Java OOP
10128

Polymorphism, Abstract

Pini shlomi

Person class

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
public class Person {

  protected String name;
  protected int id;

  public Person( int id, String name) {
    this.name = name;
    this.id = id;
    System.out.println("The person " + name + " is created");
  }

  public String getName() {
    return name;
  }

  @Override
  public String toString() {
    return "Id=" + id + "\t Name=" + name;
  }
}
```

Student class

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
public class Student extends Person {

  private float average;

  public Student( int id, String name, float average) {
    super(id, name);
    this.average = average;
    System.out.println("The Student " + name + " is created");
  }

  @Override
  public String toString() {
    return super.toString() + "\t Average=" + average;
  }

  public void registerToCourse(String courseName) {
    System.out.println(name + "register to " + courseName);
  }
}
```

Array of Persons and Students

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
public static void main(String[] args) {
  Person[] persons = new Person[3];
  persons[0] = new Person(1, "Mor");
  persons[1] = new Student(2, "Kobi", 92.5f);
  persons[2] = new Person(3, "Yael");

  System.out.println("\npersons list:");
  for (int i = 0; i < persons.length; i++) {
    System.out.println(persons[i]);
  }
}
```

Console

```
The person Mor is created
The person Kobi is created
The Student Kobi is created
The person Yael is created

persons list:
Id=1 Name=Mor
Id=2 Name=Kobi Average=92.5
Id=3 Name=Yael
```

Polymorphism

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- Creating collection of common base class
- Using **instanceof** word to find out object type

```
public static void main(String[] args) {
  Person[] persons = new Person[3];
  persons[0] = new Person(1, "Mor");
  persons[1] = new Student(2, "Kobi", 92.5f);
  persons[2] = new Person(3, "Yael");

  System.out.println("\npersons list:");
  for (int i = 0; i < persons.length; i++) {
    System.out.println(persons[i]);
    if (persons[i] instanceof Student student) {
        student.registerToCourse("Java");
    }
  }
}
```

Console

```
The person Mor is created
The person Kobi is created
The Student Kobi is created
The person Yael is created

persons list:
Id=1 Name=Mor
Id=2 Name=Kobi Average=92.5
Kobi register to Java
Id=3 Name=Yael
```

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

תרגיל 1 פולימורפיזם : מוסך מכוניות

במוסך של חברה ישנם כלי רכב שונים: מכוניות, אופנועים, ומשאיות.
לכל כלי רכב יש מספר רישוי, שנת ייצור, ומהירות מקסימלית.
לכל כלי רכב יש יכולת לנסוע – אך ההתנהגות שונה לפי סוג הרכב.

צרו את המחלקה vehicle ובה השדות הנדרשים והפעולות הנדרשות.
צרו מחלקות נוספות עבור Car, Motorcycle ו-Truck.
ממשו (הדפיסו למסך) את הפעולה drive באופן הבא עבור כלי הרכב השונים:
מכונית: "Car driving smoothly on the road"
אופנוע: "Motorcycle speeding between cars"
משאית: "Truck hauling heavy cargo"

הפלט בהרצת ה-main

```
Starting drive:
Ops.. Need to implement
Car driving smoothly on the road.
Motorcycle speeding between cars!
Truck hauling heavy cargo.
```

קישור ל-starter

אתר להורדת ספריות מ-github.

Abstract class and method

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- Restricted class that cannot be used to create objects.
- To access it, you must create a class that inherits from it.

```
public abstract class Animal {
  protected String color;
  protected int numOfLegs;

  public Animal(String color, int numOfLegs) {
    this.color = color;
   this.numOfLegs = numOfLegs;
  }

  public abstract void makeNoise();

  @Override
  public String toString() {
    return getClass().getSimpleName() + ": color= " + color + ", numOfLegs = " + numOfLegs + ", ";
  }

  public String getColor() {
    return color;
  }

  public int getNumOfLegs() {
    return numOfLegs;
  }
}
```

Abstract method

Inheritance class - Horse

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
public class Horse extends Animal {
  public static final int NUM_OF_LEGS = 4;
  private int tailLen;

  public Horse(String color, int tailLen) {
    super(color, NUM_OF_LEGS);
    this.tailLen = tailLen;
  }

  @Override
  public void makeNoise() {
    System.out.println("Hi Yahah");
  }

  public void ride() {
   System.out.println("I'm riding!");
  }

  @Override
  public String toString() {
    return super.toString() + "tailLen= " + tailLen;
  }
}
```

