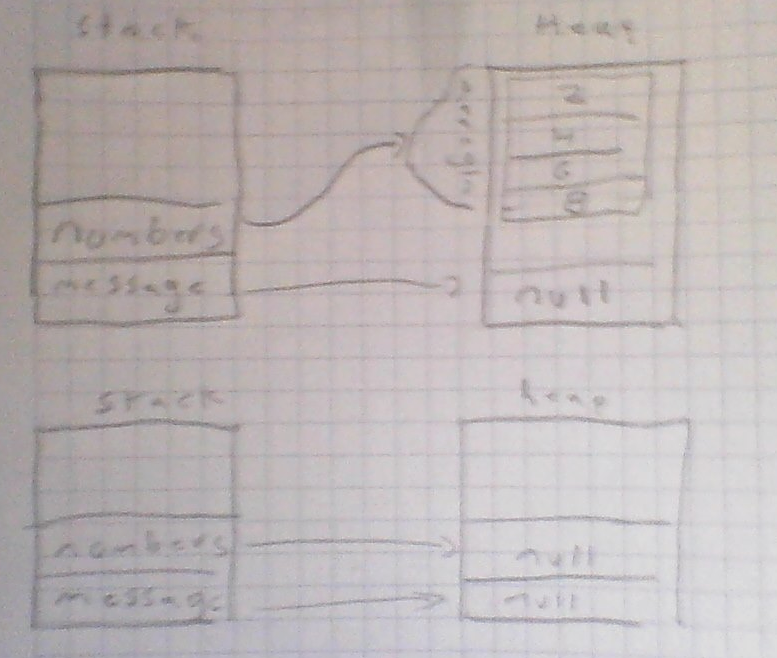
Programación, semana 2

1. Dentro de una solución pueden haber varios proyectos, dentro de cada proyecto hay un ensamble
2. Para indicarle al compilador las bibliotecas que va a usar
3. Int, byte, Double, Bool, Char, Float, short, son estructuras
4. signed puede llevar números negativos y los Unsigned solo positivos
5. No funciona ya que el rango se sale para un sbyte
6. El tamaño, el rango se utiliza el short para optimizar los programas
7. Si se puede, entra en el rango y es positivo
8. La f es float, L es long, UL es ulong y m es decimal
9. Porque esta esperando una respuesta
10. Cambiar tipos de datos
11. Console.WriteLine("C:\\Users\\juanfh\\Desktop\\MiArchivo.txt");

C2 float result = (float)((float)(a) / (float)(b)) + (c / d);

1. La suma desborda la variable y por eso no da el resultado correcto
2. línea 1: int[] scores en esta línea se crea un vector  
   línea 2: int[] scores = new int[10]; en esta línea se crea un vector y se le asigna un tamaño
3. int[] scores = new int[10] { 100, 95, 92, 87, 55, 50, 48, 40, 35, 10 }; crea un vector, le asigna un tamaño y le ingresa valores a cada casilla del vector
4. En la 12 lee las instrucciones que se le dan salta de línea para ejecutar la función, pero antes de saltar deja una nota o instrucción que le hace volver a la línea 12 una vez ejecuta la función
5. En el stack
6. El garbash collector borra la información que no se usa
7. La variable message queda gaurdada en el stack y la cadena “hello!” en el heap
8. Message es almacenada en el stack y el resto en el heap
9. Numbers queda en el stack y los 2,3,4 en el heap
10. 
11. a=3 y b=4
12. imprime 17 ya que al ejecutar b[0]=17 se cambia el arreglo en el heap
13. returnvalue=3 y returnvalue2 = 4 ya que se modifica el .MyValue en el heap y cambia el 3 por el 4
14. es publico para acceder a las variables o atributos que están en privado y no es necesario puede tener cualquier nombre
15. el código no funciona porque el X=-5; esta fuera de contexto, ya que la X no esta definida fuera del for
16. para referirse a una variable title fuera del metodo
17. sirve para separar separar las palabras en espacios (o con el carácter que esta dentro de los paréntesis)
18. “Al metodo WriteLine() le pasamos como parámetro una string. Además, cuando le pasamos una string a un método, la escribimos entre comillas” http://retoscript.com/hola-mundo-en-c-sharp/ Lo que yo entiendo es que a ese método todo lo que uno escribe entre comillas es guardado como una string y este al ejecutarse lo imprime tal cual como esta escrito
19. (gracias por la ayuda) La primera imagen es una estructura entonces esta hecho con en tipo valor y la segunda a ser una clase está hecha con tipo estructura
20. ¿Qué implica la diferencia de tipo? explique con un ejemplo

-Tipo valor: el tipo valor guarda la información en el stack y seria:

int X= 4;

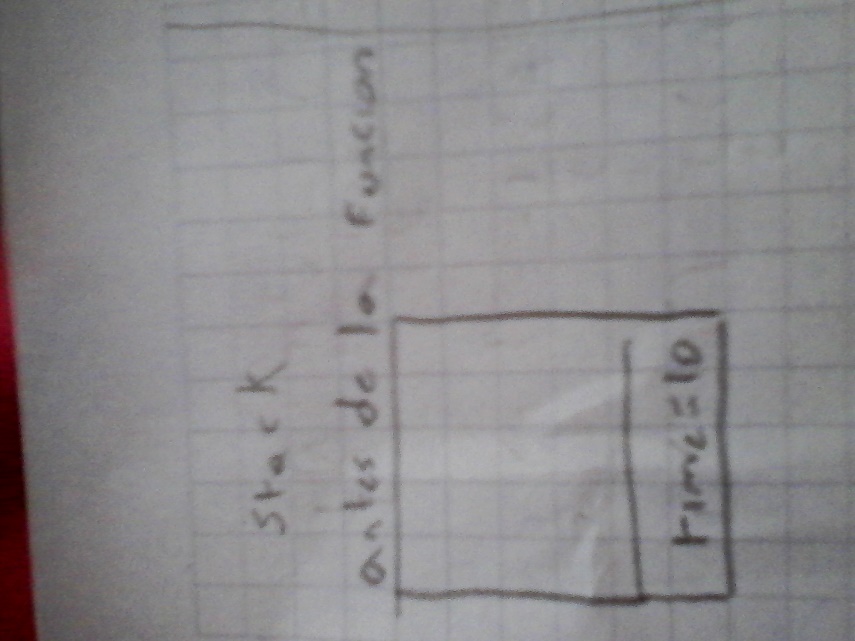
-Tipo referencia: los tipo referencia se asignan en el Heap y seria:

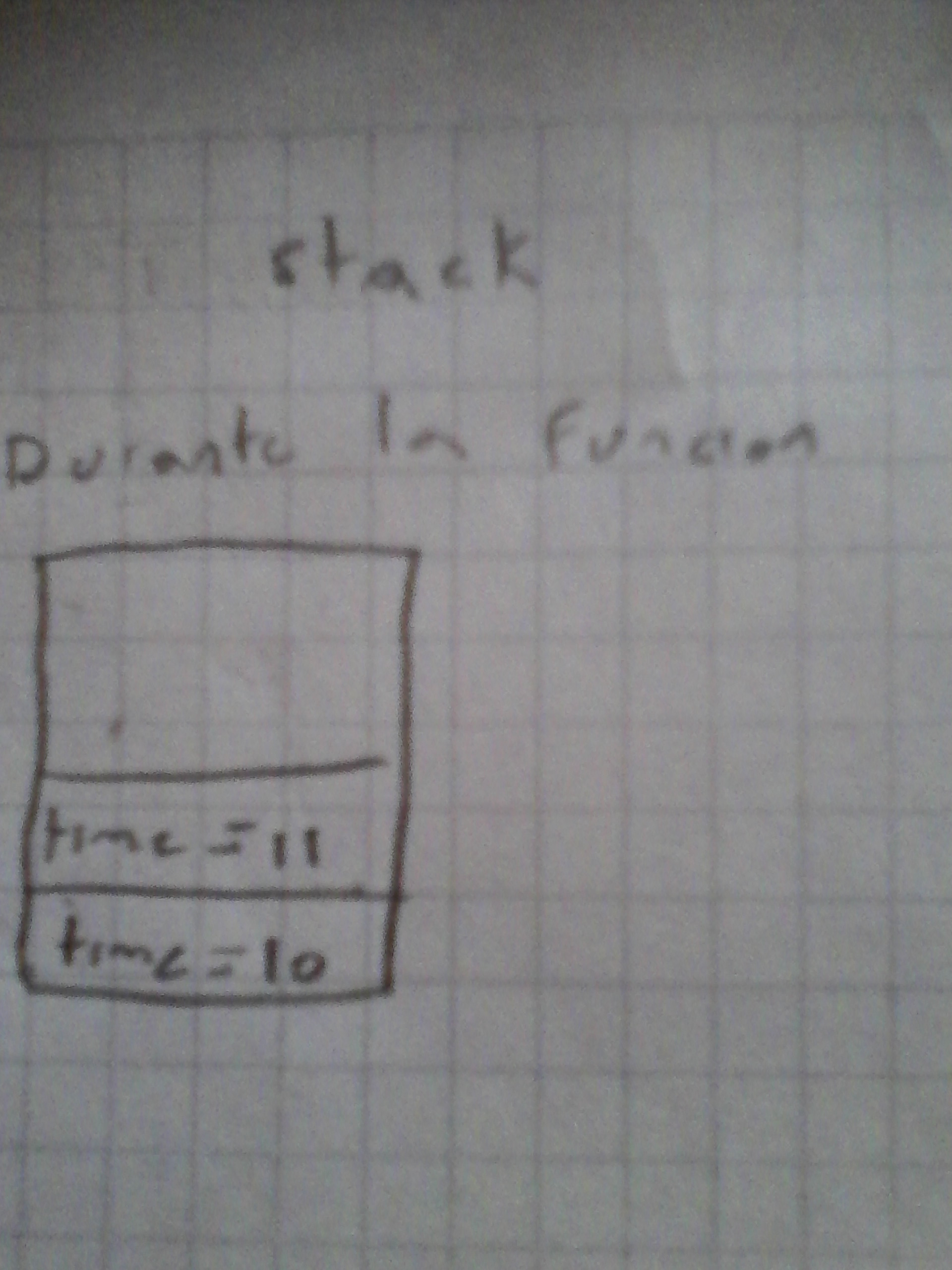
Int[] x =new int[10];

Que tendría un vector de tamaño 10

C5:- Se espera que aparezca un 11 pero sale un 10

-Porque es un tipo estructura entonces en el stack cuando se llama la función crea una copia de la variable pero al no ser un tipo referencia al terminar la función esta se borra y queda el mismo 10





C6: -se espera que aparezca 200

-Guarda en el stack la dirección del arreglo que se encuentra en el heap y al ejecutarse la función se copia la variable y por ende copia la dirección del arreglo que esta en el heap, y este cambia lo que esta en la posición [1]

