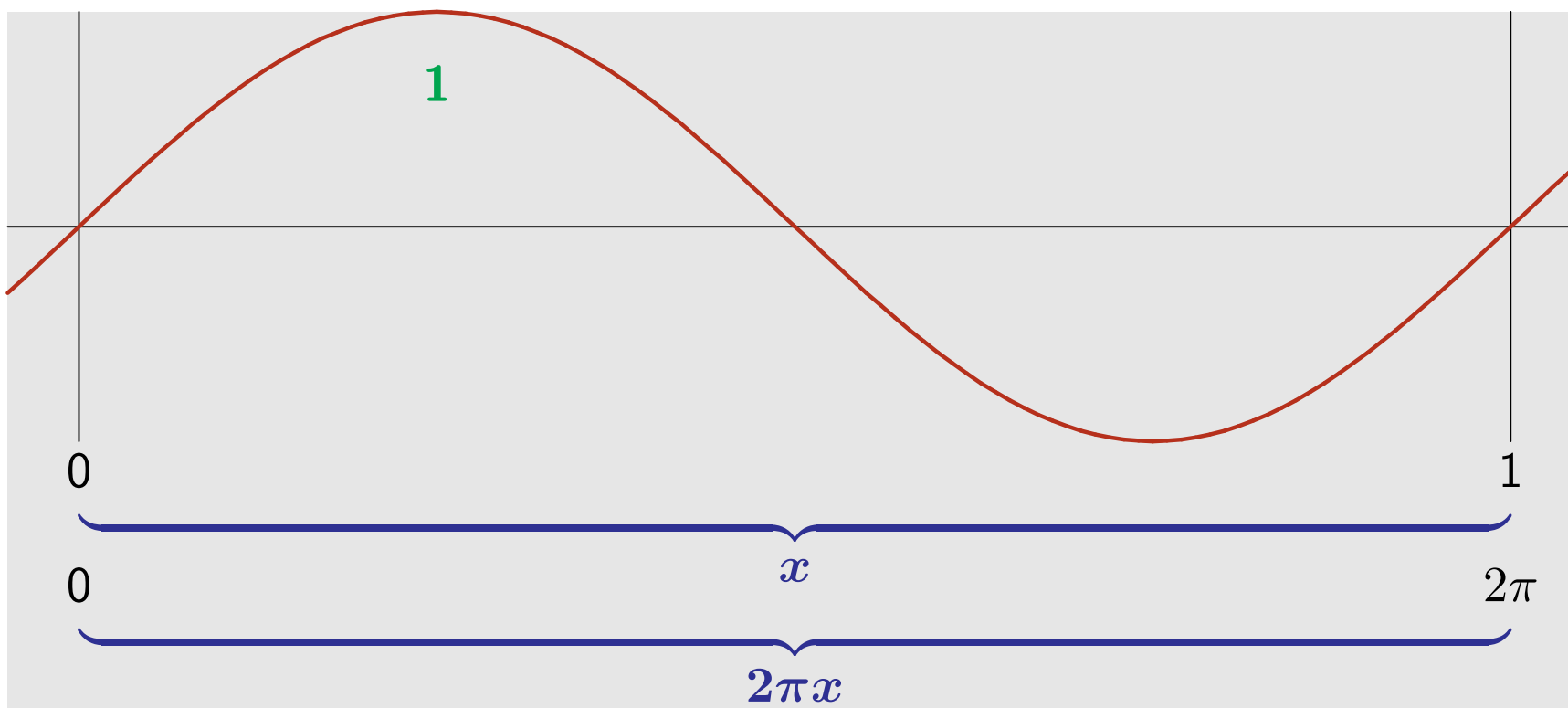


Z nut

Dzwiek.graj(*funkcja*) — gra wg tej funkcji

Dzwiek.nieGraj() — kończy granie

Granie wg funkcji:



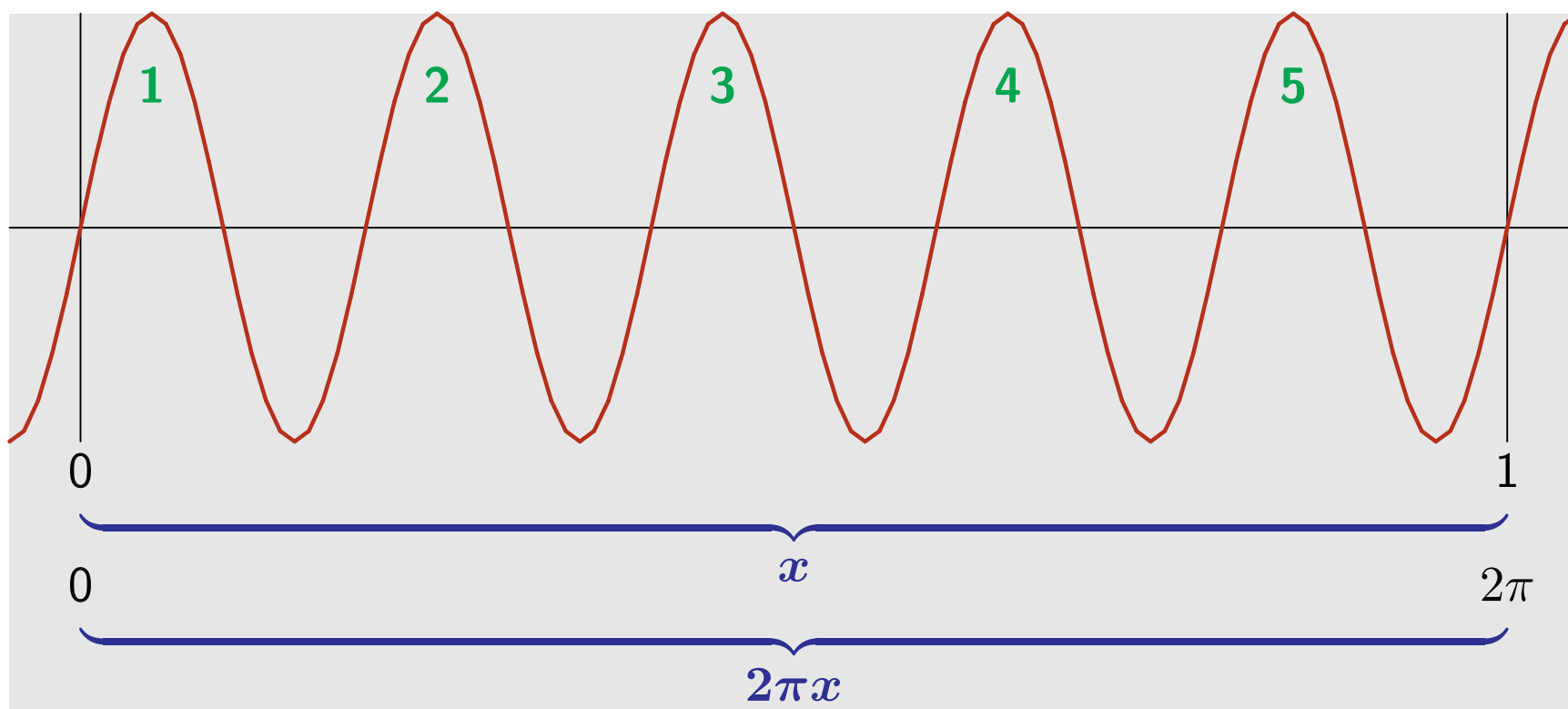
$$f(x) = \sin(2\pi x \cdot \nu) \quad \nu \text{ — częstotliwość (tutaj 1 Hz)}$$

Z nut

Dźwięk.graj(*funkcja*) — gra wg tej funkcji

Dźwięk.nieGraj() — kończy granie

Granie wg funkcji:



$$f(x) = \sin(2\pi x \cdot \nu) \quad \nu \text{ — częstotliwość (tutaj 5 Hz)}$$

Z nut

Komenda

```
Dzwiek.graj(  
  x -> { return Math.sin(2*Math.PI * herce * x); }  
)
```



gra wg tej funkcji, aż zostanie zatrzymana.

Z nut

Komenda

```
Dzwiek.graj(  
  x -> { return Math.sin(2*Math.PI * herce * x); }  
)
```



gra wg tej funkcji, aż zostanie zatrzymana.

Komenda

```
Dzwiek.nieGraj()
```

zatrzymuje granie.

Z nut

Komenda

```
Dzwiek.graj(  
  x -> { return Math.sin(2*Math.PI * herce * x); }  
)
```



gra wg tej funkcji, aż zostanie zatrzymana.

Komenda

```
Dzwiek.nieGraj( )
```

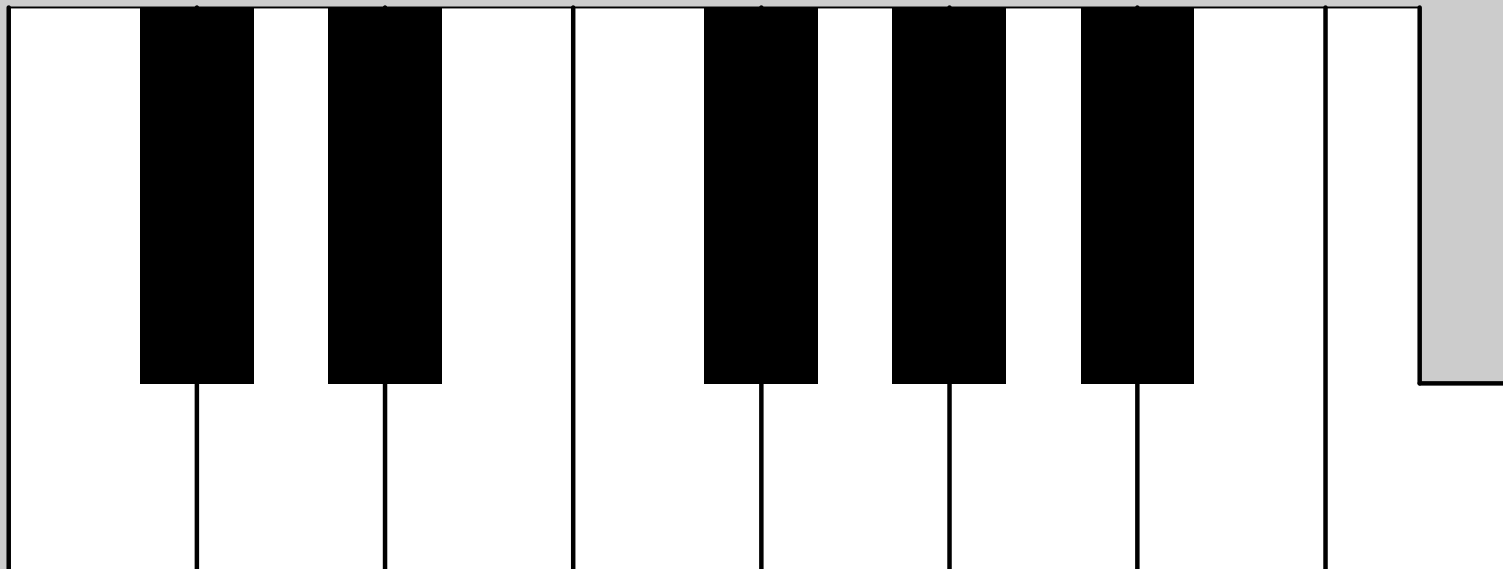
zatrzymuje granie.

