

BLM 2012 OBJECT ORIENTED PROGRAMMING Sec.1 (2018/2) D012				30	25	25	20	
Sıra	No	Adı	Soyadı	Q1	Q2	Q3	Q4	2. Ara Sınav
1	08011057	FATİH	OKYAY	6	0	19	8	33
2	10011057	ALI KEMAL CEM	ÖZDEMİR	4	4	8	1	17
3	10011094	YİĞİT	YANDAS	18	6	20	0	44
4	11011017	HAKAN CAN	İPEK	23	21	25	0	69
5	11011084	ZÜLFÜ	KORKMAZ	G	G	G	G	Girmedi
6	12011015	FATİH	ÇOMAK	21	0	14	0	35
7	12011048	SALİH MERT	YİĞİT	G	G	G	G	Girmedi
8	12011065	NAZİM BURAK	GENÇ	21	11	18	11	61
9	12011079	ÖĞÜZHAN	TIRYAKI	13	7	24	0	44
10	13011047	ENES	ÖZKAYA	15	6	15	2	38
11	13011089	FATMA BETÜL	KARAPINAR	4	0	10	0	14
12	14011021	BETÜL	AKTAŞ	13	6	20	0	39
13	14011052	RAHMİ CEMRE	ÜNAL	27	16	15	0	58
14	14011063	EMİNE BETÜL	SINAR	13	7	17	5	42
15	14011064	YAĞIZ	AKYÜZ	22	25	18	5	70
16	14011704	HATİCE	TEKİN	4	0	10	0	14
17	15011003	YUSUF EMIN	KINATAŞ	G	G	G	G	Girmedi
18	15011012	UFUK CAN	AKSOY	24	18	24	11	77
19	15011024	MUHAMMET CELAL	USTAÖMER	22	8	19	17	66
20	15011030	KÜBRA	KARAKUYU	26	11	21	7	65
21	15011032	ABDULLAH	GENÇ	11	11	21	0	43
22	15011040	MUHAMMED KAĞAN	ÜLKER	13	2	20	0	35
23	15011044	DILARA	SUVEREN	14	0	5	2	21
24	15011050	SERRA	SEMİZ	11	5	19	0	35
25	15011051	AŞIR FURKAN	KAYACIK	9	7	15	5	36
26	15011053	ORÇUN	ÇELİK	22	17	19	10	68
27	15011057	BAHADIR	GÜLTEKİN	12	7	21	8	48
28	15011058	HALİT	GÖRMEZ	21	8	21	0	50
29	15011063	METEHAH	ÇAYLAK	13	9	19	0	41
30	15011064	ALTUĞ NUMAN	YILDIZ	19	9	15	5	48
31	15011066	ALPEREN ÖNDER	ÖZKAN	6	2	15	0	23
32	15011070	FURKAN	ZEREY	24	13	17	0	54
33	15011081	ALPER	SELMAN	22	21	17	2	62
34	15011702	ÖZGÜR	KAN	17	8	24	2	51
35	15011907	HUSEYN	VALİYEV	15	3	13	1	32
36	16011002	AYKUT	AKDENİZ	23	7	22	14	66
37	16011003	AHMET	AYDIN	18	9	21	10	58
38	16011004	SERDAR	YILDIZ	26	17	24	15	82
39	16011005	MEHMET SERDAR	ORMANCI	2	0	3	0	5
40	16011006	MUSTAFA	ÖZTÜRK	G	G	G	G	Girmedi
41	16011008	ÇİHAN	KAYA	23	17	24	5	69
42	16011010	FURKAN NECATİ	ÜRKMEZ	G	G	G	G	Girmedi
43	16011016	AHMET	ELGÜN	20	7	17	0	44
44	16011017	MERT	ÖZ	23	14	12	0	49
45	16011023	MEHMET HAYRİ	ÇAKIR	18	16	25	15	74
46	16011030	İLAYDA	ŞAHİN	12	2	15	10	39
47	16011032	FARUK	ARSLAN	22	13	20	20	75
48	16011033	YUSUF	ANI	24	18	22	15	79
49	16011039	ONUR	AYDEMİR	19	21	23	20	83
50	16011043	RECEP FURKAN	KOÇYİĞİT	23	15	25	20	83
51	16011048	AHMET	BAGCI	16	8	18	8	50
52	16011057	BUŞRA	KÜDEN	16	4	22	20	62
53	16011061	BERKAY	HAMARAT	18	6	18	2	44
54	16011062	EMİN TEYHAN	USLU	22	0	18	3	43
55	16011063	BURAK	DURSUNLAR	24	19	25	11	79
