**Класс хоорондын харьцаа**

Лаб №9

Д.Самдан (16B1SEAS2873)

ХШУИС, ПХ-ийн 3-р түвшин, pri.samdan@gmail.com

**1. ОРШИЛ**

Үндсэн хүн класс байгуулаж түүнээс удамшуулан ажилчин, гэрэлсэн хүн, хүүхэд зэрэг классуудыг үүсэгсэн. Мөн ажилчины ажилах хэлтэс, хийдэг ажилын мэдээлэл зэрэг кдассулыг үүсэгсэн. Эдгээр классиудын харьцааг судалж. Ажилчин класс болон бусад класс хоорондын харьцааг програмчилсан.

**2. ЗОРИЛГО**

Класс хоорондын харьцааны талаар судлах. Ажилчин классид эхнэр, хүүхэд, ажилын мэдээлэл, хэлтэс зэрэг классыг зав харьцаанд оруулах. Үүний тулд дараах зорилтуудыг тавьж ажилласан:

1. ArrayList-ийн талаар судах.
2. Классын харьцааны талаар судах.
3. One to One
4. One to Many
5. Классуудыг байгуулах.
   1. Хүн класс
   2. Ажилчин класс
   3. Spouse класс
   4. Хэлтэс класс
   5. Ажилын мэдээлэл
6. Байгуулсан класс хоородын харьцааг програмчилах.

**3. ОНОЛЫН СУДАЛГАА**

**ArrayList**

Java collection-ийн List-ийн доор оршидог юм. ArrayList нь динамик хүснэгт ба түүний хэмжээн нь елеметнмийн тоогоос хамаарч өсөж, багсна. Хүснэгтээс елемент авах, хэмжээгийн харах зэрэг үйлдэлүүдийг маш хурдан гүйцэгдэг. Элемент нэмхэд O(n) хугцаа зарцуулдаг[1].

Жишээ:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | **import** **java.util.ArrayList**;  **import** **java.util.Collection**;  ArrayList<Integer> a = **new** ArrayList<Integer>();  // Элемент нэмэх  a.add(**5**);  // Хүснэгтийн хэмжээг харах  a.size();  // Тухайн инлекс дэх элементийг авах  a.get(**0**);  // Тухайн инлекс дэх элементийг утгах  a.remove(**0**); |

**Класс хоорондын харьцаа**

Нэг классын гишүүн өгөгдөлд заваамал төрөлөөс гадна. Өөрийн тодорхойлж өгсөн өгөгдөлийн төрөл буюу үүсмэл классыг ашиглаж болно.

**Нэг – Нэг, Тэг – Нэг**

Энэ харьцаан нь нэг класс дотор өөр нэгэн классын объект нэг эсвэл тэг байхыг хэлнэ. Энэ хоёр харьцааг үүсэгхэд нэлээд төстэй ба нэг – нэг харьц нь байгуулагч функц ашиглан үүсгнэ.

Жишээ: Тэг – Нэг (Нэг ажилчин тэг эсвэл нэг нөхөр, эхнэртэй байна)

|  |  |
| --- | --- |
| 1  2  3 | **class** **Employee** {  Spouse s;  }; |

Жишээ: Нэг – Нэг (Нэг ажилчин нэг хэлтэстэй байна)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | **class** **Employee** {  Division d;  Employee(Division d) {  **this**.d = d;  }  }; |

Дээрэх жишээ нь хэлтэсийн заалт үүсгэн, түүнд байгуулагч функцид орж ирсэн хэлтэсээ өгнө.

**Нэг – Олон, Тэг – Олон**

Энэ харьцаан нь нэг классит өөр нэгэн класс хоёр болон түүнээс дээш объектийг хадаглахыг хэлнэ. Мөн нэг – олон, тэг – олонг үүсгхэд Java хэлэнд зааяамал байдаг ArrayList буюу динамик хүнэгтийг ашиглна. Нэг – олонг үүсгэхэд байгуулагч функц ашиглан үүсгнэ.

Жишээ:

Нэг ажилчин тэг болон түүнээс олон хүүхэдтэй байна

Нэг ажилчин нэг болон түүнээс олон ажилийн мэдээлэлтэй байна.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | **public** **class** **Employee** {  **public** ArrayList<Child> children;  **public** ArrayList<JobDescription> jobDescriptions;    Employee(JobDescription jobDescriptions) {  **this**.jobDescriptions = jobDescriptions;  }  } |

**4.Хэрэгжүүлэлт**

**Person.java** энэ файлд хүн гэсэн хамгийн ерөнхий класс байна. Үүнд нэр, регистэрийн дугаар, нас зэрэг гишүүн өгөгдөлтэй ба түүнийг өөрчилөх GET, SET функц байна.

**Child.java** – Хүүхэд класс нь хүн классаас удамшисан ба түүний хамгийн дуртай тоглоомыг агуулсан. Мөн дуртай тогоомийг авах, өөрчилөх GET, SET функцтай.