To daje szansę na zaprogramowanie **organów**, w których dźwięk z naciśniętego klawisza ma trwać ciągle, aż puścimy klawisz.

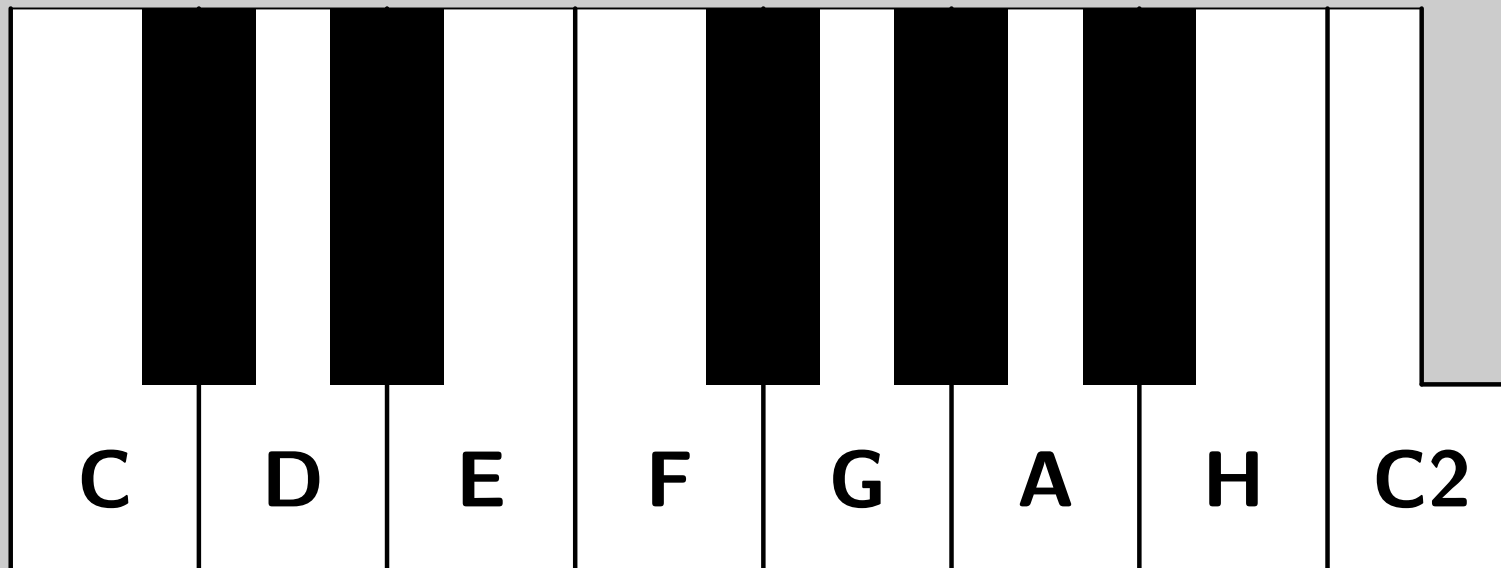
Na organach



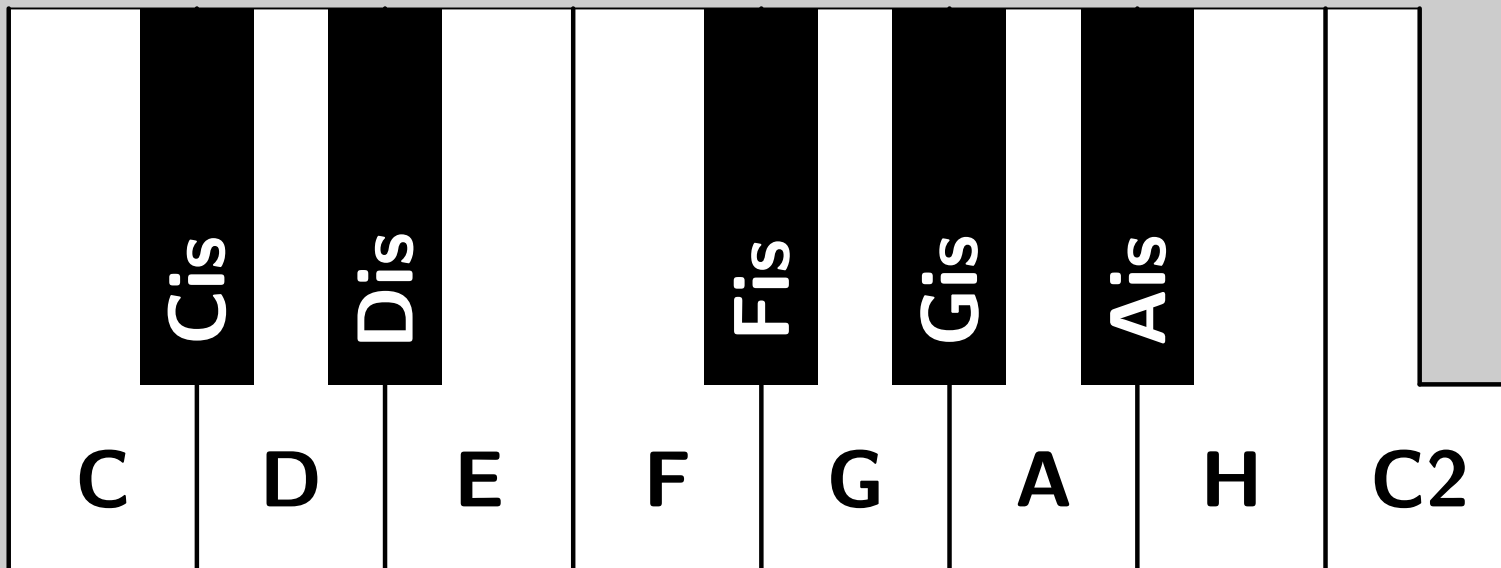
Na organach



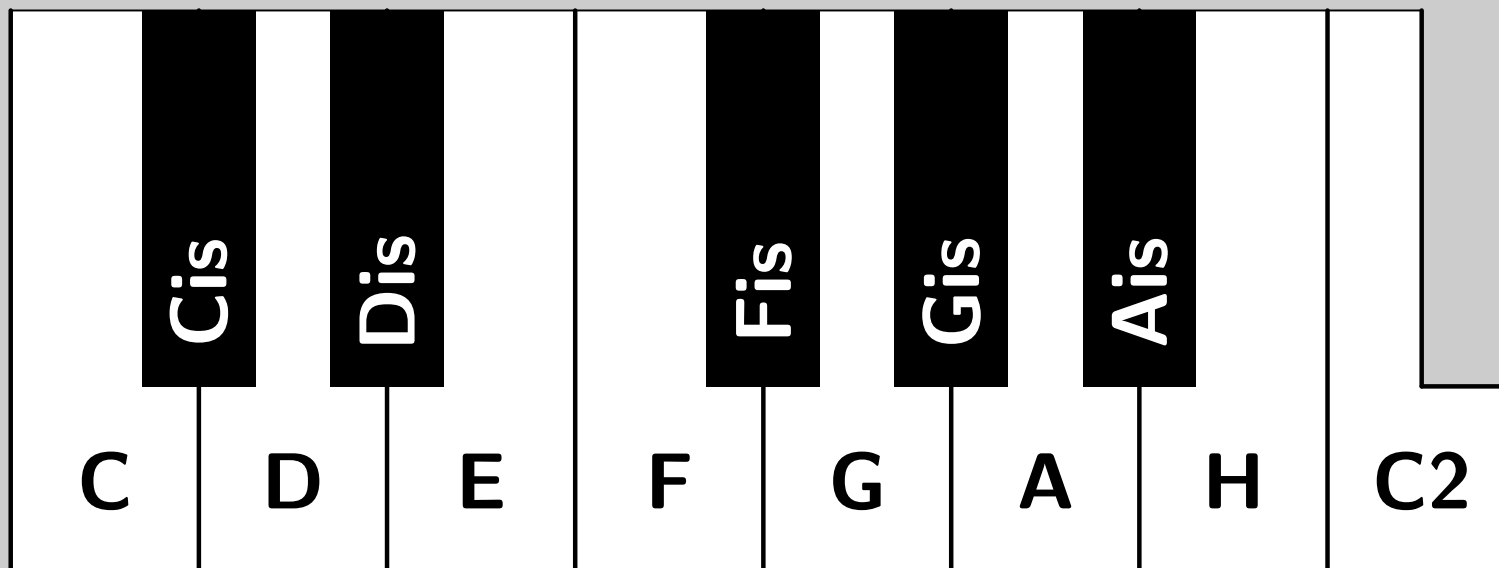
Na organach



Na organach

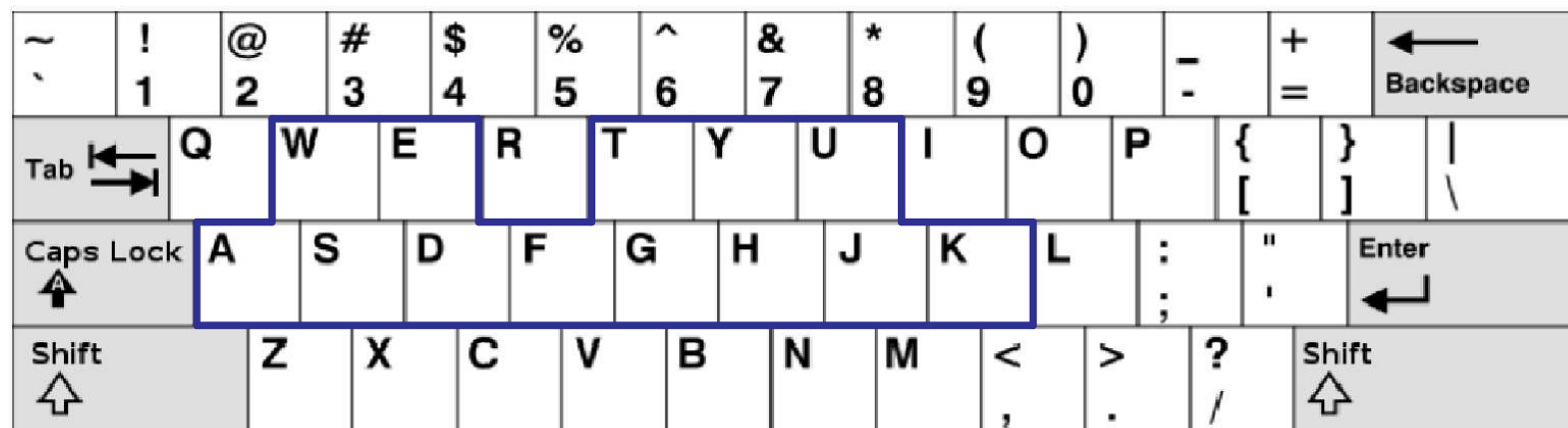
