Enumerated Types

תרכז 1 בשפת יאוה - סמסטר א"י תש"י
אוניברסיטת תל אביב
Enumerated Types

.package cards.domain;

.public class PlayingCard {

    // pseudo enumerated type
    public static final int SUIT SPADES = 0;
    public static final int SUIT HEARTS = 1;
    public static final int SUIT CLUBS = 2;
    public static final int SUIT DIAMONDS = 3;

    private int suit;
    private int rank;

    public PlayingCard(int suit, int rank) {
        this.suit = suit;
        this.rank = rank;
    }
}
public String getSuitName() {
    String name = "";
    switch (suit) {
    case SUIT_SPADES:
        name = "Spades";
        break;
    case SUIT_HEARTS:
        name = "Hearts";
        break;
    case SUIT_CLUBS:
        name = "Clubs";
        break;
    case SUIT_DIAMONDS:
        name = "Diamonds";
        break;
    default:
        System.err.println("Invalid suit.");
    }
    return name;
}
Enumerated Types

consistent with φονή and overall vs. 1 and is not consistent with φονή

```java
package cards.tests;

import cards.domain.PlayingCard;

public class TestPlayingCard {
    public static void main(String[] args) {
        PlayingCard card1 = new PlayingCard(PlayingCard.SUIT_SPADES, 2);
        System.out.println("card1 is the \" + card1.getRank() + \" of \\
        + card1.getSuitName());
        // You can create a playing card with a bogus suit.
        PlayingCard card2 = new PlayingCard(47, 2);
        System.out.println("card2 is the \" + card2.getRank() + \" of \\
        + card2.getSuitName());
    }
}
```
Enumerated Types

ל ObjectType מינה בצורה זו כמה הסרונות:

- Not typesafe
- Ainu Shomar al bizitou tiposim
- Ainu Shomar al marhab shomot
- Minou Shebriri
- Urek ha'dofsa ainu me'mouot
New Enumerated Types

enum в Java 5.0 добавил возможность создания обобщенных

использование типа enum позволяет решить проблему уникальности

public enum Suit {
    SPADES,
    HEARTS,
    CLUBS,
    DIAMONDS
}
package cards.domain;

public class PlayingCard2 {

    private Suit suit;
    private int rank;

    public PlayingCard2(Suit suit, int rank) {
        this.suit = suit;
        this.rank = rank;
    }

    public Suit getSuit() {
        return suit;
    }
}
New Enumerated Types

```java
public String getSuitName() {
    String name = "";
    switch (suit) {
    case SPADES:
        name = "Spades";
        break;
    case HEARTS:
        name = "Hearts";
        break;
    case CLUBS:
        name = "Clubs";
        break;
    case DIAMONDS:
        name = "Diamonds";
        break;
    default:
        assert false : "ERROR: Unknown type!";
        break;
    }
    return name;
}
```
package cards.tests;

import cards.domain.PlayingCard2;
import cards.domain.Suit;

public class TestPlayingCard2 {
    public static void main(String[] args) {

        PlayingCard2 card1 = new PlayingCard2(Suit.SPADES, 2);
        System.out.println("card1 is the "+ card1.getRank() + " of " + card1.getSuitName());

        // PlayingCard2 card2 = new PlayingCard2(47, 2);
        // This will not compile.
    }
}

New Enumerated Types

המענה המודל פותר את ב- Java כדבר חום לצה - על ק. החיבור
בעדות הביניים או לא גמרא הקונספט של סה"כ laten מונח
את שאר הבientes את מחלפת (עמ"ש שדות, מתודות, בנאים...)
חוכנה 1 בשפת יואוה
סיופס מגייה כמותלקלת

package cards.domain;

public enum Suit {
    SPADES("Spades"),
    HEARTS("Hearts"),
    CLUBS("Clubs"),
    DIAMONDS("Diamonds");

    private final String name;

    private Suit(String name) {
        this.name = name;
    }  

    public String getName() {
        return name;
    }  
}
package cards.tests;

import cards.domain.PlayingCard2;
import cards.domain.Suit;

public class TestPlayingCard3 {
    public static void main(String[] args) {

        PlayingCard2 card1 = new PlayingCard2(Suit.SPADES, 2);
        System.out.println("card1 is the " + card1.getRank() + " of " + card1.getSuit().getName());