**Spouse.java** – Эхнэр, нөхөр класс нь хүн классаас удамшисан ба гэрлэлтйин өдөрийг агуулсан. Мөн гэрлэлтийн өдөрийг авах, өөрчилөх GET, SET функцтай.

**Division.java** – Хэлтэс класс нь тухай нь хэлтэсийн нэрийг агуулсан ба түүний утгыг нь авах GET, өөрчилөх SET функцтай байна.

**JobDescription.java** – Ажилийн мэдээлэл класс нь ажилийн мэдээлэл агуулсан ба түүний утгыг нь авах GET, өөрчилөх SET функцтай байна.

**Employee.java** – Ажилтан класс нь хүн классаас удамшисан ба ажичины дугаар, ажилчины цол, ажилд орсон огноо зэрэг өгөгдөлийн заяамал төрөлийн гишүүн өгөдөлтэй ба тэдгээрийн утгыг нь авах GET, солих SET функуудтай. Мөн дээр үүсэгсэн классуудтай хамааралтай юм. Ажилчин тэг эсвэл нэг нөхөр, эхнэртэй байна. Ажилчин тэг болон олон хүүхэдтэй байж болно, энэ харьцааг ArrayList-ийг ашиглан үүсэгсэн. Ажилчин заавал нэг хэлтэсд оршиж байх ёстой ба түүний байгуулагч функц ашиглан харьцаагын үүсэгсэн. Ажилчин нэг болон олон ажилын мэдээлэлтэй байж болох ба түүнийг байгуулагч функц, ArrayList ашиглан ажилчин, ажилийн мэдээлэл класс хоорондийн харьцааг бий болгосон.

**5. ДҮГНЭЛТ**

б

**Номзүй**

1. Class ArrayList<E>, <https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>

**ХАВСРАЛТ**

**Main**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | **package** com.company;  **import** **java.util.Date**;  **public** **class** **Main** {  **public** **static** **void** **main**(String[] args) {  Division marketing = **new** Division("Marketing");  Division finance = **new** Division("Finance");  Division reseacher = **new** Division("Researcher");  JobDescription calculate = **new** JobDescription("Calculate salary");  JobDescription tell = **new** JobDescription("Tell other people about company");  JobDescription find = **new** JobDescription("Find the best possible solution for company");  Employee a = **new** Employee("Sam Dan", "AAB1", **21**, "CID\_01", "CEO", **new** Date("2015/02/27"), reseacher, calculate);  Employee b = **new** Employee("Danzo Polima", "BOM9", **30**, "CID\_99", "Employee", **new** Date("2017/12/01"), marketing, tell);  a.marry(**new** Spouse("Leonie Kathleen", "BBA1", **20**, **new** Date("2017/03/24")));  a.addChild(**new** Child("Kodomo Kun", "OIU1", **1**, "Superman"));  b.marry(**new** Spouse("Dale Holmes", "QAZ9", **29**, **new** Date("2000/09/12")));  b.addChild(**new** Child("Onnanoko Chan", "NHY9", **9**, "Bear"));  b.addChild(**new** Child("Alexis Aisha", "BVZ9", **5**, "Rubik's Cube"));  System.out.println(a);  System.out.println(b);  }  } |

**Person.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48 | **package** com.company;  **import** **java.util.Collection**;  **public** **class** **Person** {  **private** String name;  **private** String SSN;  **private** **int** age;  Person() {  **this**.setName("Default");  **this**.setSSN("0000");  **this**.setAge(**1**);  }  Person(String name, String SSN, **int** age) {  **this**.setName(name);  **this**.setSSN(SSN);  **this**.setAge(age);  }  **public** String **getName**() {  **return** name;  }  **public** String **getSSN**() {  **return** SSN;  }  **public** **int** **getAge**() {  **return** age;  }  **public** **void** **setName**(String name) {  **this**.name = name;  }  **public** **void** **setSSN**(String SSN) {  **this**.SSN = SSN;  }  **public** **void** **setAge**(**int** age) {  **if** (age < **0**) {  age = **0**;  }  **this**.age = age;  }  } |

**Spouse.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | **package** com.company;  **import** **java.util.Date**;  **public** **class** **Spouse** **extends** Person {  **private** Date anniversaryDate;  Spouse() {  **super**("DEFAULT Spouse", "DEFAULT Spouse SSN", **20**);  }  Spouse(String name, String SSN, **int** age, Date anniversaryDate) {  **super**(name, SSN, age);  setAnniversaryDate(anniversaryDate);  }  **public** Date **getAnniversaryDate**() {  **return** anniversaryDate;  }  **public** **void** **setAnniversaryDate**(Date anniversaryDate) {  **this**.anniversaryDate = anniversaryDate;  }  } |

**Child.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | **package** com.company;  **public** **class** **Child** **extends** Person {  **private** String favoriteToy;  Child() {  **super**("DEFAULT Child", "DEFAULT Child SSN", **1**);  }  Child(String name, String SSN, **int** age, String favoriteToy) {  **super**(name, SSN, age);  setFavoriteToy(favoriteToy);  }  **public** String **getFavoriteToy**() {  **return** favoriteToy;  }  **public** **void** **setFavoriteToy**(String favoriteToy) {  **this**.favoriteToy = favoriteToy;  }  } |

**Division.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | **package** com.company;  **public** **class** **Division** {  **private** String divisionName;  Division() {  setDivisionName("DEFAULT DIVISION!");  }  Division(String divisionName) {  setDivisionName(divisionName);  }  **public** String **getDivisionName**() {  **return** divisionName;  }  **public** **void** **setDivisionName**(String divisionName) {  **this**.divisionName = divisionName;  }  } |

**JobDescription.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | **package** com.company;  **public** **class** **JobDescription** {  **private** String description;  JobDescription() {  setDescription("DEFAULT JOB DESCRIPTION!");  }  JobDescription(String description) {  setDescription(description);  }  **public** String **getDescription**() {  **return** description;  }  **public** **void** **setDescription**(String description) {  **this**.description = description;  }  } |

**Employee.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156 | **package** com.company;  **import** **java.util.ArrayList**;  **import** **java.util.Collection**;  **import** **java.util.Date**;  **public** **class** **Employee** **extends** Person {  **private** String companyID;  **private** String title;  **private** Date startDate;  **public** Spouse spource;  **public** ArrayList<Child> children;  **public** Division division;  **public** ArrayList<JobDescription> jobDescriptions;  Employee() {  **super**();  **this**.setCompanyID("0");  **this**.setTitle("Employee");  **this**.setStartDate(**new** Date());  // Заавал нэг салбартай байна  **this**.addDivision(**new** Division());  // Заавал нэг болон түүнээс дээш ажилын тайлбартай байна  **this**.addJobDescription(**new** JobDescription());  }  Employee(String name, String SSN, **int** age, String companyID, String title, Date startDate, Division division, JobDescription jobDescription) {  **super**(name, SSN, age);  **this**.setCompanyID(companyID);  **this**.setTitle(title);  **this**.setStartDate(startDate);  // Заавал нэг салбартай байна  **this**.addDivision(division);  // Заавал нэг болон түүнээс дээш ажилын тайлбартай байна  **this**.addJobDescription(jobDescription);  }  **void** **marry**(Spouse bride) {  **if** (**this**.spource != **null**) {  System.out.println("Already Married!");  **return**;  }  **this**.spource = bride;  }  **void** **divorce**() {  **if** (**this**.spource == **null**) {  System.out.println("You are not married!");  **return**;  }  **this**.spource = **null**;  }  **int** **countSpource**() {  **if** (**this**.spource == **null**) {  **return** **0**;  }  **return** **1**;  }  **void** **addChild**(Child child) {  **if** (**this**.children == **null**) {  **this**.children = **new** ArrayList<Child>();  }  **this**.children.add(child);  }  **void** **removeChild**(String name) {  **this**.children.removeIf((Child child) -> child.getName() == name);  }  **int** **countChild**() {  **return** **this**.children.size();  }  **void** **addDivision**(Division division) {  **if** (**this**.division != **null**) {  System.out.println("Already Has Division!");  **return**;  }  **this**.division = division;  }  **void** **swapDivision**(Division division) {  **this**.division = division;  }  **int** **countDivision**() {  **return** **1**;  }  **void** **addJobDescription**(JobDescription jobDescription) {  **if** (**this**.jobDescriptions == **null**) {  **this**.jobDescriptions = **new** ArrayList<JobDescription>();  }  **this**.jobDescriptions.add(jobDescription);  }  **void** **removeJobDescription**(String description) {  **if** (jobDescriptions.size() == **1**) {  System.out.println("CANNOT REMOVE DESCRIPTION. ONLY ONE LEFT!");  }  **this**.jobDescriptions  .removeIf((JobDescription JobDescription) -> JobDescription.getDescription() == description);  }  **int** **countJobDescription**() {  **return** **this**.jobDescriptions.size();  }  **public** String **getCompanyID**() {  **return** companyID;  }  **public** String **getTitle**() {  **return** title;  }  **public** Date **getStartDate**() {  **return** startDate;  }  **public** **void** **setCompanyID**(String companyID) {  **this**.companyID = companyID;  }  **public** **void** **setTitle**(String title) {  **this**.title = title;  }  **public** **void** **setStartDate**(Date startDate) {  **this**.startDate = startDate;  }  **public** String **toString**() {  String name = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";  name += "Name: " + **this**.getName() + "\n";  name += "SSN: " + **this**.getSSN() + "\n";  name += "Age: " + **this**.getAge() + "\n";  name += "\n" + "Spouse" + "\n";  name += "Name: " + **this**.spource.getName() + "\n";  name += "\n" + "Children" + "\n";  **for** (Child temp : **this**.children) {  name += "Name: " + temp.getName() + "\n";  }  name += "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";  **return** name;  }  } |