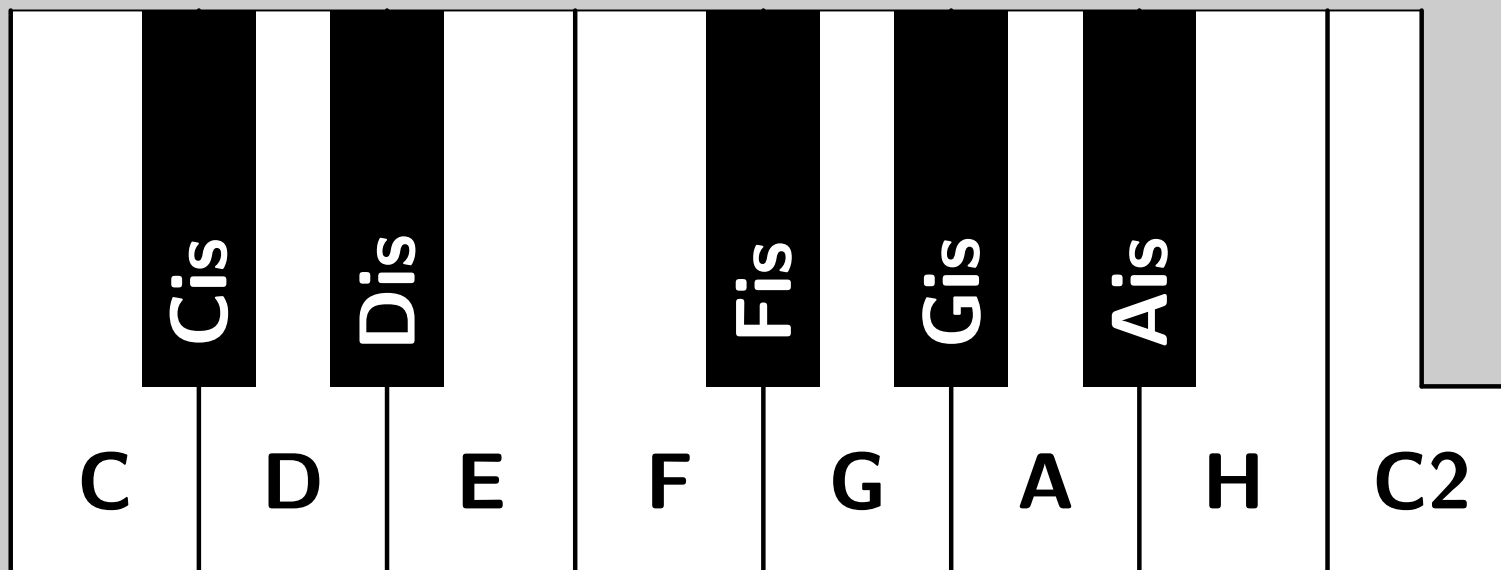


Na organach

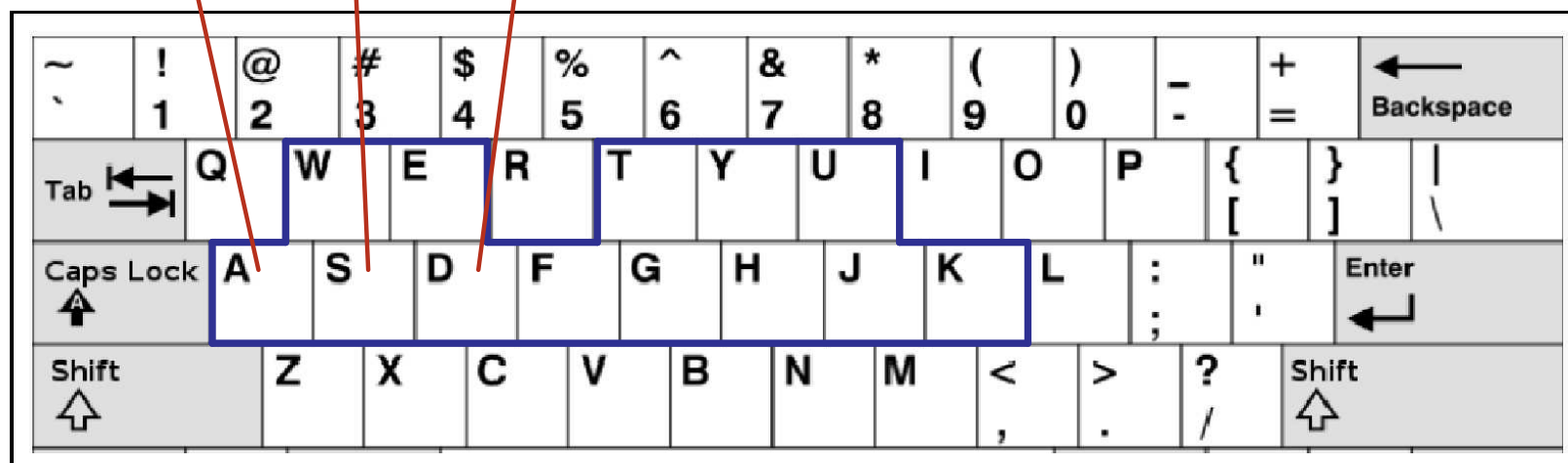
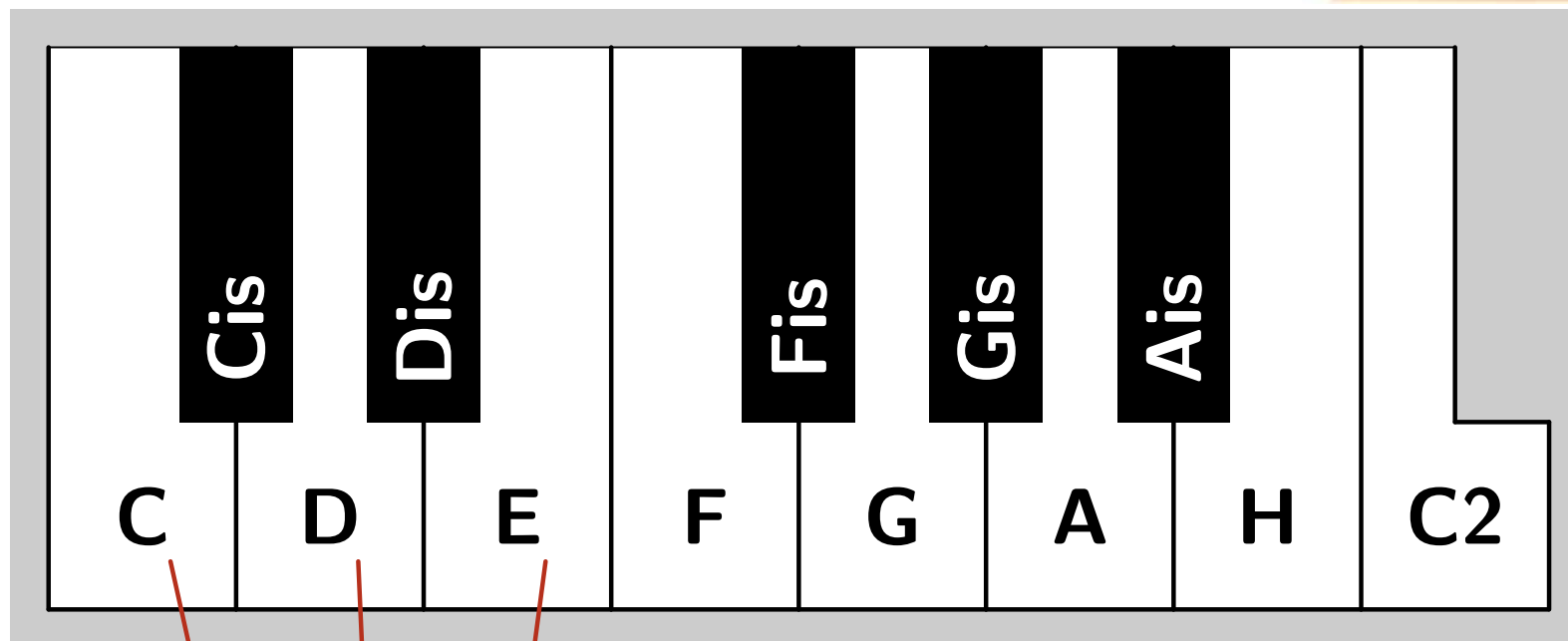


~ `	! 1	@ 2	# 3	\$ 4	% 5	^ 6	& 7	* 8	(9	) 0	- _	+ =	← Backspace
Tab ⇐ ⇒	Q	W	E	R	T	Y	U	I	O	P	{ [	}]	 \ /
Caps Lock ⬆	A	S	D	F	G	H	J	K	L	: ;	" '	Enter ↵	
Shift ⬆	Z	X	C	V	B	N	M	< ,	> .	? /	Shift ⬆		