        // NewPlayingCard2 card2 = new NewPlayingCard2(47, 2);
        // This will not compile.
    }
}
public enum ArithmeticOperator {  
    // The enumerated values
    ADD, SUBTRACT, MULTIPLY, DIVIDE;

    // Value-specific behavior using a switch statement
    public double compute(double x, double y) {
        switch (this) {
            case ADD:       return x + y;
            case SUBTRACT:  return x - y;
            case MULTIPLY:  return x * y;
            case DIVIDE:    return x / y;
            default: throw new AssertionError(this);
        }
    }
}
public enum ArithmeticOperator {
    ...

    // Test case for using this enum
    public static void main(String args[]) {
        double x = Double.parseDouble(args[0]);
        double y = Double.parseDouble(args[1]);
        for (ArithmeticOperator op : ArithmeticOperator.values()) {
            System.out.printf("%f %s %f = %f\n", x, op, y, op.compute(x, y));
        }
    }
}
public enum ArithmeticOperator2 {

    ADD {
        public double compute(double x, double y) {
            return x + y;
        }
    },

    SUBTRACT {
        public double compute(double x, double y) {
            return x - y;
        }
    },

    MULTIPLY {
        public double compute(double x, double y) {
            return x * y;
        }
    },

    DIVIDE {
        public double compute(double x, double y) {
            return x / y;
        }
    };

    public abstract double compute(double x, double y);
}
לעיות לעצמים יש 무' מאפיינים/תכונות

לובר כל עצ יכולת להتكنיה של התכונות, החלק

אף אוח עשת מנה

למשל צורה גיאומטרית יכולה להיה

 времени, צורה, מלאה, חלולה, ישנה, עגולה, בעונו...

איך התים מייצבים אתה?
Bit Flags

Dr. Ahnat – Shomor meshugat ha-Bolyan lekhal miafin

boolean isConvex;
boolean isFull;
...

Bit Flags - Shniha – Meshugat v-

int shapeAttributes;
boolean isConvex() {
    return shapeAttributes & 0x02;
}

7 6 5 4 3 2 1 0

Tovena 1 be-shafet la-avoh

16
EnumSet

בג'אواו 5 נסף מימוע חודי ל- Set- המבוסס על Enum

cל הערכין יסתיجموعים לעב קו- מתנה ומוגדר
כבר, או זה המוגדר ביצירתו הסט
פגימיה, הערכין שהוקדקסים בטיס, 2"א מאד
יילימ
import java.util.*;

def enum ShapeAttributes {
    CONVEX, FULL, STRAIGHT, COLORED
}

def public class Testing {
    public static void main(String[] args) {
        def Set<ShapeAttributes> s1 = EnumSet.of(ShapeAttributes.COLORED);
        def if (s1.contains(ShapeAttributes.CONVEX))
            def System.out.println("S1 is convex");

        def Set<ShapeAttributes> s2 = EnumSet.of(ShapeAttributes.CONVEX,
                                             ShapeAttributes.FULL);
        def if (s2.contains(ShapeAttributes.CONVEX))
            def System.out.println("S2 is convex");

        def Set<ShapeAttributes> s3 = EnumSet.allOf(ShapeAttributes.class);
        def System.out.println(s3);
    }
}
enum Colors {
    RED, GREEN, BLUE, YELLOW
}

public class Testing {
    public static void main(String[] args) {
        Map<Colors, String> m = new EnumMap<Colors, String>(Colors.class);
        m.put(Colors.RED, "Red");
        m.put(Colors.BLUE, "Blue");
        System.out.println(m);
    }
}

EnumMap

EnumSet אחותי ההוריה של
חג שמח!