CSE 8A Lecture 14

- Reading for next class: None (INTERM EXAM #3)
- Today's topics:
  - More Sounds!
- PSA 6 interviews due Friday 2/22 7pm (deCAF 9:30-2pm)
  - Best to complete interview by Thu 2/21 evening due to tutors attending deCAF
- PSA 7 (sounds) due next Monday (2/25)

Exam 3 practice problem

- Write a method in the Picture class that takes 3 parameters (a target Picture object reference, starting x coordinate, starting y coordinate), and copies the calling object's picture onto the target picture.
- Can you handle these cases:
  - where the target is smaller than the calling object?
  - where only the top half is copied onto the target ?

Exam 3 practice problem

- Write a nested for loop that displays a right triangle with side length of 10 (see diagram) using "System.out.print()" and System.out.println()

```
x
xx
xxx
xxxx
xxxxx
xxxxxx
xxxxxxx
xxxxxxxx
xxxxxxxxx
xxxxxxxxxx
```

Exam 3 practice problem

- Given the code segment below, write a while loop with logical operators that repeats prompting for input (see **OUTPUT**), until the character value is a ‘n’ or ‘N’.

```
char answer = 'y';

_____________
{
    // while loop repeats until answer is 'n' or 'N'
  System.out.print("Want to repeat program (y/n)? ");
    // Code reads user character input into answer
}
```

OUTPUT:

```
Want to repeat program (y/n)?  y
Want to repeat program (y/n)?  X
Want to repeat program (y/n)?  N
```

The Sample Rate that the Sound class ASSUMES is 22 KHz: How long is a SoundSample[] in a Sound object of 1 second?

A. 22 elements

B. 11,000 elements

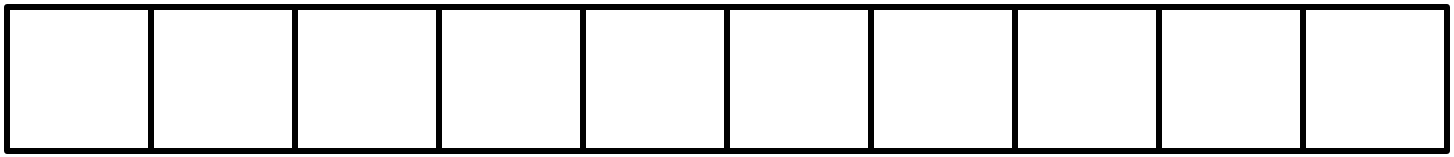
C. 22,000 elements

D. 44,000 elements

E. We can't tell from the data provided

1) Solo: (60 sec)
2) Discuss/Group: (2 min)

What's printed by this code? (assume calling object as shown)



```
public void guess()
{
  SoundSample[] noiseArray = this.getSamples();
  int a = 0, b = 0;
  for (int i=0;i<noiseArray.length;i++)
  {
    SoundSample sample = noiseArray[i];
    int foo = sample.getValue();
    if (foo > a)
    {
      a = foo;
      b = i;
    }
  }
  System.out.println(a + "," + b);
}
```

A. 0,9

B. 60,0

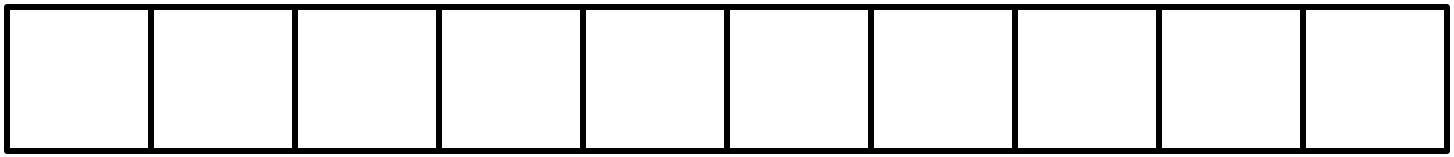
C. 90,5

D. 100,4

E. None of the above

1) 2 min group

What does this code do?



```
…
int a=0, b=0;
for (int i=0; i<noiseArray.length; i++)
  {
    SoundSample sample = noiseArray[i];
    int foo = sample.getValue();
    if (foo > a)
    {
      a = foo;
      b = i;
    }
  }
```

Why would we do that?: Normalizing…

```
public void normalize()
{
  SoundSample[] noiseArray = this.getSamples();
  int maxVal, maxIndex = 0;
  for (int i=0; i<noiseArray.length; i++)
  {
    SoundSample sample = noiseArray[i];
    int val = sample.getValue();
    if (val > maxVal)
    {
      maxVal = val;
      maxIndex = i;
    }
  }
  double factor = 32767.0 / maxVal;
  for (int i = 0; i < noiseArray.length; i++)
  {
    SoundSample sample = noiseArray[i];
    sample.setValue((int) (sample.getValue() * factor));
  }
}
```

CS Concept: Refactoring

```
public void normalize()
{
  int    i, maxVal, maxIndex = 0;
  double factor;
  SoundSample[] noiseArray = this.getSamples();

  for( i=0; i < noiseArray.length ; i++ )
  {
    SoundSample sample = noiseArray[i];
    int val = sample.getValue();
    if (val > maxVal)
    {
      maxVal   = val;
      maxIndex = i;
    }
  }

  factor = 32767.0 / maxVal;
  for( i = 0; i < noiseArray.length ; i++ )
  {
    SoundSample sample = noiseArray[i];
    sample.setValue((int) (sample.getValue() * factor));
  }
}
```