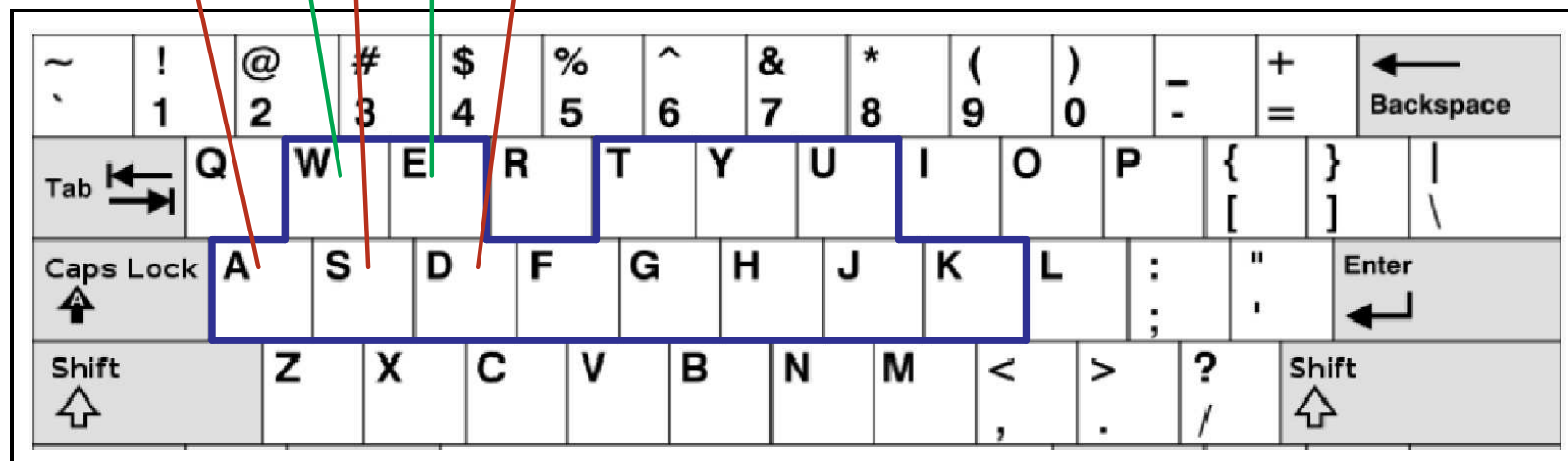
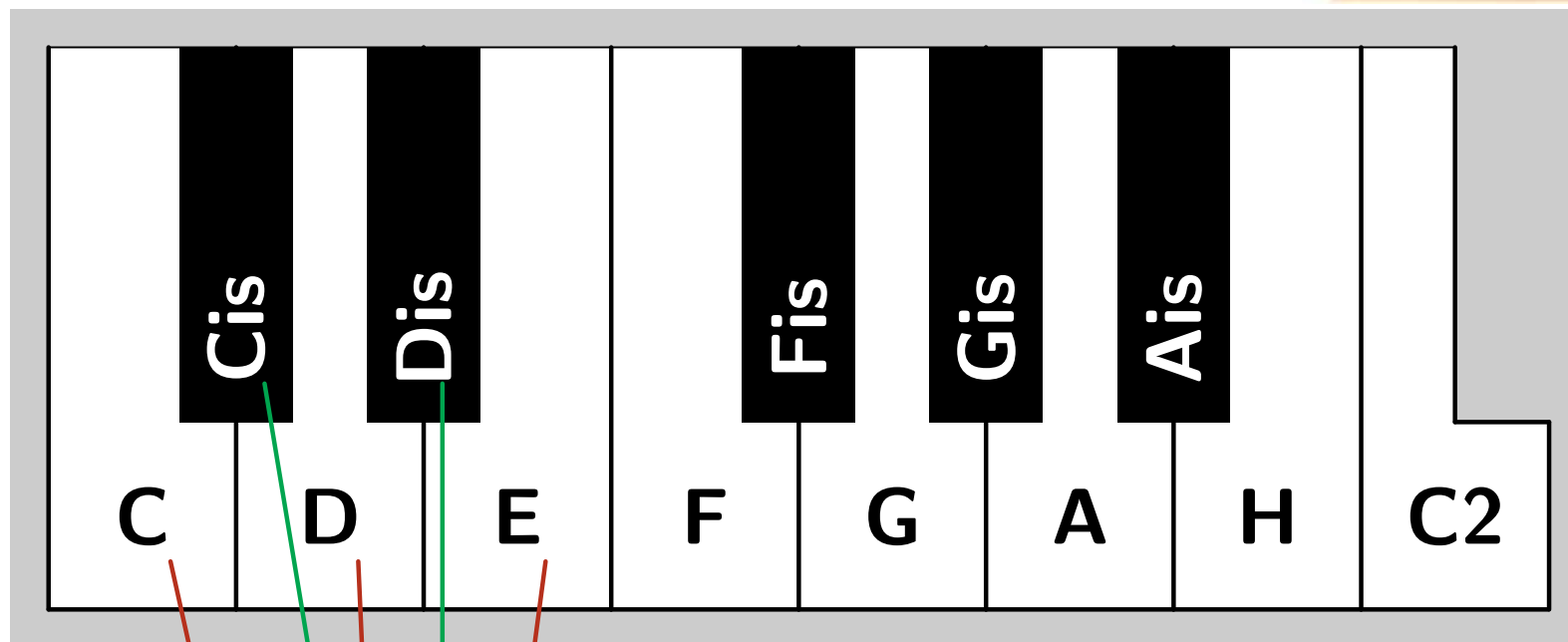
Na organach



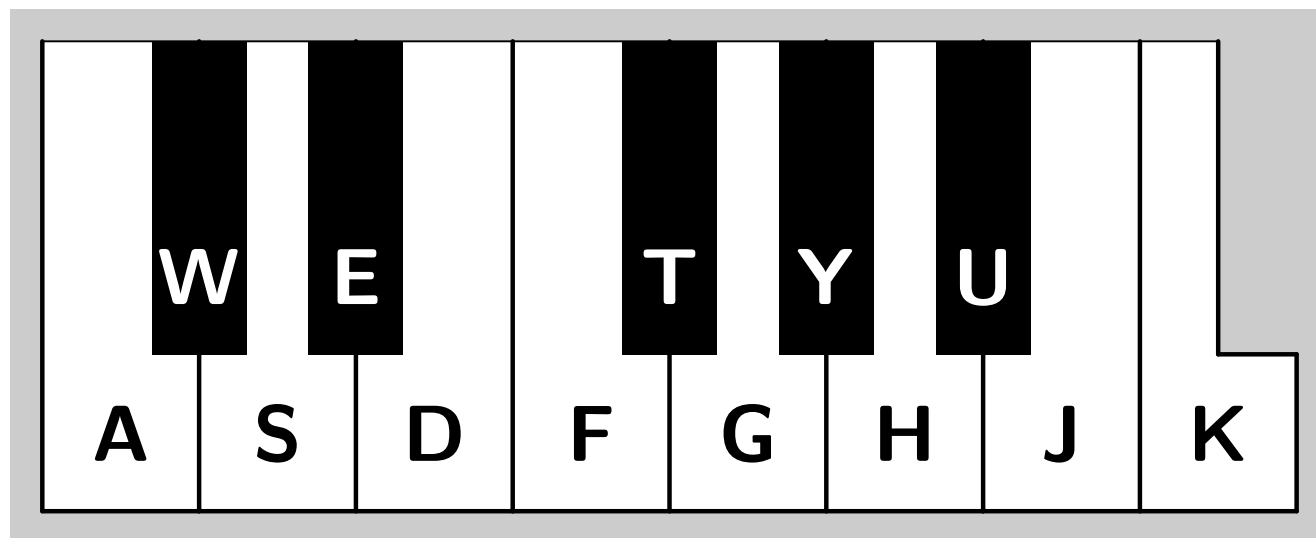
Na organach



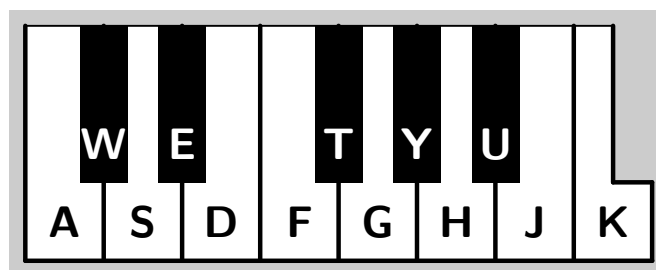
Na organach



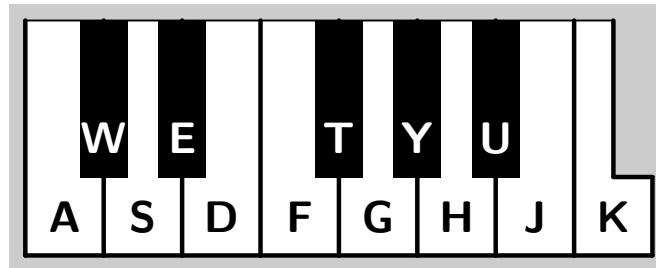
Na organach



Na organach



Na organach

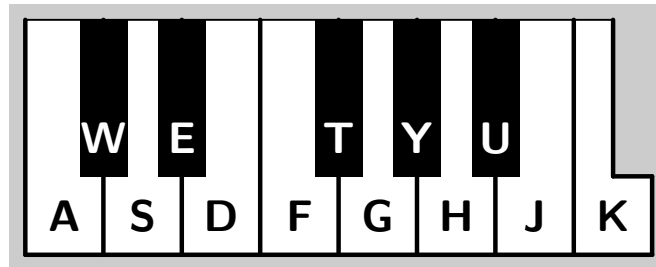


```
class Klawisze extends KeyAdapter {

    public void keyPressed(KeyEvent ev) {
        double herce;
        switch (Character.toUpperCase(ev.getKeyChar())) {
            .....
            case 'E' : /* Dis */ herce = 311.2; break;
            .....
        }
        Dzwiek.graj( x -> { return Math.sin(2*Math.PI * herce * x); } );
    }

    public void keyReleased(KeyEvent ev) {
        Dzwiek.nieGraj();
    }
}
```

Na organach



```
class Klawisze extends KeyAdapter {

    public void keyPressed(KeyEvent ev) {
        double herce;
        switch (Character.toUpperCase(ev.getKeyChar())) {
            .....
            case 'E' : /* Dis */ herce = 311.2; break;
            .....
        }
        Dzwiek.graj( x -> { return Math.sin(2*Math.PI * herce * x); } );
    }

    public void keyReleased(KeyEvent ev) {
        Dzwiek.nieGraj();
    }
}
```

Na organach



Dlaczego nie przestaje grać po zwolnieniu?!

