Tek Sınıf Kullanılarak Çözülebilecek Sorular:

Soru 1:

a. Komut satırına * işaretlerinden dikdörtgen çizen bir sınıf tasarlayın. Bu sınıf dikdörtgenin kenarlarını nextInt komutu kullanmadan belirlemeye yarayacak en azından bir metot içermelidir. Tasarımınızı bir UML sınıf şeması ile gösteriniz.
b. Teoride bir dikdörtgen herhangi bir ebatta olabilir. Ancak tipik bir komut satırı en fazla 24 satır ve 80 sütundan ibarettir. Tasarımınızı buna göre güncelleyin ve Java kaynak kodunu yazın. UML sınıf şemasını tekrar çizmek zorunda değilsiniz.
c. Sınıfınıza kullanıcının dikdörtgen çizmesine yarayacak bir main metodu ekleyin. Kenar uzunluklarını kullanıcıdan almak için nextInt komutları kullanınız.
d. Main metodunun sıralama şemasını çiziniz.

Soru 2:

Farklı bilgilerin tutarlılığını denetlemek üzere bir hizmet sınıfı yazmak istiyorsunuz. Sınıfınızı aşağıdaki işlevleri sağlayacak şekilde kodlayınız:

a. Kitaplara verilen ISBN numaralarının son basamağı sınama amaçlı olarak kullanılır. 10 basamaklı bir ISBN numarasının basamakları soldan sağa x_1, x_2, … x_{10} olsun. Sınama basamağı olan x_{10}, aşağıdaki hesaplama ile bulunacak rakamla aynı olmalıdır:
    - $x_{10} = 11 – (10 * x_1 + 9 * x_2 + 8 * x_3 + 7 * x_4 + 6 * x_5 + 5 * x_6 + 4 * x_7 + 3 * x_8 + 2 * x_2)$ mod 11.
    - Eğer son basamak 10 olarak hesaplanmışsa, son basamak olarak X kullanılır.
b. Bir e-posta adresinin geçerli olması için içerisinde bir @ karakteri ve en az bir nokta bulunmalıdır.

İpuçları:

- String sınıfının şu metotlarının javadoc'una bakabilirsiniz:
    i. public char[]toCharArray()
    ii. public static String valueOf(char c)
    iii. public int indexOf(String ch)
    iv. public int indexOf(String ch, int fromIndex)
    v. public int lastIndexOf(String ch)
- Integer sınıfının şu metotlarının javadoc'una bakabilirsiniz
    i. public static Integer valueOf(String s)
- Tüm bu değinilen metotların dışında metotlarla da bu işi yapanlar çıkabilir mi?

Yanıtlar (kısmi, farklı çözümler de bulunabilir):

```
package examples01;
import java.util.Scanner;
public class Rectangle {
	private int sideA, sideB;
	private final static int maxSideA = 24, maxSideB = 80;
	public Rectangle(int sideA, int sideB) {
		if( sideA > maxSideA )
			this.sideA = maxSideA;
		else
			this.sideA = sideA;
		if( sideB > maxSideB )
			this.sideB = maxSideB;
		else
			this.sideB = sideB;
	}
	public void setSideA( int sideA ) {
		if( sideA > maxSideA )
			this.sideA = maxSideA;
		else
			this.sideA = sideA;
	}
	public void setSideB( int sideB ) {
		if( sideB > maxSideB )
			this.sideB = maxSideB;
		else
			this.sideB = sideB;
	}
	public void draw() {
		for( int i=0; i<sideA; i++ ) {
			for( int j=0; j<sideB; j++ ) {
				if( i == 0 || i == sideA-1
					|| j == 0 || j == sideB-1)
					System.out.print("*");
				else
					System.out.print(" ");
			}
			System.out.println();
		}
	}
	public static void main( String[] args ) {
		Scanner in = new Scanner( System.in );
		System.out.print("Enter side A: ");
		int a = in.nextInt();
		System.out.print("Enter side B: ");
		int b = in.nextInt();
		Rectangle r = new Rectangle( a, b );
		r.draw();
		in.close();
	}
}
```

Bire-bir Sahiplik İlişkisi ile İlgili Sorular:

Soru 1: Bir füzenin kilometre cinsinden menzili ve kilogram cinsinden ağırlığı mevcuttur. Bir füze ya kara hedefleri ya da hava hedefler için tasarlanır. Füzelerin bu özellikleri daha sonradan değiştirilemez. Füze sınıfının (Missile) Java kodunu yazınız.

Soru 2: Her uçağın bir boş ağırlığı ve bir maksimum kalkış ağırlığı vardır. Haddinden fazla yüklenen uçak uçamaz. Bir uçak ancak uçarken füze ateşleyebilir. Üzerinde pilot olmayan uçaklara UAV denir. Bazı UAV'ler füze taşıyamaz, taşıyabilenler ise tek türden ve sınırlı sayıda füze taşır. Verilen tüm bu bilgilere göre UAV sınıfının kaynak kodunu yazınız.

Soru 3: Önceki sorularda yazdığınız sınıfları test etmek üzere main metodu içeren bir sınıfın kaynak kodunu yazınız. Main metodunda kullanıcıdan en azından bir tamsayı bilgi girişi almaya çalışınız. Her sınıftan bir nesne oluşturup bir füze ateşlemek yeterlidir.