56	16011066	MEHMET FURKAN	ŞAHİN	22	8	25	7	62
57	16011067	KEREM	YOLCU	21	18	19	0	58
58	16011068	SELİN	GEZER	8	3	18	0	29
59	16011073	İPEK	KOÇ	19	6	16	3	44
60	16011078	EDİZ	AYDOĞAN	17	0	0	0	17
61	16011080	TOYGAR	KAYAŞ	27	13	22	18	80
62	16011081	FAZLI	BOZATAY	19	25	22	12	78
63	16011082	CEM ÖZGÜR	DEDE	G	G	G	G	Girmedi
64	16011085	FATMA	TANRIKULU	12	5	17	10	44
65	16011087	CANER	KAYA	25	12	25	17	79
66	16011088	İBRAHİM	TİPİRDİK	25	16	26	15	82
67	16011090	AHMET	KAYGISIZ	16	14	2	14	46
68	16011093	HALİL İBRAHİM	ULUOĞLU	17	12	22	20	71
69	16011112	ENES	GÜNDÜZ	1	1	5	2	9
70	16011114	SEDAT	AKGÜL	9	8	6	3	26
71	16011116	ÖMÜR	ÇİÇEN	G	G	G	G	Girmedi
72	16011120	HÜSEYİN MURAT	GEZER	4	0	19	11	34
73	16011133	ÜMİT RASİM	TURHAN	17	10	16	10	53
74	16011135	UGURCAN	YILMAZ	G	G	G	G	Girmedi
75	16011609	SEYYİD İBRAHİM	GÜLEÇ	18	10	22	6	56
76	16011701	ATAKAN	TEKOĞLU	19	0	16	0	35
77	16011703	KAAN	AKINTÜRK	G	G	G	G	Girmedi
78	16011705	YUNUS EMRE	DEMİR	5	2	21	5	33
79	16011906	KHAGANI	KHANKISHIYEV	12	2	5	2	21
80	17011001	ELİF	ÖZCAN	22	16	20	9	67
81	17011017	MEHMET ALİ	VARGÜN	22	16	16	0	54
82	17011024	ELİF	ÖZKAN	20	16	21	8	65
83	17011029	SEZİN	BİNER	16	13	22	3	54
84	17011045	BEYZA NUR	SEZGİN	23	8	22	15	68
85	17011048	TOLGA	ZİFTÇİ	23	20	25	20	88
86	17011052	HÜSREV	YUMUŞAK	22	14	22	10	68
87	17011053	MAHMUT MURAT	AKTAN	24	22	22	20	88
88	17011060	YUSUF ETKİN	KIZILDAĞ	11	0	20	0	31
89	17011064	ÖĞÜZHAN	SARIALIOĞLU	24	11	25	17	77
90	17011066	ALP BINTUĞ	UZUN	21	16	19	0	56
91	17011068	FATMA ZEHRA	ÇETİN	19	3	19	3	44
92	17011069	ALPEREN	ÖZER	22	14	15	0	51
93	17011072	BERKE KAAH	AÇIKGÖZ	19	2	19	7	47
94	17011091	EREN	TERZİOĞLU	18	12	21	20	71
95	17011602	SENA	OCAKÇI	13	0	17	0	30
96	17011609	İSMAIL	BAYRAM	24	6	23	8	61
97	17011611	BETÜL	ÖN	16	13	22	4	55
98	17011902	NİNİ	KVATCHANTIRADZE	19	14	18	16	67
99	17011903	NIHAD	GULUZADE	15	8	16	12	51
100	17011904	KAMRAN	BALAYEV	9	6	15	2	32
101	17011908	MECDEDDİN	HARRAD	23	19	22	7	71
102	18011602	CANBERK	GÜLLÜOĞLU	22	16	21	10	69
103	18011613	MEHMET EMRE	GÜL	16	15	21	15	67
104	E18011005	ABDUFATTOKH	ABDUFATTOKHOV	G	G	G	G	Girmedi
105	E18011006	DASTAN	YULDASHEV	G	G	G	G	Girmedi
106	E18012004	EMİNA	FAJIC	5	0	5	0	10
107	E18012005	MERJEMA	MEHMEDOVIC	10	6	5	0	21
108	F18011003	HÜSEYİN BERK	BUGUR	19	10	13	20	62
109	13059023	SABİHA YAĞMUR	BAYRAM	8	0	18	10	36
110	13067701	TURAN	ÖZEV	G	G	G	G	Girmedi
			<b>Ort.</b>	17.06	9.41	17.96	6.95	51.38
			<b>Başarı</b>	57%	38%	72%	35%	51%