Na organach



Dlaczego nie przestaje grać po zwolnieniu?!

Klawisz naciśnięty i niezwolniony działa tak, jakby był wielokrotnie naciskany.

Na organach



Dlaczego nie przestaje grać po zwolnieniu?!

Klawisz naciśnięty i niezwolniony działa tak, jakby był wielokrotnie naciskany.

Eksperyment: w jakimś edytorze nacisnąć literę i trzymać...

Na organach



Dlaczego nie przestaje grać po zwolnieniu?!

Klawisz naciśnięty i niezwolniony działa tak, jakby był wielokrotnie naciskany.

Eksperyment: w jakimś edytorze nacisnąć literę i trzymać...

nam się wydaje, że:	naciskamy, trzymamy i zwalniamy
program uważa, że:	naciskamy wiele razy a tylko raz zwalniamy

Na organach



Dlaczego nie przestaje grać po zwolnieniu?!

Klawisz naciśnięty i niezwolniony działa tak, jakby był wielokrotnie naciskany.

Eksperyment: w jakimś edytorze nacisnąć literę i trzymać...

nam się wydaje, że:	naciskamy, trzymamy i zwalniamy
program uważa, że:	naciskamy wiele razy a tylko raz zwalniamy

Trzeba to poprawić: jeżeli jakiś klawisz jest wciśnięty, to ignorować następne naciśnięcia.

Na organach



```
class Klawisze extends KeyAdapter {
    private boolean wcisniety = false;

    public void keyPressed(KeyEvent ev) {
        if (!wcisniety) {
            double herce; wcisniety = true;
            switch (Character.toUpperCase(ev.getKeyChar())) {
                .....
                case 'E' : /* Dis */ herce = 311.2; break;
                .....
            }
            Dzwiek.graj( x -> { return Math.sin(2*Math.PI * herce * x); } );
        }
    }

    public void keyReleased(KeyEvent ev) {
        wcisniety = false;
        Dzwiek.nieGraj();
    }
}
```

Na organach



```
class Klawisze extends KeyAdapter {
    private boolean wcisniety = false;

    public void keyPressed(KeyEvent ev) {
        if (!wcisniety) {
            double herce; wcisniety = true;
            switch (Character.toUpperCase(ev.getKeyChar())) {
                .....
                case 'E' : /* Dis */ herce = 311.2; break;
                .....
            }
            Dzwiek.graj( x -> { return Math.sin(2*Math.PI * herce * x); } );
        }
    }

    public void keyReleased(KeyEvent ev) {
        wcisniety = false;
        Dzwiek.nieGraj();
    }
}
```

czy jakiś klawisz
jest wciśnięty

Na organach



```
class Klawisze extends KeyAdapter {
    private boolean wcisniety = false;

    public void keyPressed(KeyEvent ev) {
        if (!wcisniety) {
            double herce; wcisniety = true;
            switch (Character.toUpperCase(ev.getKeyChar())) {
                .....
                case 'E' : /* Dis */ herce = 311.2; break;
                .....
            }
            Dzwiek.graj( x -> { return Math.sin(2*Math.PI * herce * x); } );
        }
    }
}
```

czy jakiś klawisz
jest wciśnięty

```
public void keyReleased(KeyEvent ev) {
    wcisniety = false;
    Dzwiek.nieGraj();
}
}
```

wciśnięcie klawisza, kiedy już jakiś jest wciśnięty,
jest **ignorowane**

Na organach



iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Organy.java

Na organach



iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Organy.java

Problemy:

- gra tylko jednogłosowa,

Na organach



iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Organy.java

Problemy:

- gra tylko jednogłosowa,
- przerwy w grze — po zwolnieniu klawisza przed naciśnięciem następnego system musi „dojść od siebie”.

Efekty

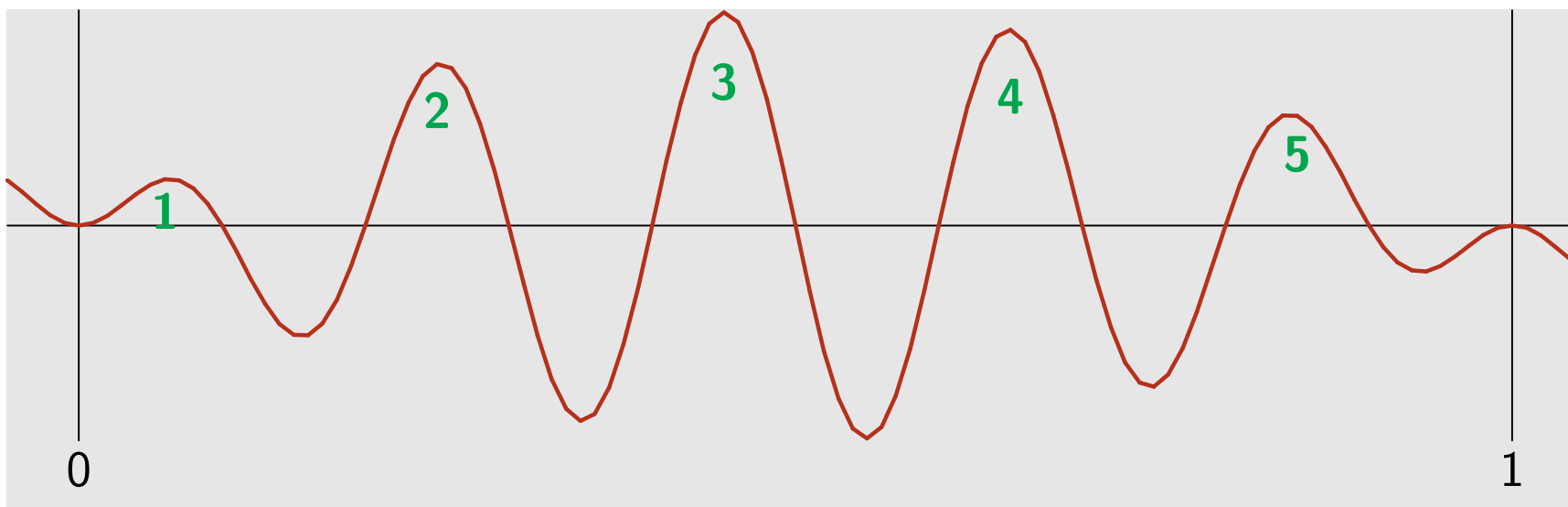
Zmiana głośności



Efekty



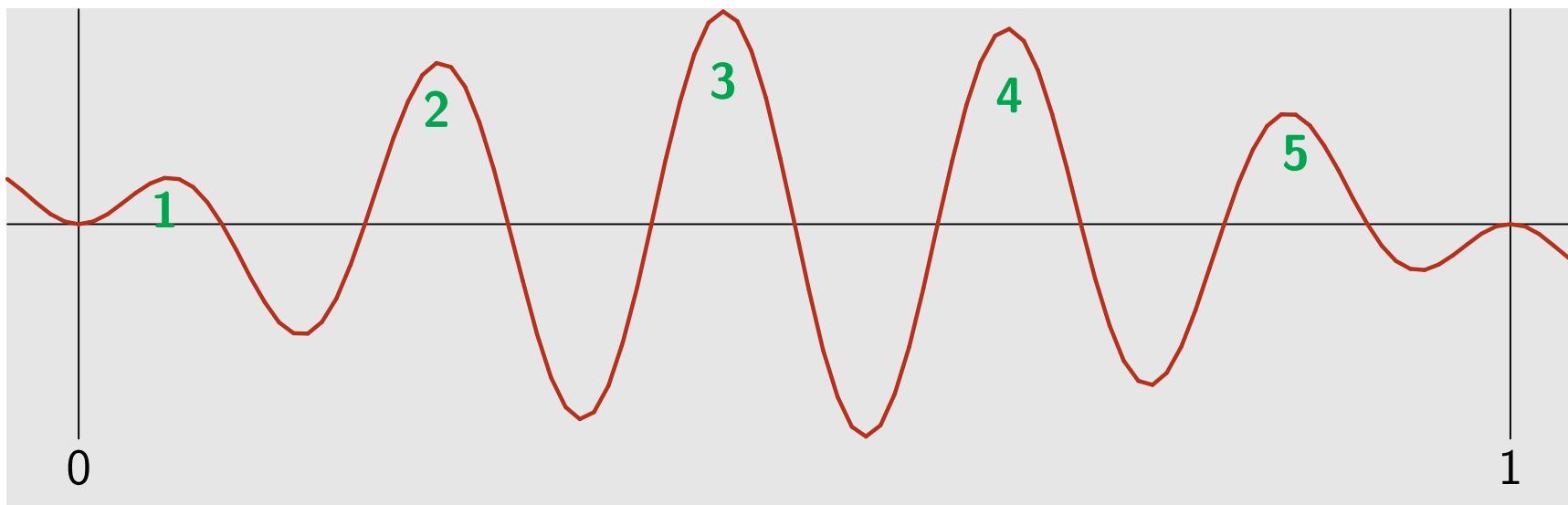
Zmiana głośności



Efekty



Zmiana głośności:

