**מטלה 3 – ירושה, פולימורפיזם ואופרטורים**

**הערות חשובות:**

**במסגרת מטלה זו – אין לבדוק/לזרוק חריגות!**

**ניתן (ואף נדרש) להוסיף מתודות/אופרטורים חדשים למחלקות אף על פי שלא מסופקת מראש החתימה שלהן.**

**יש לממש את כל המתודות בקובץ אחד: allClasses.cpp.**

**בניית עץ ההורשה -**

במטלה זו עלייך לבנות מערכת לניהול צורות (מרובע, ריבוע ומעגל).

עץ ההורשה נראה כך:

Shape

Quad Circle

Square

מחלקת Shape הינה מחלקה אבסטרקטית. לצורה יש שם (string), מרכז (אובייקט נקודה), ומספר הצורות הכולל שנוצרו (משתנה static).

למחלקת Quad יש 4 אורכי צלעות: up,down,left,right

למחלקת Square אין תוספות.

למחלקת Circle יש רדיוס: radius.

יש להגדיר המתודות והאופרטורים הבאים במחלקות הנדרשות (קראו היטב את התיעוד):

במחלקת Point:

//To implement:

int getX() const;//returns the x

int getY() const;//returns the y

void setX(int x);//sets the x

void setY(int y);//sets the y

void setPoint(int x, int y);//sets the x and y

void setPoint(const Point & other);//sets the point with other values

bool operator==(const Point & other) const;//returns true if and only if the two

points are equal

bool operator!=(const Point & other) const;//returns true if and only if the two

points are not equal

Point operator+(int num) const;//returns a new point with updated x(x+num) and

y(y+num)

Point & operator+=(int num);//updates the point with x(x+num) and y(y+num) and returns

the updated point

operator int() const;//returns the sum of x and y

Point operator+(const Point & other) const;//adds cordinats (x,y) and returns a new

point

Point operator++(int);//adds 1 to x and y and returns the updated point - postfix

Point & operator++();//adds 1 to x and y and returns the updated point - prefix

friend Point operator\*(int num, const Point & other);//rturns a new point with (x,y)\*num

במחלקת Shape:

//To implement:

virtual ~Shape();//shape destructor

Shape(string sn = "noName");//Creates a new shape (name=sn,centerPoint = (0,0))

Shape(const Shape & other);//copy constructor of shape

void setName(string name);//sets the shape's name

void setCenter(const Point & p);//sets the center point

string getName() const;//returns the shape's name

Point getCenter() const; //returns the center point

void setShape(string sn, const Point & other);//set the shape's name and the center point

static int numOfShapes();//returnS the total number of shapes

//Pure virtual methos:

virtual int calcArea() const = 0;//returns the shape's area (rounded to int)

virtual int calcPerimeter() const = 0;//returns the shape's Perimeter (rounded to int)

במחלקת Quad:

//To implement:

Quad(double up=0,double down=0,double right=0,double left=0,string sn =

"noName");//constructor. center point = (0,0), name=sn

Quad(const Quad & other);//copy constructr

virtual ~Quad();//destructor

virtual int calcArea() const;//returns the quad's area (right\*up) (rounded to int)

virtual int calcPerimeter() const;//returns the quad's Perimeter (the sum of the four ribs) (rounded to int)

במחלקת Square:

//To implement:

Square(double up = 0, double down = 0, double right = 0, double left = 0, string sn =

"noName");//constructor. center point = (0,0), name=sn

virtual ~Square();//destructor

void shiftX(int x);//shifts the center of the square by x (right and left)

void shiftY(int y);//shifts the center of the square by y (up and down)

במחלקת Circle:

//To implement:

Circle(double r = 1, string sn = "noName");//create a circle with radius=1, name=sn

and center=(0,0)

Circle(const Circle & other);//copy constructor

virtual ~Circle();//destructor

double getRadius() const;//returns the circle's radius

void setRadius(double r);//set the radius

virtual int calcArea() const;//returns the circle's area (rounded to int)

virtual int calcPerimeter() const;//returns the circle's Perimeter (rounded to int)

//operators:

friend Circle operator+(int r,const Circle & other);//returns a new Circle with added

radius

Circle & operator=(string name);//updates the name of the shape and returns the updated shape

**בניית ה-container –**

מחלקת allShapes מכילה: מערך פולימורפי של צורות ומספר הצורות.

יש לממש את המתודות והאופרטורים הבאים (קראו היטב את התיעוד):

//To implement:

int getSize() const;//returns the number of shapes;

allShapes();//constructor, zero elements

~allShapes();//destructor to all elements

void addShape(Shape \* newShape);//add a new shape (any) to the end of array

void removeShape(int index);//remove shape by index

int totalArea() const; //returns the total area of all the shapes (rounded

to int)

int totalPerimeter() const;//returns the total Perimeter of all the shapes (rounded

to int)

int totalCircleArea() const;//returns the total sum of circle areas (rounded

to int)

int totalSquarePerimeter() const;//returns the total sum of square Perimeter

(rounded to int)

//operators:

allShapes & operator+=(Shape \* newS);//adds a new shape

Shape \* operator[](int ind) const; //returns the shape located at index ind

allShapes operator+(const allShapes & other) const;//returns a new allShape

with all elements located at this and other

operator int() const;//returns the number of elements