# 03. Halloween



## Preparation

Download the skeleton provided in Judge. **Do not** change the **packages**!

**Pay attention to name the package halloween, all the classes, their fields, and methods the same way they are presented in the following document. It is also important to keep the project structure as described.**

## Problem description

Your task is to create a repository that stores items by creating the classes described below.

### Kid

First, write a Java class **Kid** with the following fields:

* **name: String**
* **age: int**
* **street: String**

The class **constructor** should receive a **name,** an **age,** andthe **street** it’s from**.** You need to create the appropriate **getters and setters**. The class should override the **toString()** method in the following format:

**"{name}, {age} years old, from {street}"**

### House

**Next**, write a Java class **House** with **data** (a collection that stores the **Kids**). All entities inside the repository have the **same fields**. Also, the **House** class should have those fields:

* **capacity: int**
* **data: List<Kid>**

The class **constructor** should receive **capacity**. Also, it should initialize the **data** with a new collection instance**.** Implement the following features:

* Method addKid(Kid kid) – **adds** an **entity** to the data **if** **there** **is** an **empty space** for the kid.
* Method removeKid(String name) – removes the kid by **given name,** if such **exists**, and **returns boolean**.
* Method **getKid(String street)** – returns the kid **of a given street** or **null if no such kid exists**.
* Getter getAllKids() – **returns** the **number** of kids.
* **getStatistics()** – **returns** a **String** in the following **format**:

**"** **Children who visited a house for candy:  
 {name} from {street} street  
 {name} from {street} street  
 (…)**"

## Constraints

* The **age** of the kids will always be a **positive number**.

## Examples

This is an example of how the **House** class is **intended to be used**.

|  |
| --- |
| Sample code usage |
| *// Initialize the repository* House house = new House(6);  *// Initialize entity* Kid firstKid = new Kid("Yoana", 10, "Osogovska"); *// Print Кid* System.*out*.println(firstKid); *// Yoana, 10 years old, from Osogovska  // Add First Кid* house.addKid(firstKid);  *// Remove Kid* System.*out*.println(house.removeKid("Sara")); *//false* System.*out*.println(house.removeKid("Yoana")); *//true* Kid secondKid = new Kid("Lily", 5, "Pirina"); Kid thirdKid = new Kid("Anna", 7, "Pirotska"); Kid fourthKid = new Kid("Selin", 8, "Solunska");  house.addKid(secondKid); house.addKid(thirdKid); house.addKid(fourthKid);  Kid kid = house.getKid("Pirotska");  System.*out*.println(kid); *// Anna, 7 years old, from Pirotska  // All kids in house* System.*out*.println(house.getAllKids()); *// 3  // Information Statistics* System.*out*.println(house.getStatistics());  *//Children who visited a house for candy: //Lily from Pirina street // Anna from Pirotska street // Selin from Solunska stret* |

## Submission

Submit a **single .zip file** containing the **halloween package, with the classes inside (Kid, House, and the Main class)**, and there is no specific content required inside the Main class e. g. you can do any local testing of your program there. However, there should be a **main(String[] args)** method inside.