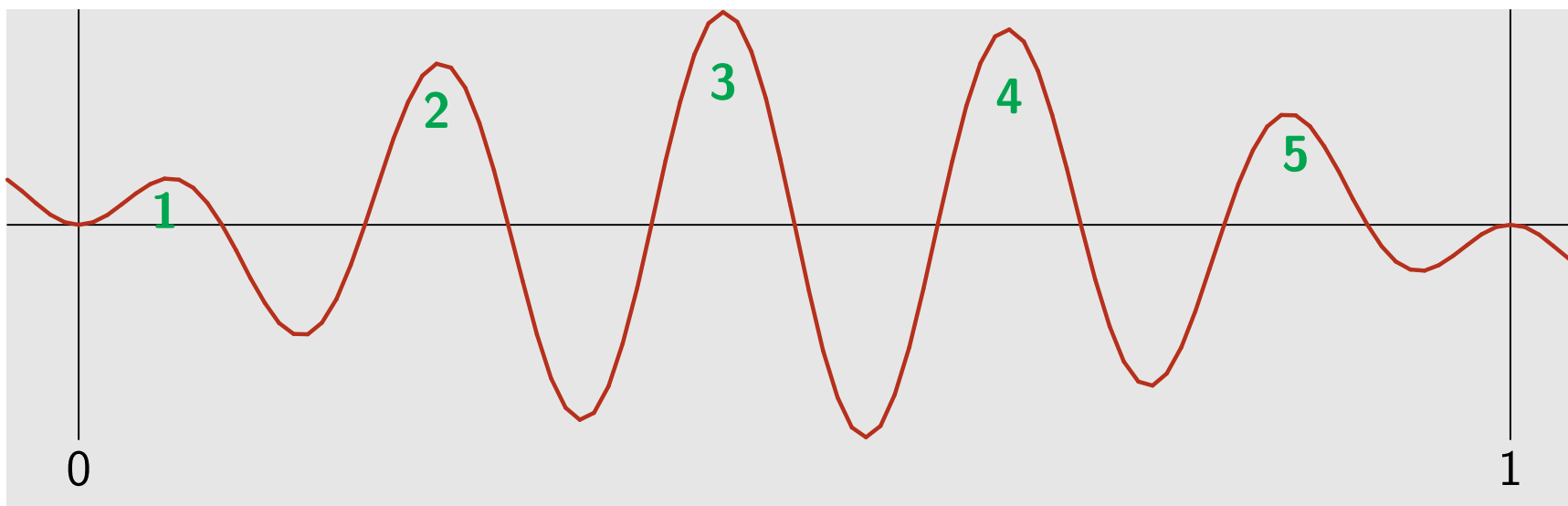


$$f(x) = 4 \cdot x \cdot (1 - x) \cdot \sin(2\pi x \cdot \nu)$$

Efekty



Zmiana głośności:

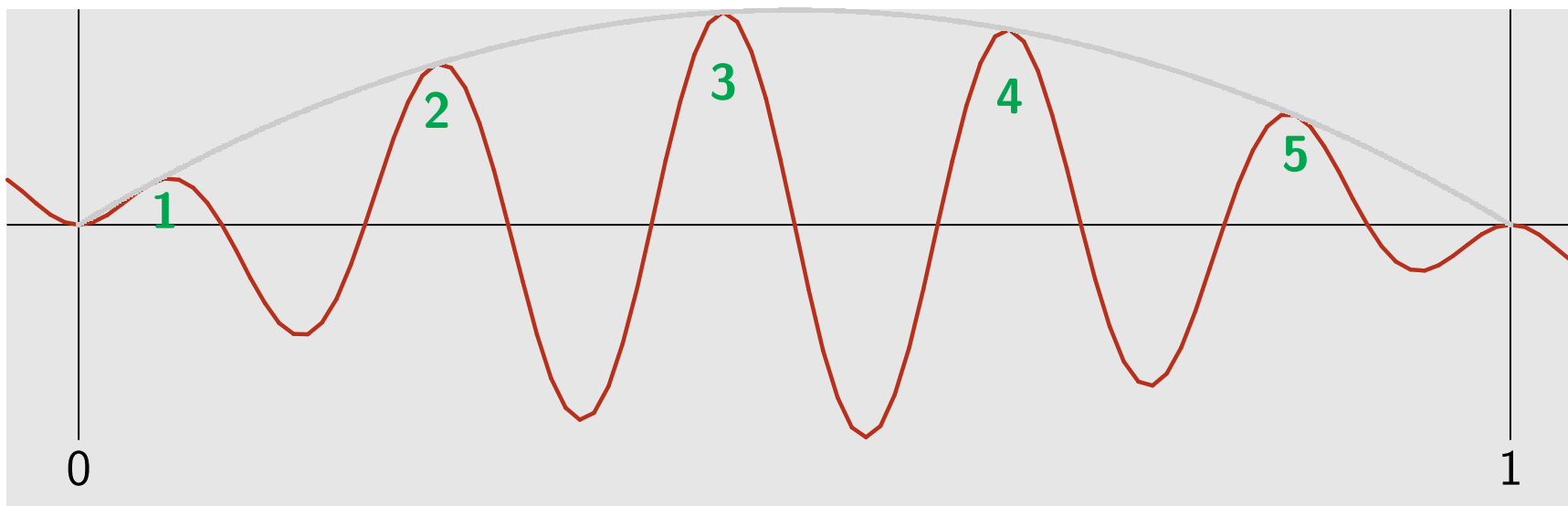


$$f(x) = \underbrace{4 \cdot x \cdot (1 - x)}_{\text{amplituda}} \cdot \sin(2\pi x \cdot \underbrace{\nu}_{\text{częstotliwość}})$$

Efekty



Zmiana głośności:



$$f(x) = \underbrace{4 \cdot x \cdot (1 - x)}_{\text{amplituda}} \cdot \sin(2\pi x \cdot \nu)$$

↑
częstotliwość

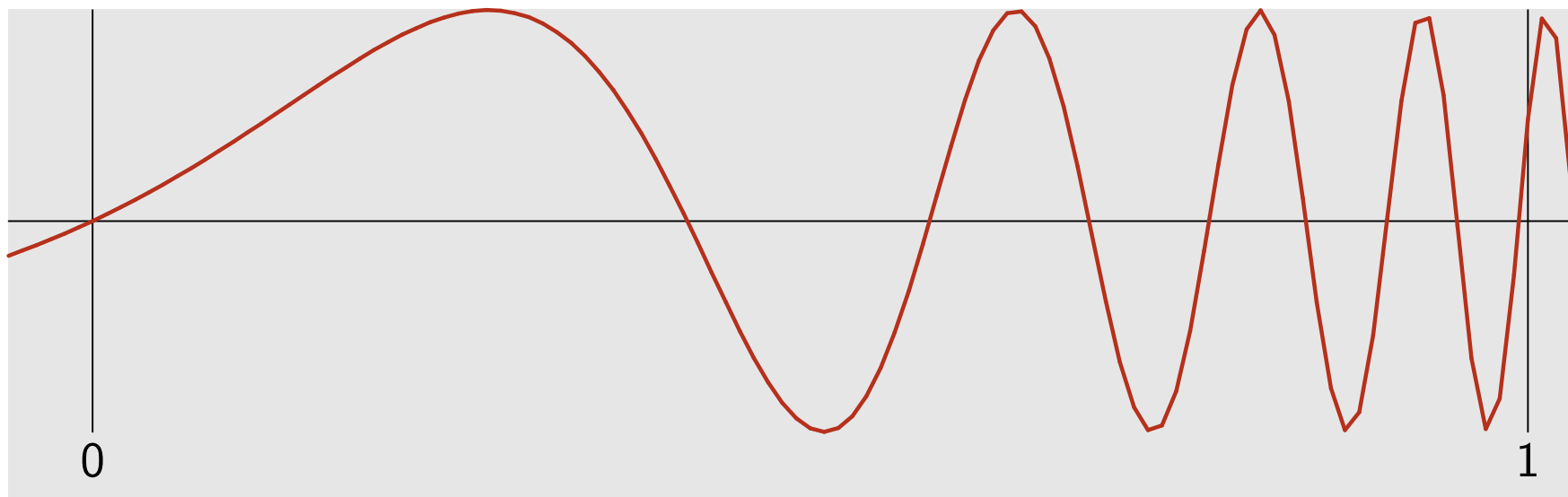
Efekty

Zmiana wysokości



Efekty

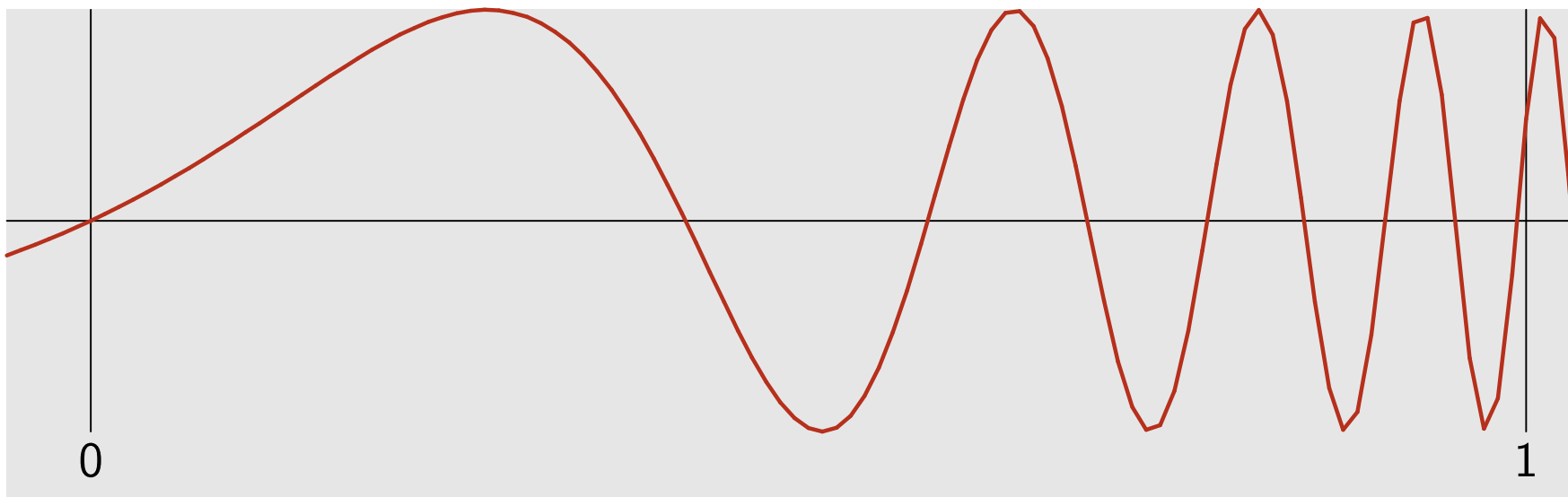
Zmiana wysokości:



Efekty



Zmiana wysokości:

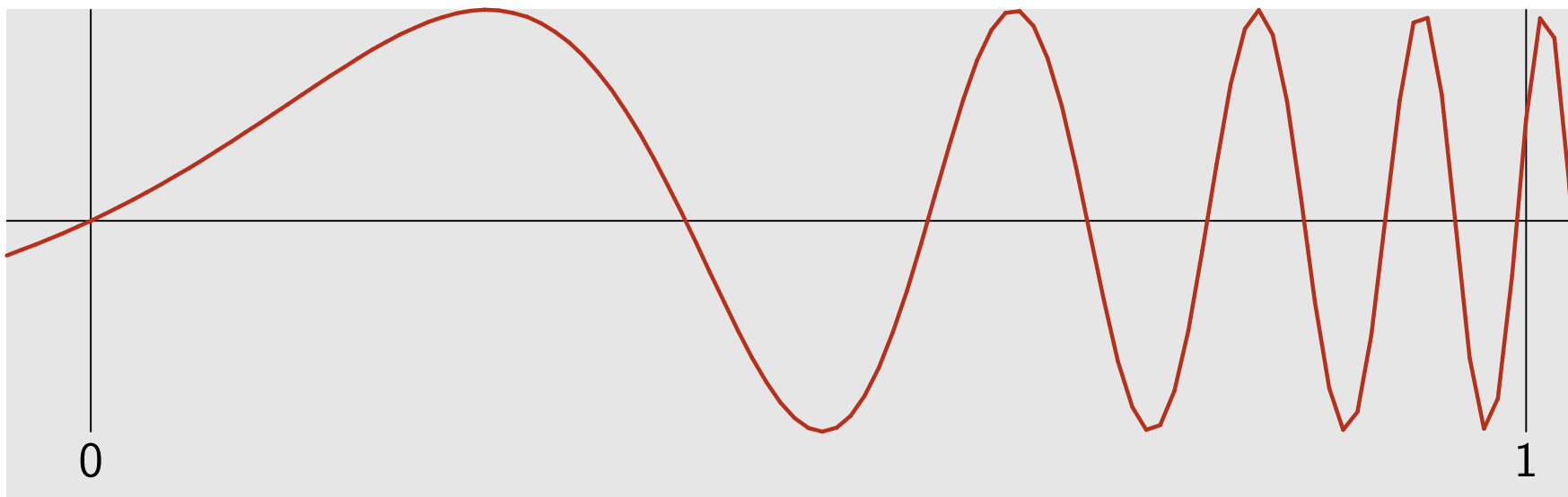


$$f(x) = \sin(2\pi x \cdot \underbrace{2^x})$$

Efekty



Zmiana wysokości:

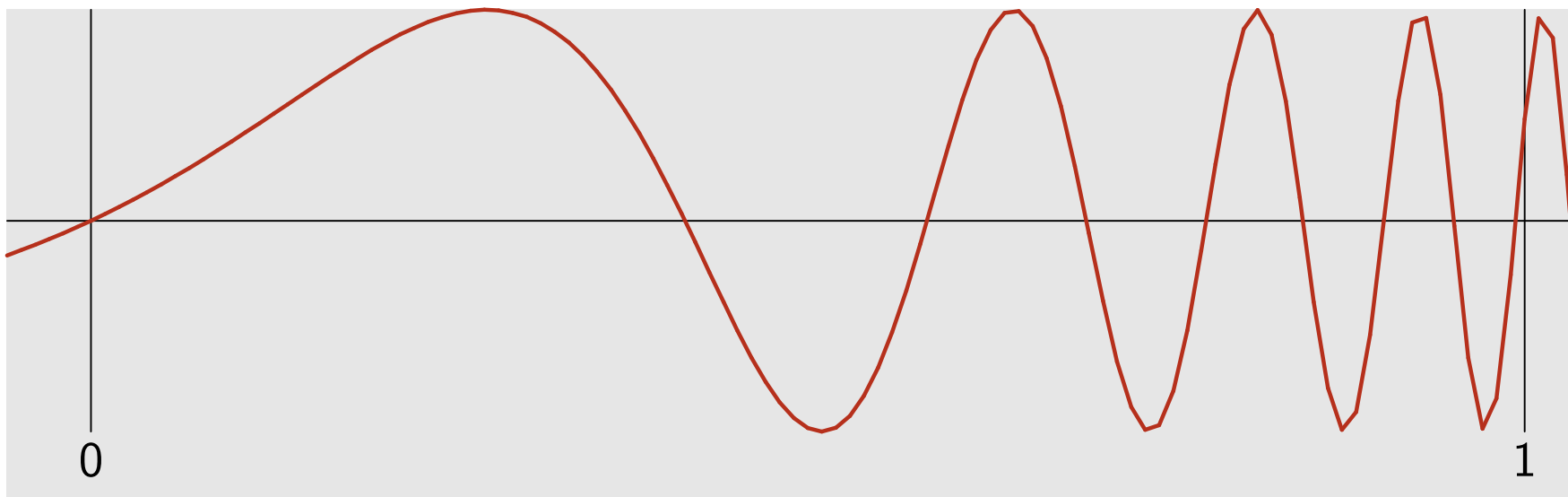


$$f(x) = \sin(2\pi x \cdot \underbrace{2^x}_{\text{częstotliwość}})$$

Efekty



Zmiana wysokości:



$$f(x) = \sin(2\pi x \cdot \underbrace{2^x}_{\text{częstotliwość}})$$

Efekty

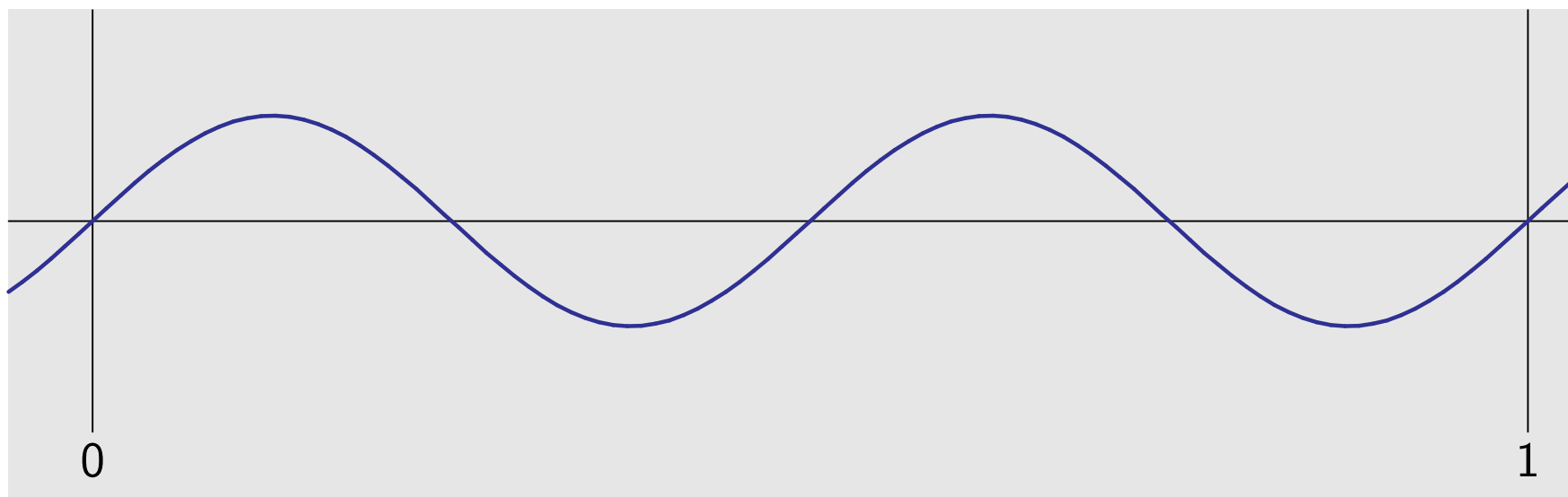
Suma dźwięków



Efekty



Suma dźwięków:

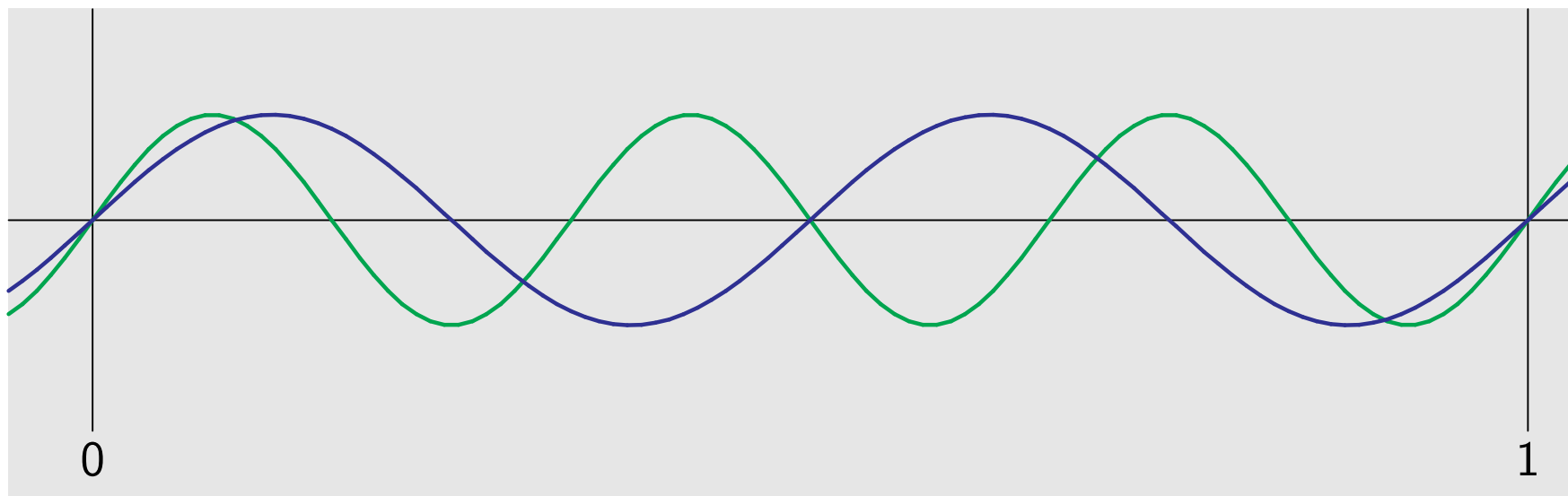


$$\sin(2\pi x \cdot 2)/2$$

Efekty



Suma dźwięków:

