File - V:\ITMO\2 course\web-programming\lab3-20.10.19\src\META-INF\persistence.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence" version="2.0">

    <persistence-unit name="db_con">
        <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
        <class>Point</class>
        <properties>
            <property name="eclipselink.jdbc.url" value="jdbc:postgresql://localhost:5432
/web3"/>
            <property name="eclipselink.jdbc.driver" value="org.postgresql.Driver"/>
            <property name="eclipselink.jdbc.user" value="postgres"/>
            <property name="eclipselink.jdbc.password" value="root"/>
        </properties>
    </persistence-unit>
</persistence>
```

```java
import javax.persistence.*;
import java.util.Date;

@Entity
@Table(name = "points")
public class Point {

    @SequenceGenerator(name = "seqGen", sequenceName = "gen_seq", allocationSize = 1)
    @Id @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "seqGen") private int id;
    private String owner;
    private double x, y, r;
    private boolean coordsStatus;
    private Date bornDate;

    public Point(String owner, double x, double y, double r) {
        this.owner = owner;
        this.x = x;
        this.y = y;
        this.r = r;
        coordsStatus = checkCoordinates(x, y, r);
        bornDate = new Date();
    }

    public Point() {}

    private boolean checkCoordinates(double x, double y, double r) {
        return (x <= 0 && y >= 0 && x >= -r && y <= r/2) || (x >= 0 && y >= 0 && y <= (x - r/2) * (-2)) ||
                (x >= 0 && y <= 0 && x * x + y * y <= Math.pow(r, 2));
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof Point)) return false;
        Point point = (Point) o;
        return Double.compare(point.x, x) == 0 &&
                Double.compare(point.y, y) == 0 &&
                Double.compare(point.r, r) == 0 &&
                coordsStatus == point.coordsStatus;
    }

    @Override
    public String toString() {
        return "<tr><td>" + x + "</td>" +
                "<td>" + y + "</td>" +
                "<td>" + r + "</td>" +
                "<td style='color: " + ((coordsStatus) ? "green" : "red") + "'>" + coordsStatus + "</td>" +
                "<td>" + bornDate + "</td></tr>";
    }

    public long getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getOwner() {
        return owner;
    }

    public void setOwner(String owner) {
        this.owner = owner;
    }

    public double getX() {
        return x;
    }

```

```
71      public void setX(double x) {
72          this.x = x;
73      }
74
75      public double getY() {
76          return y;
77      }
78
79      public void setY(double y) {
80          this.y = y;
81      }
82
83      public double getR() {
84          return r;
85      }
86
87      public void setR(double r) {
88          this.r = r;
89      }
90
91      public boolean isCoordsStatus() {
92          return coordsStatus;
93      }
94
95      public void setCoordsStatus(boolean coordsStatus) {
96          this.coordsStatus = coordsStatus;
97      }
98
99      public Date getBornDate() {
100         return bornDate;
101     }
102
103     public void setBornDate(Date bornDate) {
104         this.bornDate = bornDate;
105     }
106 }
```

```java
import javax.faces.context.FacesContext;
import javax.persistence.*;
import java.io.Serializable;
import java.util.List;
import java.util.Map;

public class RealHumanBean implements Serializable {

    private static final long serialVersionUID = 4L;
    private EntityManagerFactory managerFactory;
    private EntityManager manager;
    private EntityTransaction transaction;
    private List<Point> points;

    public void validate() {
        try {
            FacesContext facesContext = FacesContext.getCurrentInstance();
            managerFactory = Persistence.createEntityManagerFactory("db_con");
            manager = managerFactory.createEntityManager();
            transaction = manager.getTransaction();
            Map<String, String> params = facesContext.getExternalContext().
getRequestParameterMap();
            Point point = new Point(facesContext.getExternalContext().getSessionId(true
), Double.parseDouble(params.get("X-value")),
                    Double.parseDouble(params.get("Y-value")), Double.parseDouble(params.
get("R-value")));
            addPointToDB(point);
            points = getAllEntitiesFromDB(facesContext.getExternalContext().getSessionId(
true));
        } finally {
            manager.close();
            managerFactory.close();
        }
    }

    private void addPointToDB(Point addedPoint) {
        transaction.begin();
        manager.persist(addedPoint);
        transaction.commit();
    }

    private List<Point> getAllEntitiesFromDB(String sessionID) {
        transaction.begin();
        TypedQuery<Point> query = manager.createQuery("SELECT p FROM Point p WHERE p.
owner = :owner", Point.class);
        return query.setParameter("owner", sessionID).getResultList();
    }

    public List<Point> getPoints() {
        return points;
    }

    public void setPoints(List<Point> points) {
        this.points = points;
    }
}
```