Java OOP
10128

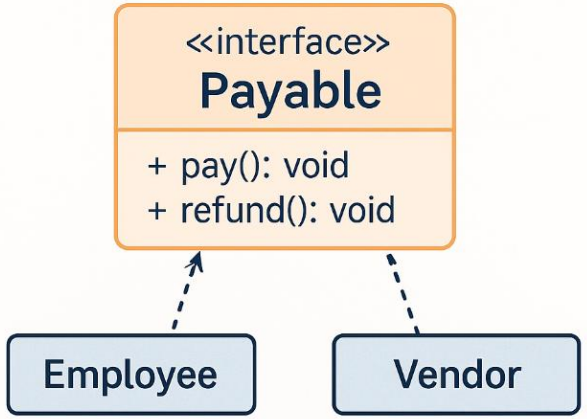
Interface

Pini shlomi

Interface

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

- An interface is a completely "abstract class" that is used to group related methods with empty bodies



- interface name usually end with 'able'
  e.g., Comparable, Cloneable , Printable

- Contains behaviors that have a shared purpose
  - Borrowable : Student and Lecturer can borrow book from library
  - Printable : Student and Circle can print themselves

Printable interface

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

```
public interface Printable {
    void print();
}

public class Person implements Printable{
  private String name;
  private int age;

  public Person(String name, Int age) {
    this.name = name;
    this.age = age;
  }

  @Override
  public void print() {
    System.out.println("name: " + name + ", age: " + age);
  }
}
```

All methods are public and abstract

Implement interface

Must Implement all interface's methods

Printable interface

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

```
public class Rectangle implements Printable{
  private int width;
  private int height;

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

  @Override
  public void print() {
    for (Int I = 0; I < height; I++) {
      for (int j = 0; j < width; j++) {
        System.out.print('*');
      }
      System.out.println();
    }
  }
}
```

Implement interface

Must Implement all interface's methods

main

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

```
public static void main(String[] rags) {
  Person p1 = new Person("Mor", 26);
  Rectangle r1 = new Rectangle(5, 8);
  p1.print();
  r1.print();
}
```

Running interface's methods

```
name: Mor, age: 26
*****
*****
*****
*****
*****
*****
*****
*****
```

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

תרגיל 1: תוכנית למימוש קלט פלט

קישור לקובץ התרגיל

קישור ל-starter

אתר להורדת ספריות מ-github.

Animal and Noiseable

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

- Animal class has makeNoise method
- Remove this makeNoise methos
- Create new Noiseable interface
- All animals inherit from Animal class
- Only those that make noise will Implement this Interface

```
public interface Noiseable {
  String getNoise();
}
```

```
public class Horse extends Animal Implements Noiseable
```

Animal class

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

```
public abstract class Animal {
  private String name;
  private String color;

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

  @Override
  public String toString() {
    StringBuffer sb = new StringBuffer(getClass().getSimpleName());
    sb.append(": ").append(name).append(", ").append(color);
    return sb.toString();
  }
}
```

Horse class

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

```
public class Horse extends Animal implements Noiseable{
    private int height;

    public Horse(String name, String color, int height) {
        super(name, color);
        this. Height = height;
    }

    public void ride() {
        this. Height("I'm riding...");
    }

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

    @Override
    public String getNoise() {
        return "Hayaam";
    }
}
```

Cat class

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

```
public class Cat extends Animal implements Noiseable {
    private double whiskers Len;

    public Cat(String name, String color, double whiskers Len) {
        super(name, color);
        this. Whiskers Len = whiskers Len;
    }

    @Override
    public String toString() {
        return super.toString() + ", " + whiskers Len;
    }

    @Override
    public String getNoise() {
        return "Miyamoto";
    }
}
```

Fish class

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

```
public class Fish extends Animal{

    public Fish(String name, String color) {
        super(name, color);
    }

    public void swim() {
        this. Height("I'm swimming");
    }
}
```

main

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

```
public static void main(String[] rags) {
    Animal[] animals = new Animal[3];
    animals[0] = new Cat("Pits", "Brown", 5.7);
    animals[1] = new Fish("Digi", "gold");
    animals[2] = new Horse("Davi", "Black", 184);

    for (int I = 0; I < animals.length; I++) {
        System.out.println(animals[I]);
        if (animals[I] instanceof Noiseable) {
            System.out.println(animals[I].getClass().getSimpleName()
            + ": "
            + ((Noiseable)animals[I]).getNoise());
        }
    }
}
```