**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ / BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

Öğrencinin Adı Soyadı:	Öğrenci No:	İmza:			
Dersin Adı: BLM2012 Nesneye Yönelik Progr.	Tarih/Saat: 02/05/2019 09:00	Sınav süresi: 90dk			
Sınav Türü:	Vize1	Vize2 +	Mazeret	Final	Bütünleme
Unvan Ad Soyad (Ders Yürütücüsü):					

**QUESTIONS**

**Attention:** Make necessary checks when coding methods. For example, you ask for a student to write down something but s/he cannot do it without pen and paper.

**Question 1 (30 points):** Write the source code of classes named Target, Robot and Weapon.

A target has an ID, a range and a health value as integer primitives. Additionally, a target has a textual description. The health of a target will decrease when it is hit.

A weapon has a maximum range, a damage value that it can deal, maximum ammunition capacity and current ammunition count. First three of those are fixed values and all are integers. A weapon also has a textual description, too. A weapon always fires one round with its namesake method and if the weapon is not empty, that method returns true. It can be reloaded, too.

A robot has a fixed serial number and description, both as textual information. A robot can carry a weapon and if so, it can shoot at a target with its namesake method. That method checks whether it can hit the target with the robot's weapon. If everything is OK, it fires the robot's weapon, hits the target and returns true. Keep in mind that a robot cannot move, therefore the target must be within the range of the robot's weapon. A robot can also be queried for how many rounds it needs to shoot in order to destroy a specific target. If it is not able to successfully hit that target, this method returns zero.

**Question 2 (25 points):** Write the source code of a class named MovingRobot, a specialized version of a robot that tries its best to move closer to its target if the target is out of its weapon's range while shooting at a target. It has a fixed fuel capacity, current fuel level and burns a fixed amount of fuel per one unit of distance. These three values are real numbers. It can be refueled and can go to a specific distance. Assume 1 unit of distance as equal to 1 unit of range.

**Question 3 (25 points):** Draw a detailed UML class diagram that contains everything you have coded so far.

**Question 4 (20 points):** Draw a detailed UML sequence diagram of a main method that creates a moving robot and a target, then executes the underlined method in Question 1. Taking user input is not necessary.

**Question 1 (30 points):** Write the source code of classes named Target, Robot and Weapon.

```
public class Target {
    private final int id;
    private int range, health;
    private String description;

    public Target(int id, int range, int health) {
        this.id = id; this.range = range; this.health = health;
    }
    public int getId() { return id; }
    public int getRange() { return range; }
    public int getHealth() { return health; }
    public String getDescription() { return description; }
    public void setRange(int range) { this.range = range; }
    //public void setHealth(int health) { this.health = health; }
    public void setDescription(String descr) { description = descr; }

    //as we have this method, better not add setHealth method.
    //requirements do not say health cannot be negative.
    public void hit( int damage ) {
        health -= damage;
    }
}

public class Robot {
    private final String serialNr;
    private String description;
    private Weapon weapon;

    public Robot(String serialNr) {
        this.serialNr = serialNr;
    }
    public void setDescription(String descr) { description = descr; }
    public void setWeapon(Weapon weapon) { this.weapon = weapon; }
    public String getSerialNr() { return serialNr; }
    public String getDescription() { return description; }

    /* necessary for MovingRobot to override shootAt.
     * can define Weapon as protected instead. */
    protected Weapon getWeapon( ) { return weapon; }

    public boolean shootAt( Target target ) {
        if( canHit(target) ) {
            if( weapon.fire() ) {
                target.hit(weapon.getDamage());
                return true;
            }
        }
        return false;
    }

    public boolean canHit( Target target ) {
        if( weapon == null || target.getRange() > weapon.getMaxRange() )
            return false;
        if( weapon.getCurrentAmmo() == 0 ) return false;
        return true;
    }

    public int howManyTimesToShoot( Target target ) {
        if( canHit(target) ) {
            return (int)Math.ceil(target.getHealth()/weapon.getDamage());
        }
        return 0;
    }
}
```

```

public class Weapon {
    private final int maxRange, damage, ammoCapacity; //2018-2'de maxRange final değil
    private int currentAmmo;
    private String description;

    public Weapon(int maxRange, int damage, int ammoCapacity) {
        this.maxRange = maxRange; this.damage = damage; this.ammoCapacity = ammoCapacity;
    }
    public int getMaxRange() { return maxRange; }
    public int getDamage() { return damage; }
    public int getAmmoCapacity() { return ammoCapacity; }
    public int getCurrentAmmo() { return currentAmmo; }
    public String getDescription() { return description; }

    public void setDescription(String description) { this.description = description; }

    public boolean fire( ) {
        if( currentAmmo > 0 ) {
            currentAmmo --; return true;
        }
        return false;
    }
    public void reload( ) { currentAmmo = ammoCapacity; }
}