Abstract and inheritance class - Cat

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```java
public abstract class Cat extends Animal {
  public static final int NUM_OF_LEGS = 4;
  private int whiskersLen;

  public Cat(String color, int whiskersLen) {
    super(color, NUM_OF_LEGS);
    this.whiskersLen = whiskersLen;
  }

  public void scratch() {
    System.out.println(getClass().getSimpleName() + " is scratching!");
  }

  @Override
  public void makeNoise() {
    System.out.println("Miyaoooooo");
  }

  @Override
  public String toString() {
    return super.toString() + "whiskersLen= " + whiskersLen;
  }
}
```

Inheritance class - StreetCat

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
public class StreetCat extends Cat {
  private int numOfFights;

  public StreetCat(String color, int whiskersLen, int numOfFights) {
    super(color, whiskersLen);
    this.numOfFights = numOfFights;
  }

  @Override
  public void makeNoise() {
    super.makeNoise();
    System.out.println("I want to see a dog!");
  }

  public void fight() {
    System.out.println("I want to have a good fight!");
  }

  @Override
  public String toString() {
    return super.toString() + ", numOfFights= " + numOfFights;
  }
}
```

Inheritance class - SiamiCat

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
public class SiamiCat extends Cat {
  private String favoriteFood;

  public SiamiCat(String color, int whiskersLen, String favoriteFood) {
    super(color, whiskersLen);
    this.favoriteFood = favoriteFood;
  }

  @Override
  public void makeNoise() {
    super.makeNoise();
    System.out.println("I'm so spoiled!");
  }

  @Override
  public String toString() {
    return super.toString() + ", favoriteFood= " + favoriteFood;
  }
}
```

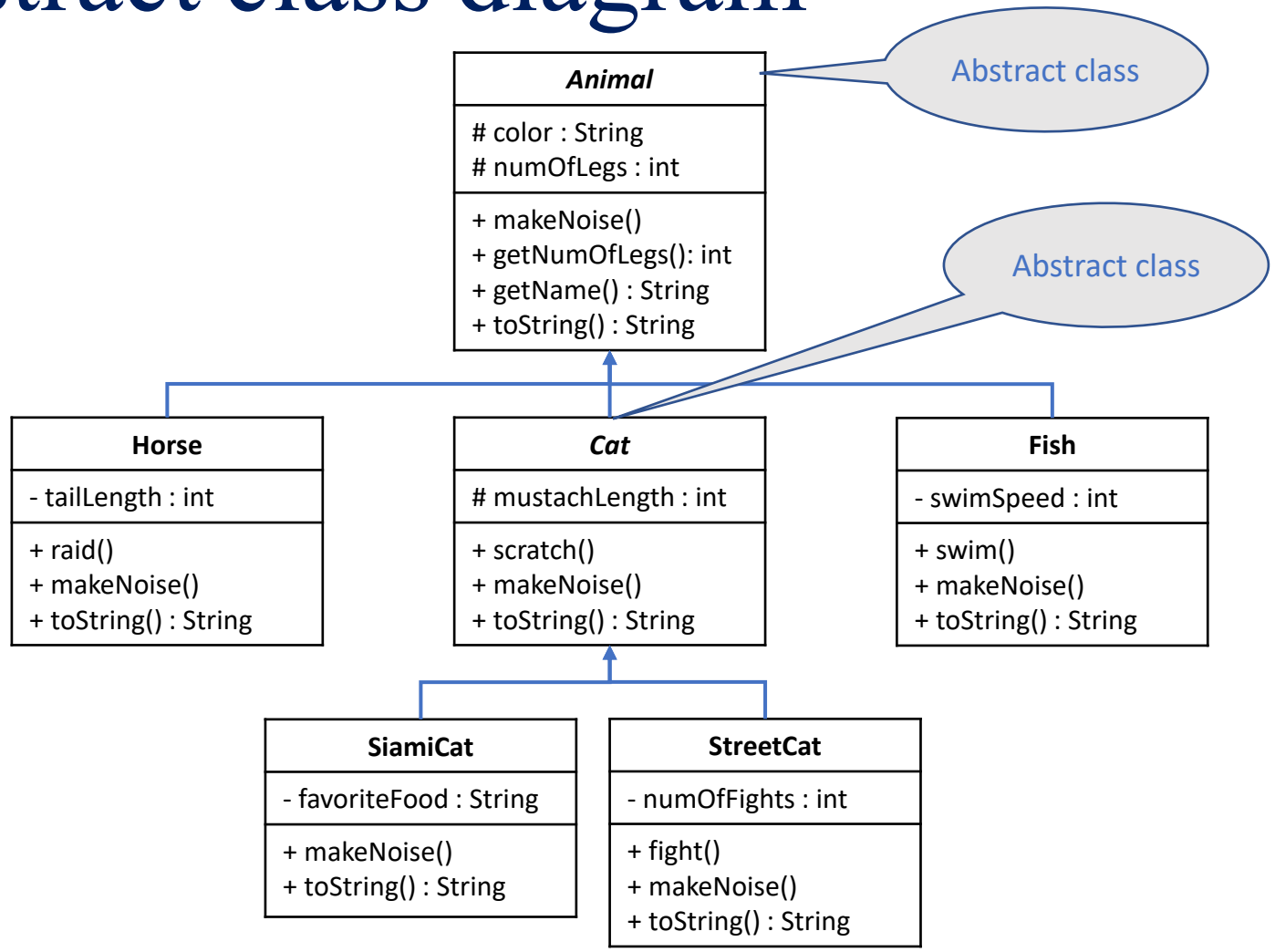
Inheritance class - Fish

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
public class Fish extends Animal {
  public static final int NUM_OF_LEGS = 0;
  private int swimSpeed;

  public Fish(String color, int swimSpeed) {
    super(color, NUM_OF_LEGS);
    this.swimSpeed = swimSpeed;
  }

  @Override
  public void makeNoise() {
    System.out.println("Blu-Blu");
  }

  @Override
  public String toString() {
    return super.toString() + "swimSpeed= " + swimSpeed;
  }
}
```

