Федеральное агентство связи

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра Математической кибернетики и информационных технологий

Отчет по лабораторной работе №3

по дисциплине «Технологии разработки программного обеспечения»

Выполнил: студент группы БВТ1801

Протасова Елена Сергеевна

руководитель:

Мосева Марина Сергеевна

Москва 2020

Цель работы: поиск пути от начального местоположения до пункта назначения с успешным преодолением препятствий.

Выполнение работы:

Переопределение методов в Location:

```
@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;
    Location location = (Location) o;
    return xCoord == location.xCoord &&
            yCoord == location.yCoord;
}
@Override
public int hashCode() {
    return Objects.hash(xCoord, yCoord);
Написание методов в AStarState:
import java.util.HashMap;
public class AStarState
    private Map2D map;
    HashMap<Location, Waypoint> openTop = new HashMap<>();
    HashMap<Location, Waypoint> closeTop = new HashMap<>();
    public AStarState(Map2D map)
    {
        if (map == null)
            throw new NullPointerException("map cannot be null");
        this.map = map;
    }
    public Map2D getMap()
    {
        return map;
    }
    public Waypoint getMinOpenWaypoint()
        // TODO: Implement.
        float cost = Float.MAX_VALUE;;
        Waypoint remeber = null;
        for (Waypoint value : openTop.values()) {
            if (value.getTotalCost()<=cost)</pre>
            {
                cost = value.getTotalCost();
                remeber = value;
            }
        if (cost == Float.MAX VALUE) return null;
        else return remeber;
    }
        public boolean addOpenWaypoint(Waypoint newWP)
    {
```

```
// TODO: Implement.
    if (!openTop.containsKey(newWP.loc))
    {
        openTop.put(newWP.loc,newWP);
        return true;
    }
    else if (newWP.getPreviousCost()<openTop.get(newWP.loc).getPreviousCost())</pre>
        openTop.put(newWP.loc,newWP);
        return false;
    }
    else
        return false;
}
    public int numOpenWaypoints()
{
    int num = openTop.size();
    // TODO: Implement.
    return num;
}
public void closeWaypoint(Location loc)
    // TODO: Implement.
    closeTop.put(loc,openTop.remove(loc));
}
public boolean isLocationClosed(Location loc)
    // TODO: Implement.
    return closeTop.containsKey(loc);
}
```