```

**Question 2 (25 points):** Write the source code of a class named MovingRobot

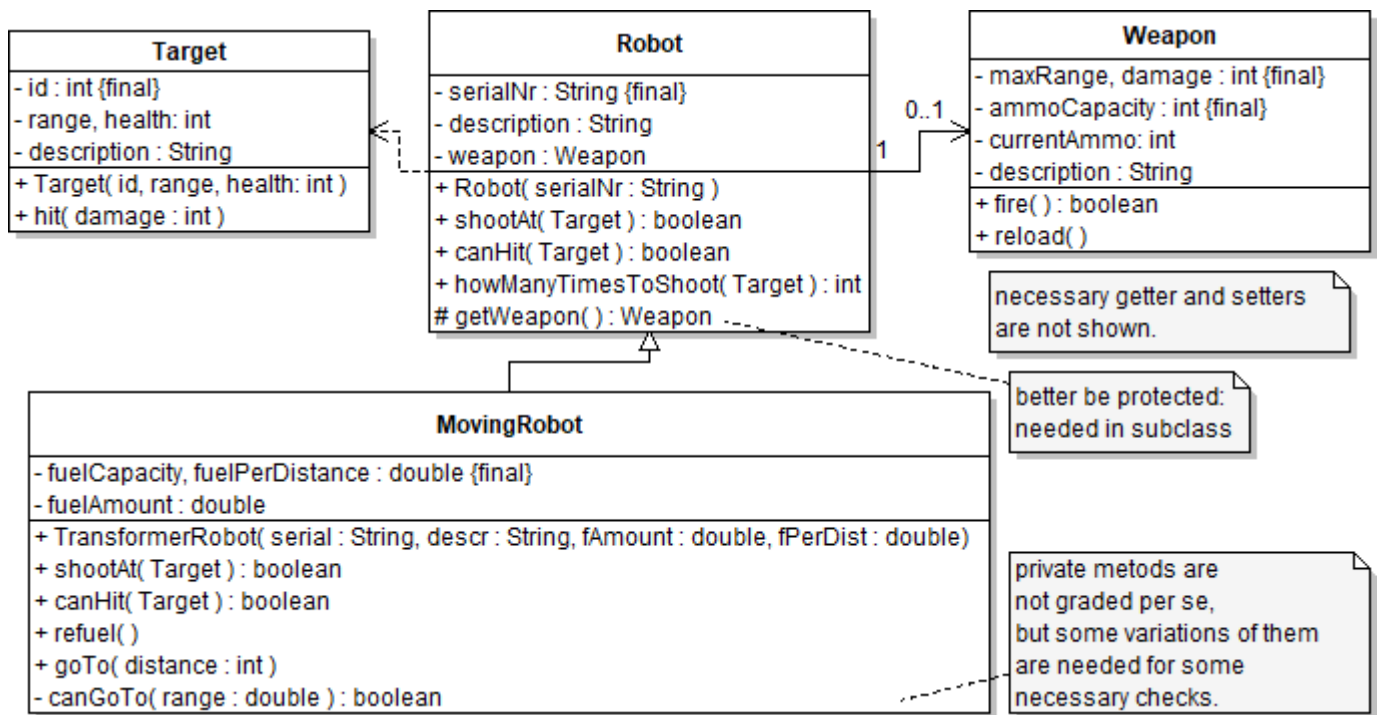
```

public class MovingRobot extends Robot {
    private final double fuelPerDistance, fuelCapacity;
    private double fuelAmount;

    public MovingRobot(String serialNr, double fuelCapacity, double fuelPerDistance) {
        super(serialNr);
        this.fuelCapacity = fuelCapacity; this.fuelPerDistance = fuelPerDistance;
    }
    private boolean canGoTo( double range ) {
        if( fuelAmount / fuelPerDistance >= range ) return true;
        return false;
    }
    public void goTo( double range ) {
        if( canGoTo(range) )
            fuelAmount -= range * fuelPerDistance;
    }
    public void refuel( ) {
        fuelAmount = fuelCapacity;
    }
    public boolean canHit( Target target ) {
        if( super.canHit(target) ) return true;
        else if( canGoTo(target.getRange()-getWeapon().getMaxRange() ) )
            return true;
        return false;
    }
    public boolean shootAt( Target target ) {
        if( canHit(target) ) {
            if( target.getRange() > getWeapon().getMaxRange()
                && canGoTo(target.getRange()-getWeapon().getMaxRange()) ) {
                goTo(target.getRange()-getWeapon().getMaxRange());
                target.setRange(target.getRange()-getWeapon().getMaxRange());
            }
            getWeapon().fire();
            return true;
        }
        return false;
    }
}

```

**Question 3 (25 points):** Draw a detailed UML class diagram



**Question 4 (20 points):** Draw a detailed UML sequence diagram of a main method that creates a target and a regular robot, then executes the underlined method.