Abstract class diagram

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992



Invoke makeNoise function

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
public class Main {

public static void main(String[] args) {
  Animal[] animals = new Animal[4];
  animals[0] = new Horse("brown", 120);
  animals[1] = new SiamiCat("gray", 12, "RoyalCat");
  animals[2] = new StreetCat("Orang", 15, 55);
  animals[3] = new Fish("gold", 2);

  for (int i = 0; i < animals.length; i++) {
    System.out.println(animals[i]);
    animals[i].makeNoise();
  }
}
```

Console

```
Horse: color= brown, numOfLegs = 4, tailLen= 120
Hi Yahah
SiamiCat: color= gray, numOfLegs = 4, whiskersLen= 12, favoriteFood= RoyalCat
Miyaoooooo
I'm so spoiled!
StreetCat: color= Orang, numOfLegs = 4, whiskersLen= 15, numOfFights= 55
Miyaoooooo
I want to see a dog!
Fish: color= gold, numOfLegs = 0, swimSpeed= 2
Blu-Blu
```

Main – using instanceof

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
public class Main {

public static void main(String[] args) {
  Animal[] animals = new Animal[4];
  animals[0] = new Horse("brown", 120);
  animals[1] = new SiamiCat("gray", 12, "RoyalCat");
  animals[2] = new StreetCat("Orang", 15, 55);
  animals[3] = new Fish("gold", 2);

  for (int i = 0; i < animals.length; i++) {
    System.out.println(animals[i]);
    if (animals[i] instanceof Cat c) {
        c.scratch();
    }
  }
}
```

Console

```
Horse: color= brown, numOfLegs = 4, tailLen= 120
SiamiCat: color= gray, numOfLegs = 4, whiskersLen= 12, favoriteFood= RoyalCat
SiamiCat is scratching!
StreetCat: color= Orang, numOfLegs = 4, whiskersLen= 15, numOfFights= 55
StreetCat is scratching!
Fish: color= gold, numOfLegs = 0, swimSpeed= 2
```

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

תרגיל 2 : מה יקרה בהרצת הקוד הבא?

```
abstract class Shape {
    public abstract double area();
}

class Circle extends Shape {
    private double radius = 1.0;

    public Circle(double radius) {
        this.radius = radius;
    }
}

class Rectangle extends Shape {
    private double width = 2.0;
    private double height = 3.0;

    public Rectangle(double width, double height){
        this.width = width;
        this.height = height;
    }

    @Override
    public double area() {
        return width * height;
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Shape s1 = new Circle(5.0);
        Shape s2 = new Rectangle(2.0, 4.0);

        System.out.println("Area 1: " + s1.area());
        System.out.println("Area 2: " + s2.area());
    }
}
```

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

תרגיל 3 : מה יקרה בהרצת הקוד הבא?

```
abstract class Instrument {
    public Instrument() {
        System.out.println("Instrument created.");
    }

    public abstract void play();
}

class Piano extends Instrument {
    private int keys = 88;

    public Piano() {
        System.out.println("Piano created.");
    }

    public void play(int times) {
        System.out.println("Playing piano " + times + "
times.");
    }
}
```

```
class Guitar extends Instrument {
    @Override
    public void play() {
        System.out.println("Strumming the guitar.");
    }
}

public class Main {
    public static void main(String[] args) {
        Instrument i1 = new Piano();
        Instrument i2 = new Guitar();

        i1.play();
        i2.play();
    }
}
```

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

תרגיל 4 : מה יקרה בהרצת הקוד הבא?

```
abstract class Payment {
    protected double amount;
    public Payment(double amount) {
        this.amount = amount;
    }
    public abstract void pay();
}

class CreditCardPayment extends Payment {
    private String cardNumber;
    public CreditCardPayment(double amount, String cardNumber) {
        super(amount);
        this.cardNumber = cardNumber;
    }
    @Override
    public void pay() {
        System.out.println("Paid " + amount +
                " using Credit Card " + cardNumber);
    }
}
```

```
abstract class CashPayment extends Payment {
    public CashPayment(double amount) {
        super(amount);
    }
}
class CoinPayment extends CashPayment {
    public CoinPayment(double amount) {
        super(amount);
    }
    @Override
    public void pay() {
        System.out.println("Paid " + amount + " using Coins");
    }
}

public class Main {
    public static void main(String[] args) {
        Payment p1 = new CreditCardPayment(200.0, "1234-5678-9012-3456");
        Payment p2 = new CoinPayment(50.0);
        p1.pay();
        p2.pay();
    }
}
```

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

תרגיל 5: הצגת צורות

כתבו מערכת ליצירה והצגת צורות, המערכת תתמוך בכמה סוגים של צורות, כרגע במערכת יש עיגול וריבוע בלבד. לניהול הצורות יהיה שם ואוסף של כל הצורות כולל יצירה אקראית ראשונית של צורות, יש לוודא שלא נוצרות צורות זהות, יש לוודא **שכל** הפרמטרים במאפיינים של הצורה זהים.
כמו כן ניתן יהיה לקבל את כל ההיקפים והשטחים של הצורות במערכת.
בריבוע נשמור את עובי מסגרת הצורה, הצבע ואת גודל הצלע.
בעיגול נשמור את עובי המסגרת, הצבע ואת הרדיוס.
עבור כל צורה ניתן לקבל את שטחה ואת היקפה.
כל צורה תדע להציג את עצמה כמחרוזת עם הפרטים הרלוונטיים כולל סוג הצורה.
ריבוע יתמוך גם בהצגת הצורה כריבוע של כוכביות.

כתבו תרשים UML עבור המערכת, שימו לב , יש לתמוך בהוספה עתידית של צורות נוספות. שימו לב שכל הקוד רץ כראוי, ראו בהמשך דוגמא אקראית של פלט התוכנית.

קישור ל-starter

אתר להורדת ספריות מ-github.

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

תרגיל 5: פלט של ה-main

```
Failed to add shape Circle, frameThickness: 1, color: YELLOW, radius: 0, Error: Shape already exist

All shapes in OOP Course Shapes System
Circle, frameThickness: 1, color: YELLOW, radius: 0
Circle, frameThickness: 1, color: BLUE, radius: 1
Circle, frameThickness: 1, color: BLACK, radius: 0
Square, frameThickness: 1, color: GREEN, size: 3
* * *
* * *
* * *

Circle, frameThickness: 1, color: BLACK, radius: 2
Square, frameThickness: 1, color: BLACK, size: 4
* * * *
* * * *
* * * *
* * * *
```

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

תרגיל 5: המשך פלט של ה-main

```
All shapes Areas
Circle: 0.0
Circle: 3.141592653589793
Circle: 0.0
Square: 9.0
Circle: 12.566370614359172
Square: 16.0

All shapes Perimeters
Circle: 0.0
Circle: 6.283185307179586
Circle: 0.0
Square: 12.0
Circle: 12.566370614359172
Square: 16.0
```