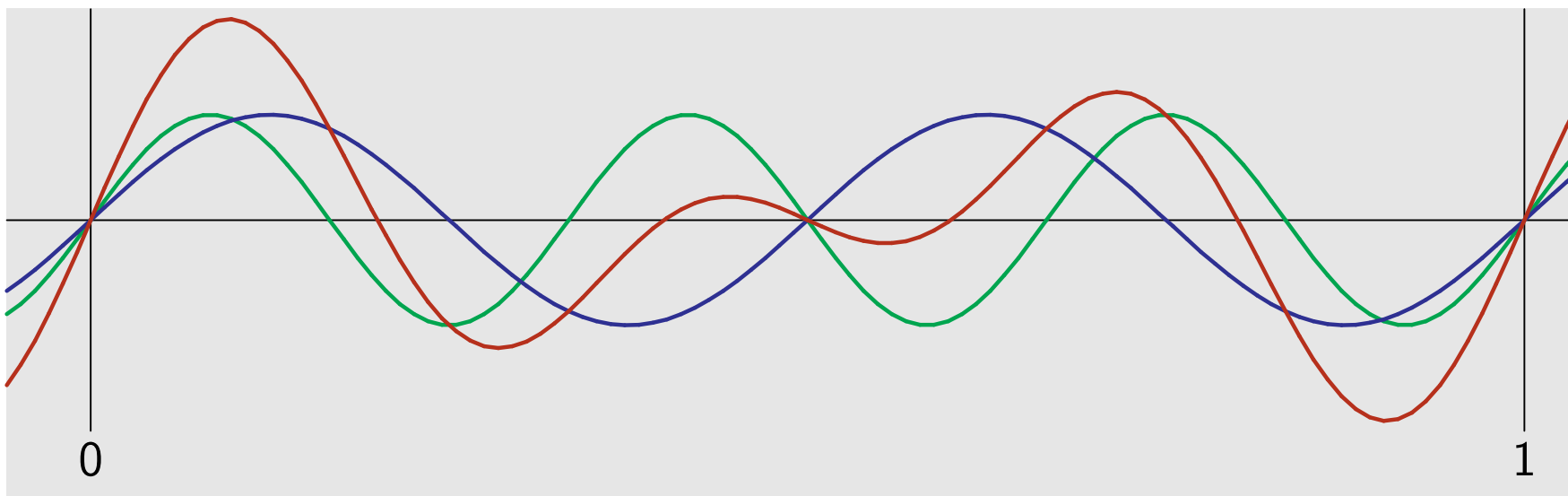


$$\sin(2\pi x \cdot 2)/2 \quad \sin(2\pi x \cdot 3)/2$$

Efekty



Suma dźwięków:



$$f(x) = (\sin(2\pi x \cdot 2) + \sin(2\pi x \cdot 3))/2$$

Efekty

Kiedy dwa dźwięki brzmią jednocześnie



Efekty

Kiedy dwa dźwięki brzmią jednocześnie, wtedy

- wykres drgań jest *sumą* wykresów drgań obu dźwięków



Efekty



Kiedy dwa dźwięki brzmią jednocześnie, wtedy

- wykres drgań jest *sumą* wykresów drgań obu dźwięków;
- jeśli stosunek ich częstotliwości ma postać $\frac{p}{q}$ z małymi całkowitymi p i q , to **konsonans**

Efekty



Kiedy dwa dźwięki brzmią jednocześnie, wtedy

- wykres drgań jest *sumą* wykresów drgań obu dźwięków;
- jeśli stosunek ich częstotliwości ma postać $\frac{p}{q}$ z małymi całkowitymi p i q , to **konsonans**; np.
 - *kwinta* — 7 półtonów

Efekty



Kiedy dwa dźwięki brzmią jednocześnie, wtedy

- wykres drgań jest *sumą* wykresów drgań obu dźwięków;
- jeśli stosunek ich częstotliwości ma postać $\frac{p}{q}$ z małymi całkowitymi p i q , to **konsonans**; np.

– *kwinta* — 7 półtonów = $\sqrt[12]{2^7}$

Efekty



Kiedy dwa dźwięki brzmią jednocześnie, wtedy

- wykres drgań jest *sumą* wykresów drgań obu dźwięków;
- jeśli stosunek ich częstotliwości ma postać $\frac{p}{q}$ z małymi całkowitymi p i q , to **konsonans**; np.

$$\text{— kwinta — 7 półtonów} = \sqrt[12]{2^7} \approx \frac{3}{2}$$

Efekty



Kiedy dwa dźwięki brzmią jednocześnie, wtedy

- wykres drgań jest *sumą* wykresów drgań obu dźwięków;
- jeśli stosunek ich częstotliwości ma postać $\frac{p}{q}$ z małymi całkowitymi p i q , to **konsonans**; np.
 - *kwinta* — 7 półtonów $= \sqrt[12]{2^7} \approx \frac{3}{2}$,
 - *mała kwarta* — 5 półtonów

Efekty



Kiedy dwa dźwięki brzmią jednocześnie, wtedy

- wykres drgań jest *sumą* wykresów drgań obu dźwięków;
- jeśli stosunek ich częstotliwości ma postać $\frac{p}{q}$ z małymi całkowitymi p i q , to **konsonans**; np.
 - *kwinta* — 7 półtonów $= \sqrt[12]{2^7} \approx \frac{3}{2}$,
 - *mała kwarta* — 5 półtonów $= \sqrt[12]{2^5}$

Efekty



Kiedy dwa dźwięki brzmią jednocześnie, wtedy

- wykres drgań jest *sumą* wykresów drgań obu dźwięków;
- jeśli stosunek ich częstotliwości ma postać $\frac{p}{q}$ z małymi całkowitymi p i q , to **konsonans**; np.
 - *kwinta* — 7 półtonów $= \sqrt[12]{2^7} \approx \frac{3}{2}$,
 - *mała kwarta* — 5 półtonów $= \sqrt[12]{2^5} \approx \frac{4}{3}$

Efekty



Kiedy dwa dźwięki brzmią jednocześnie, wtedy

- wykres drgań jest *sumą* wykresów drgań obu dźwięków;
- jeśli stosunek ich częstotliwości ma postać $\frac{p}{q}$ z małymi całkowitymi p i q , to **konsonans**; np.
 - *kwinta* — 7 półtonów $= \sqrt[12]{2^7} \approx \frac{3}{2}$,
 - *mała kwarta* — 5 półtonów $= \sqrt[12]{2^5} \approx \frac{4}{3}$;

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Efekty/E3_Konsonans.java

Efekty



Kiedy dwa dźwięki brzmią jednocześnie, wtedy

- wykres drgań jest *sumą* wykresów drgań obu dźwięków;
- jeśli stosunek ich częstotliwości ma postać $\frac{p}{q}$ z małymi całkowitymi p i q , to **konsonans**; np.

– *kwinta* — 7 półtonów $= \sqrt[12]{2^7} \approx \frac{3}{2}$,

– *mała kwarta* — 5 półtonów $= \sqrt[12]{2^5} \approx \frac{4}{3}$;

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Efekty/E3_Konsonans.java

- jeśli stosunek ich częstotliwości nie da się tak przedstawić, to **dysonans**

Efekty



Kiedy dwa dźwięki brzmią jednocześnie, wtedy

- wykres drgań jest *sumą* wykresów drgań obu dźwięków;
- jeśli stosunek ich częstotliwości ma postać $\frac{p}{q}$ z małymi całkowitymi p i q , to **konsonans**; np.

– *kwinta* — 7 półtonów $= \sqrt[12]{2^7} \approx \frac{3}{2}$,

– *mała kwarta* — 5 półtonów $= \sqrt[12]{2^5} \approx \frac{4}{3}$;

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Efekty/E3_Konsonans.java

- jeśli stosunek ich częstotliwości nie da się tak przedstawić, to **dysonans**; np.
 - *sekunda* — 2 półtony

Efekty



Kiedy dwa dźwięki brzmią jednocześnie, wtedy

- wykres drgań jest *sumą* wykresów drgań obu dźwięków;
- jeśli stosunek ich częstotliwości ma postać $\frac{p}{q}$ z małymi całkowitymi p i q , to **konsonans**; np.

– *kwinta* — 7 półtonów $= \sqrt[12]{2^7} \approx \frac{3}{2}$,

– *mała kwarta* — 5 półtonów $= \sqrt[12]{2^5} \approx \frac{4}{3}$;

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Efekty/E3_Konsonans.java

- jeśli stosunek ich częstotliwości nie da się tak przedstawić, to **dysonans**; np.

– *sekunda* — 2 półtony $= \sqrt[12]{2^2}$

Efekty



Kiedy dwa dźwięki brzmią jednocześnie, wtedy

- wykres drgań jest *sumą* wykresów drgań obu dźwięków;
- jeśli stosunek ich częstotliwości ma postać $\frac{p}{q}$ z małymi całkowitymi p i q , to **konsonans**; np.

– *kwinta* — 7 półtonów $= \sqrt[12]{2^7} \approx \frac{3}{2}$,

– *mała kwarta* — 5 półtonów $= \sqrt[12]{2^5} \approx \frac{4}{3}$;

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Efekty/E3_Konsonans.java

- jeśli stosunek ich częstotliwości nie da się tak przedstawić, to **dysonans**; np.

– *sekunda* — 2 półtony $= \sqrt[12]{2^2} \approx \frac{9}{8}$.

Efekty



Kiedy dwa dźwięki brzmią jednocześnie, wtedy

- wykres drgań jest *sumą* wykresów drgań obu dźwięków;
- jeśli stosunek ich częstotliwości ma postać $\frac{p}{q}$ z małymi całkowitymi p i q , to **konsonans**; np.

– *kwinta* — 7 półtonów $= \sqrt[12]{2^7} \approx \frac{3}{2}$,

– *mała kwarta* — 5 półtonów $= \sqrt[12]{2^5} \approx \frac{4}{3}$;

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Efekty/E3_Konsonans.java

- jeśli stosunek ich częstotliwości nie da się tak przedstawić, to **dysonans**; np.

– *sekunda* — 2 półtony $= \sqrt[12]{2^2} \approx \frac{9}{8}$.

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Efekty/E4_Dysonans.java

Efekty



Kiedy dwa dźwięki brzmią jednocześnie, wtedy

- wykres drgań jest *sumą* wykresów drgań obu dźwięków;
- jeśli stosunek ich częstotliwości ma postać $\frac{p}{q}$ z małymi całkowitymi p i q , to **konsonans**; np.

– *kwinta* — 7 półtonów $= \sqrt[12]{2^7} \approx \frac{3}{2}$,

– *mała kwarta* — 5 półtonów $= \sqrt[12]{2^5} \approx \frac{4}{3}$;

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Efekty/E3_Konsonans.java

- jeśli stosunek ich częstotliwości nie da się tak przedstawić, to **dysonans**; np.

– *sekunda* — 2 półtony $= \sqrt[12]{2^2} \approx \frac{9}{8}$.

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Efekty/E4_Dysonans.java

iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog15/Efekty/E5_Akord.java