Soru 4: Şimdiye kadar kodladığınız sınıfları gösteren ayrıntılı bir UML şeması çiziniz.

Soru 5: Füze ateşlemeye yarayan metodun UML sıralama şemasını çiziniz.

Yanıtlar (kısmi, farklı çözümler de bulunabilir):

Question 1:

```
public class Missile {
	private final int range, weight;
	private final boolean forAirTargets;

	public Missile(int range, int weight, boolean forAirTargets) {
		this.range = range;
		this.weight = weight;
		this.forAirTargets = forAirTargets;
	}
	public int getRange() { return range; }
	public int getWeight() { return weight; }
	public boolean isForAirTargets() { return forAirTargets; }
}
```

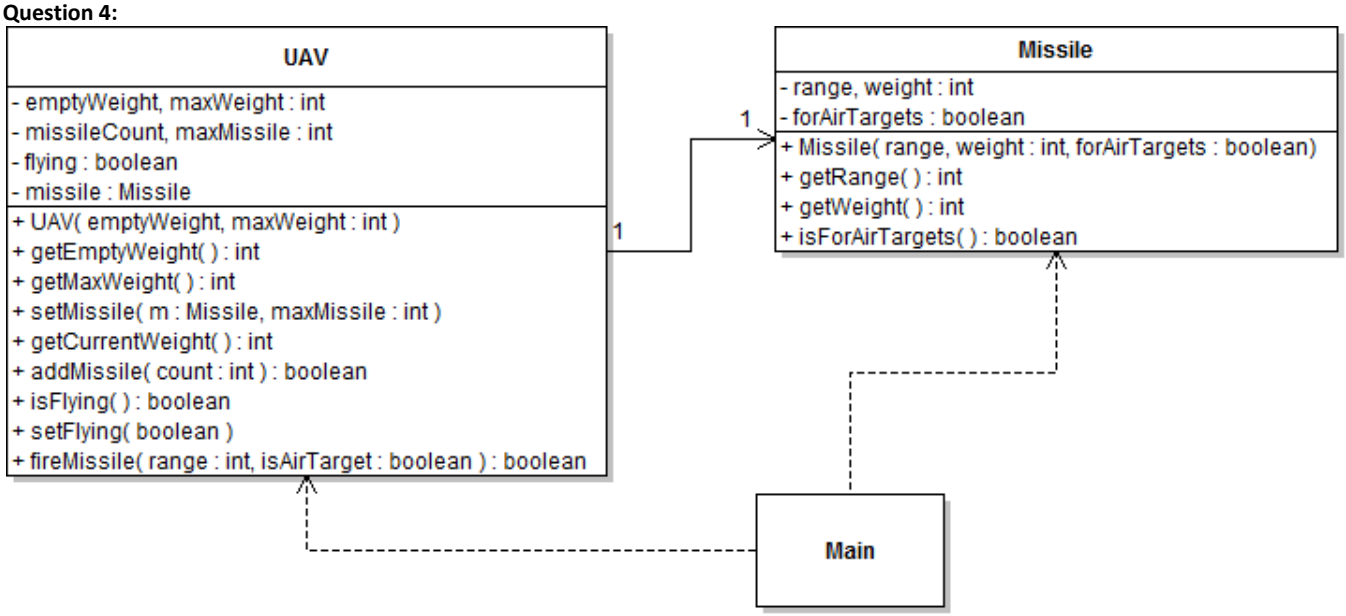
Question 2:

```
public class UAV {
	private int emptyWeight, maxWeight;
	private Missile missile;
	private int missileCount, maxMissile;
	private boolean flying;

	public UAV(int emptyWeight, int maxWeight) {
		this.emptyWeight = emptyWeight;
		this.maxWeight = maxWeight;
	}
	public int getEmptyWeight() { return emptyWeight; }
	public int getMaxWeight() { return maxWeight; }
	public boolean isFlying() { return flying; }
	public void setFlying(boolean flying) { this.flying = flying; }

	public void setMissile(Missile missile, int maxMissile) {
		this.missile = missile;
		this.maxMissile = maxMissile;
		missileCount = 0;
	}
	public int getCurrentWeight( ) {
		int result = emptyWeight;
		if( missile != null )
			result += missileCount * missile.getWeight();
		return result;
	}
	public boolean addMissile( int count ) {
		if( getCurrentWeight() + count * missile.getWeight() < maxWeight
				&& missileCount + count <= maxMissile ) {
			missileCount += count;
			return true;
		}
		return false;
	}
	public boolean fireMissile( int range, boolean isAirTarget ) {
		if( missile != null && missileCount > 0
				&& missile.getRange() > range && isFlying() &&
				missile.isForAirTargets() == isAirTarget )
			return true;
		return false;
	}
}
```

Question 3:

```
import java.util.*;
public class Main {
	public static void main(String[] args) {
		Missile sidewinder = new Missile(1200, 100, true);
		UAV predator = new UAV(5000, 20000);
		Scanner in = new Scanner( System.in );
		System.out.print("How many missiles? ");
		int count = in.nextInt();
		predator.setMissile(sidewinder, count);
		predator.addMissile(count);
		predator.setFlying(true);
		if(predator.fireMissile(100, true))
			System.out.println("Test is successful");
		else
			System.out.println("Test has failed");
		in.close();
	}
}
```

Question 4:



Question 5: