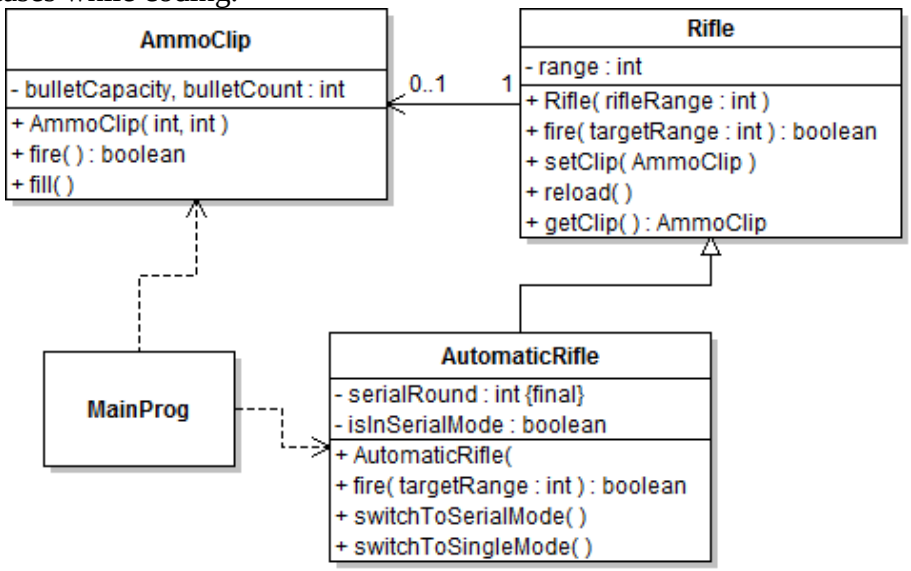
OBJECT ORIENTED CONCEPTS (BLM2911), Gr.1&Gr.2, 2nd MIDTERM EXAM (13/12/2017)

Student Number:				Name, Surname:						Signature:
Scoring:	1	2	3	4	5	6	7	Group	Duration:	
	20	25	25	15	15				90mins.	

QUESTIONS

Answer the questions according to the incomplete UML class diagram below. Keep in mind that the diagram can be incomplete. Therefore you may need to add necessary members to classes. Take necessary precautions to prevent illogical cases while coding.



Question 1: Write the source code of the class AmmoClip. Details of its members are as follows:

a) Two member fields for storing the bullet capacity of the ammo clip and the current number of bullets in the ammo clip. The clip cannot be filled over its capacity.

b) A constructor for initializing those members.

c) A method for firing one bullet from the ammo clip. Bullet count cannot be negative at any times.

d) A method for filling the ammo clip.

e) Additional necessary method(s)

Question 2: Write the source code of the class Rifle. Each rifle has a range; a rifle cannot shoot beyond its range. A typical range for a rifle is several hundred meters. Additionally, the Rifle class must have the following methods:

a) A constructor for initializing the range.

b) A method for firing a bullet. This method must take the range of the target as its only parameter.

c) A method for inserting an ammo clip into the rifle.

d) A method for reloading the rifle.

Question 3: Write the source code of the class AutomaticRifle. Such rifles shoot multiple bullets after a trigger pull, determined by the serialRound member, if it is currently in serial mode. Otherwise, they shoot only 1 bullet.

Question 4: Please write the source code of a class with a main method where an automatic rifle is created and fired. All information must be entered by the user at runtime.

Question 5: Draw the UML sequence diagram of the main method you have coded.

Question 1: Design a class named AmmoClip.

```
public class AmmoClip {
    private int bulletCapacity, bulletCount;

    public AmmoClip(int bulletCapacity, int bulletCount) {
        this.bulletCapacity = bulletCapacity;
        if( bulletCapacity < 0 )
            this.bulletCapacity = 6; //At least a 6-shooter, could also be 1
        this.bulletCount = bulletCount;
        if( bulletCount > bulletCapacity )
            this.bulletCount = bulletCapacity;
    }
    public boolean fire( ) {
        if( bulletCount > 0 ) {
            bulletCount--; return true;
        }
        return false;
    }
    public void fill( ) {
        bulletCount = bulletCapacity;
    }
    public int getBulletCapacity() { return bulletCapacity; }
    public int getBulletCount() { return bulletCount; }
}
```

```
/* DO NOT ADD
setBulletCount and
setBulletCapacity;
that will make all
capacity checking code
meaningless! */
```

Question 2: Design a class named Rifle which has a member of type AmmoClip.

```
public class Rifle {
    private int range;
    private AmmoClip clip;

    public Rifle( int range ) {
        this.range = range;
    }
    public boolean fire( int targetRange ) {
        if( clip == null )
            return false;
        if( targetRange <= range )
            return clip.fire();
        else return false;
    }
    public void reload( ) {
        if( clip != null )
            clip.fill( );
    }
    public AmmoClip getClip() {  return clip; }
    public void setClip(AmmoClip clip) { this.clip = clip; }
}
```

Görüldüğü üzere bu sorunun cevabı clip üyesine dayanıyor. Bu üyeyi eklemeyenler bu sorudan çok puan kaybettiler.

Question 3: Design a class named AutomaticRifle.

```
public class AutomaticRifle extends Rifle{
    private final int serialRound;
    private boolean isInSerialMode;

    public AutomaticRifle(int range, int serialRound) {
        super( range );
        if( serialRound < 1 ) this.serialRound = 3;
        else                  this.serialRound = serialRound;
    }
    public boolean fire( int targetRange ) {
        boolean result = super.fire(targetRange);
        if( isInSerialMode )
            for( int i = 0; i < serialRound-1; i++ )
                result = result || super.fire(targetRange) ;
        return result;
    }
    public void switchToSerialMode( ) { isInSerialMode = true; }
    public void switchToSingleMode( ) { isInSerialMode = false; }
}
```

Question 4: Please write the source code of a class with a main method.

```
import java.util.*;
public class MainProg {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter bullet capacity for ammo clip: ");
        int capacity = in.nextInt();
        AmmoClip clip = new AmmoClip(capacity, capacity);
        System.out.println("Enter range of automatic rifle: ");
        int range = in.nextInt();
        System.out.println("Enter bullet count to shoot at one trigger pull: ");
        int serial = in.nextInt();
        AutomaticRifle ak47 = new AutomaticRifle(range, serial);
        ak47.setClip(clip);
        System.out.println("Enter target range: ");
        range = in.nextInt();
        if( ak47.fire(range) )
            System.out.println("Target is hit.");
        else
            System.out.println("I have missed!");
        in.close();
    }
}
```

Question 5: Draw the UML sequence diagram of the main method you have coded.