Find the maximum value

normalize

CS Concept: Refactoring

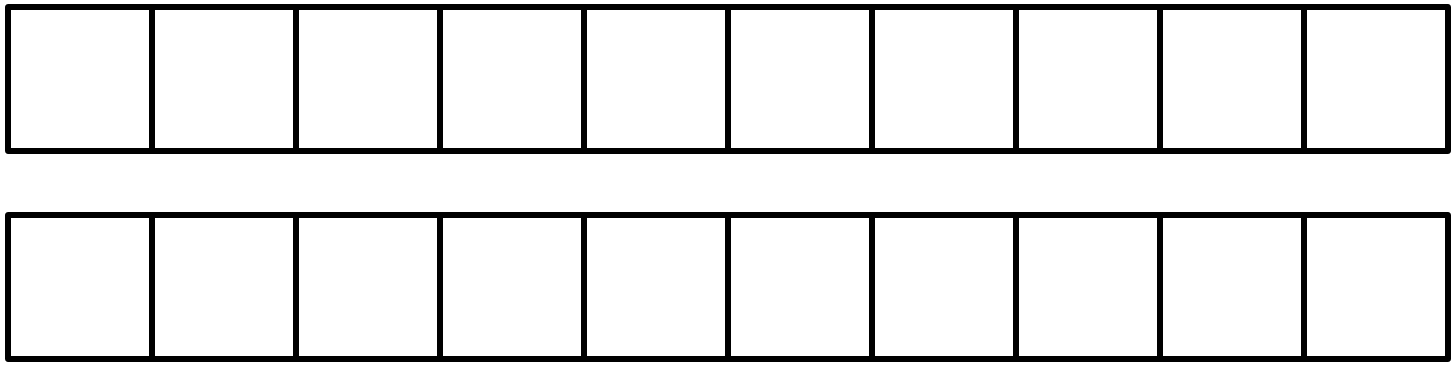
```
public int findMax( SoundSample[] noiseArray )
{
  int  i, val, maxVal;
  for( i=0; i < noiseArray.length ; i++ )
  {
    SoundSample sample = noiseArray[i];
    val = sample.getValue();
    if (val > maxVal)
      maxVal = val;
  }
  return maxVal;
}

public void normalize()
{
  SoundSample[] noiseArray = this.getSamples();
  int    i, maxVal = findMax( noiseArray );
  double factor = 32767.0 / maxVal;
  for( i = 0; i < noiseArray.length ; i++ )
  {
    SoundSample sample = noiseArray[i];
    sample.setValue((int) (sample.getValue() * factor));
  }
}
```

Changing Pitch of a Sound

- Play a recording of someone reading a sentence
- Now play it so that it sounds “high pitched”
  - How long does it take to play the sound high pitched, compared to how long the original takes?
- Now play it so that it sounds “low pitched”
  - How long does it take to play the sound low pitched, compared to how long the original takes?

Raise the pitch of a Sound

- Take only every n^{th} sample from the original sound
- The length of the sound sample is 1/n of the original, and all frequencies in the sound have been increased by a factor of n
- Example, with n == 2:



Options to raisePitch

- Create new Sound
  - V1) Of exact length needed for higher pitched sound
  - V2) Of same length as original with “silence” at end

1) Solo: (60 sec)
2) Discuss/Group: (2 min)

V1: Raise pitch by 2 and return a new sound half as long

Write a method as part of the Sound class that returns a new Sound object whose pitch is double the calling object and whose length is half as long. Create the new sound by taking every other sample from the calling object.

What is the method header for this method?

```
A.public void raisePitch( Sound s )
B.public void raisePitch()
C.public Sound raisePitch()
D.public Sound raisePitch( Sound s )
```

1) Solo: (60 sec)
2) Discuss/Group: (2 min)

A start on raisePitch

```
public Sound raisePitch()
{
    int newPlace = 0;
    SoundSample[] original = this.getSamples();

    Sound highP = new Sound( original.length / 2 );
    SoundSample[] higher = highP.getSamples();

    for (int origI = 0; origI < original.length; origI+=2)
    {
```

What object reference will this method return?

A. this
B. highP
C. original
D. higher
E. void

1) Solo: (60 sec)
2) Discuss/Group: (2 min)

A start on raisePitch

```
public Sound raisePitch()
{
  int origI, newPlace = 0;

  SoundSample[] original = this.getSamples();

  Sound  highP = new Sound( original.length / 2 );

  SoundSample[] higher = highP.getSamples();

  for( origI = 0; origI < original.length; origI+=2 )
  {
```

What do you think **newPlace** should be used for ?

A. As an index into the original sample array
B. As an index into the higher sample array
C. To store the value to be copied from original
D. To store the length of the higher sample array

1) Solo: (60 sec)
2) Discuss/Group: (2 min)

Complete the raisePitch method

```
public Sound raisePitch()
{
  int origI, newPlace = 0;

  SoundSample[] original = this.getSamples();

  Sound highP = new Sound( original.length / 2 );

  SoundSample[] higher = highP.getSamples();

  for( origI = 0; origI < original.length ; origI+=2 )
  {
```

Complete V2: Create new sound of same length with 0 at end

```
public Sound raiseP()
{
  int   origI, newPlace = 0;
  Sound highP = new Sound(this);

  SoundSample[] original = this.getSamples();
  SoundSample[] higher   = highP.getSamples();

  for( origI = 0; origI < original.length; origI+=2 )
  {
```

Concept Summary

- When you want to create a “new” object…
  - Call a “constructor” with new.
  - Look in the file of that class to find out what constructors are available
    - What parameters you can send
- Don’t forget to return the object you created with a return statement!
- When working with 2 (multiple) arrays
  - Sometimes you will want 2 index variables (to index into them) moving independently
  - If you are indexing “in synchrony” then use one index variable– it’s easier to understand!

TODO

- Reading for next class: None
- Study for exam 3
- Start on PSA7 (Bring headphones to lab!)