Check interface

```
Cat: Pitzi, Brown, 5.7
Cat: Miyaooooo
Fish: Dagi, gold
Horse: Davi, Black, 184
Horse: Hiyaaaa
```

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

תרגיל 2 : מה יקרה בהרצת הקוד הבא?

```
class ConnectionException extends Exception { }

interface Connectable {
    void connect() throws ConnectionException;
}
class UsbDevice implements Connectable {
    @Override
    public void connect() {
        System.out.println("USB connected");
    }
}
public class Main {
    public static void main(String[] args) {
        new UsbDevice().connect();
    }
}
```

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

תרגיל 3 : מה יקרה בהרצת הקוד הבא?

```
class TemperatureException extends RuntimeException {
    TemperatureException(String msg) { super(msg); }
}

interface Heatable {
    void heat(int degrees);
}

class Oven implements Heatable {
    private static final int MAX = 250;

    @Override
    public void heat(int degrees) {
        if (degrees > MAX) {
            throw new TemperatureException("Too hot!");
        }
        System.out.println("Heating to " + degrees);
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Heatable h = new Oven();
        h.heat(180);
        h.heat(260);
    }
}
```

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

תרגיל 4: מה יקרה בהרצת הקוד הבא?

```
interface Printable {
    String show();
}
abstract class Document implements Printable {
    protected abstract String getContent();
}
class Report extends Document {
    @Override
    public String show() {
        return getContent();
    }

    @Override
    protected String getContent() {
        return "Quarterly Report";
    }
}
```

```
public class Main {
    public static void main(String[] args){
        Printable p = new Report();
        System.out.println(p.show());
    }
}
```

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

תרגיל 5 : מה יקרה בהרצת הקוד הבא?

```
interface Startable {
    void start() throws EngineException;
}

class EngineException extends Exception { }

class Bicycle implements Startable {
    @Override
    public void start() {
        System.out.println("Pedaling...");
    }
}
public class Main {
    public static void main(String[] args) {
        Startable s = new Bicycle();
        s.start();
    }
}
```

Interface inherit

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

```
public interface Runable {
        void run();
}

public interface Swimmable {
        void swim();
}

public interface Rideable {
        void ride();
}

public interface BetterSwimmable extends Swimmable {
        void fastSwim();
}

public interface Athletable extends BetterSwimmable, Rideable, Runable{
        void breath();
}
```

Interface extends

Extends more then one

Inherit and implements

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

```
public class Athlet extends Person implements Athletable{

    public Athlet(String name, int age) {
        super(name, age);
    }
    @Override
    public void swim() {
        System.out.println("I'm swimming");
    }
    @Override
    public void fastSwim() {
        System.out.println("I'm fast swimming");
    }
    @Override
    public void ride() {
        System.out.println("I'm riding");
    }
    @Override
    public void run() {
        System.out.println("I'm running");
    }
    @Override
    public void breath() {
        System.out.println("I Have good breathing");
    }
}
```

First extends and after interfaces

Interface default methods

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

```
interface Alpha {
    default void hello() {
        System.out.println("Alpha");
    }
}
interface Beta {
    default void hello() {
        System.out.println("Beta");
    }
}
class Gamma implements Alpha, Beta {
    @Override
    public void hello() {
        Alpha.super.hello();
        Beta.super.hello();
        System.out.println("Gamma says hello!");
    }
}
```

```
class Delta implements Alpha {}

public class Main {
    public static void main(String[] args) {
        new Gamma().hello();
        new Delta().hello();
    }
}
```

```
Alpha
Beta
Gamma says hello!
Alpha
```

Interface static method

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

```
interface MyInterface {
    static void staticMethod() {
        System.out.println("This is a static method in the interface.");
    }
}

public class Main {
    public static void main(String[] args) {
        MyInterface.staticMethod();

    }
}
```

This is a static method in the interface.

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

תרגיל 6 : מה יקרה בהרצת הקוד הבא?

```
interface Readable {
    void read() throws IOException;
}
interface Writable {
    void write() throws IOException ;
}
interface Loggable extends Writable {}
interface ReadableWritable extends Readable, Loggable {}
class Device implements ReadableWritable {
    @Override
    public void read() {
        System.out.println("read");
    }

    @Override
    public void write() {
        System.out.println("write");
    }
}
```

```
public class Main {
    public static void main(String[] args)
                        throws IOException {
        ReadableWritable device = new Device();
        device.read();
        device.write();
        }
}
```

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

תרגיל 7: תוכנית להכנת פיצה

קישור לקובץ התרגיל

קישור ל-starter

אתר להורדת ספריות מ-github.