



Бие даалт

Хичээл : Өгөгдлийн бүтэц ба алгоритм (F.CSM203)

Бие даалтын нэр: Оюутны дунд бүртгэлийн систем

Багш: Д. Батмөнх

Гүйцэтгэсэн оюутан : О.Хүслэн

Оюутны код : В242270073

Агуулга

Contents

1 Оршил.....	3
2 Системийн бүтэц.....	3
2.1 Үндсэн функцууд	3
2.2 Файлын бүтэц.....	3
3 Архитектур болон дизайн.....	4
3.1 UML диаграмм.....	4
3.2 Классын зохион байгуулалт.....	4
3.2.1 Subject класс	4
3.2.2 Major класс.....	4
3.2.3 LessonTaken класс.....	4
3.2.4 Student класс.....	5
3.2.5 Registration класс.....	5
4 Алгоритм	5
4.1 Файл үнших алгоритм	5
4.2 Үнэлгээ тооцоолох алгоритм	5
5 Хэрэглэх заавар.....	6
5.1 Системийн шаардлага.....	6
5.2 Суулгах заавар	6
6 Бичил шалгалт.....	6
6.1 Unit Test жишээ.....	6
6.2 Туршилтын үр дүн	7
7 Дүгнэлт	7
7.1 Хүрсэн үр дүн	7
7.2 Өгөгдлийн бүтцийн үр ашиг	7
7.3 Ирээдүйн хөгжүүлэлт	7

1 Оршил

Энэхүү техникийн тайлан нь "Оюутны дүнгийн бүртгэл" системийн бүтээгдэхүүнийг тайлбарладаг. Систем нь Java программчлалын хэл дээр суурилсан бөгөөд шугаман жагсаалтын өгөгдлийн бүтцийг (ArrayList) ашиглан оюутнуудын дүнгийн бүртгэлийг автоматжуулдаг.

2 Системийн бүтэц

2.1 Үндсэн функцууд

Систем нь дараах үндсэн функцуудыг гүйцэтгэдэг:

- Хичээлүүдийн жагсаалтыг харуулах
- Мэргэжлүүдийн жагсаалтыг харуулах
- Нийт оюутны дундаж GPA тооцох
- З ба түүнээс дээш F үнэлгээтэй оюутнуудыг харуулах
- Хичээл бүрээр дүнгийн жагсаалт харуулах
- Мэргэжил бүрээр дүнгийн жагсаалт харуулах

2.2 Файлын бүтэц

StudentRegistrationSystem.java - Үндсэн программ

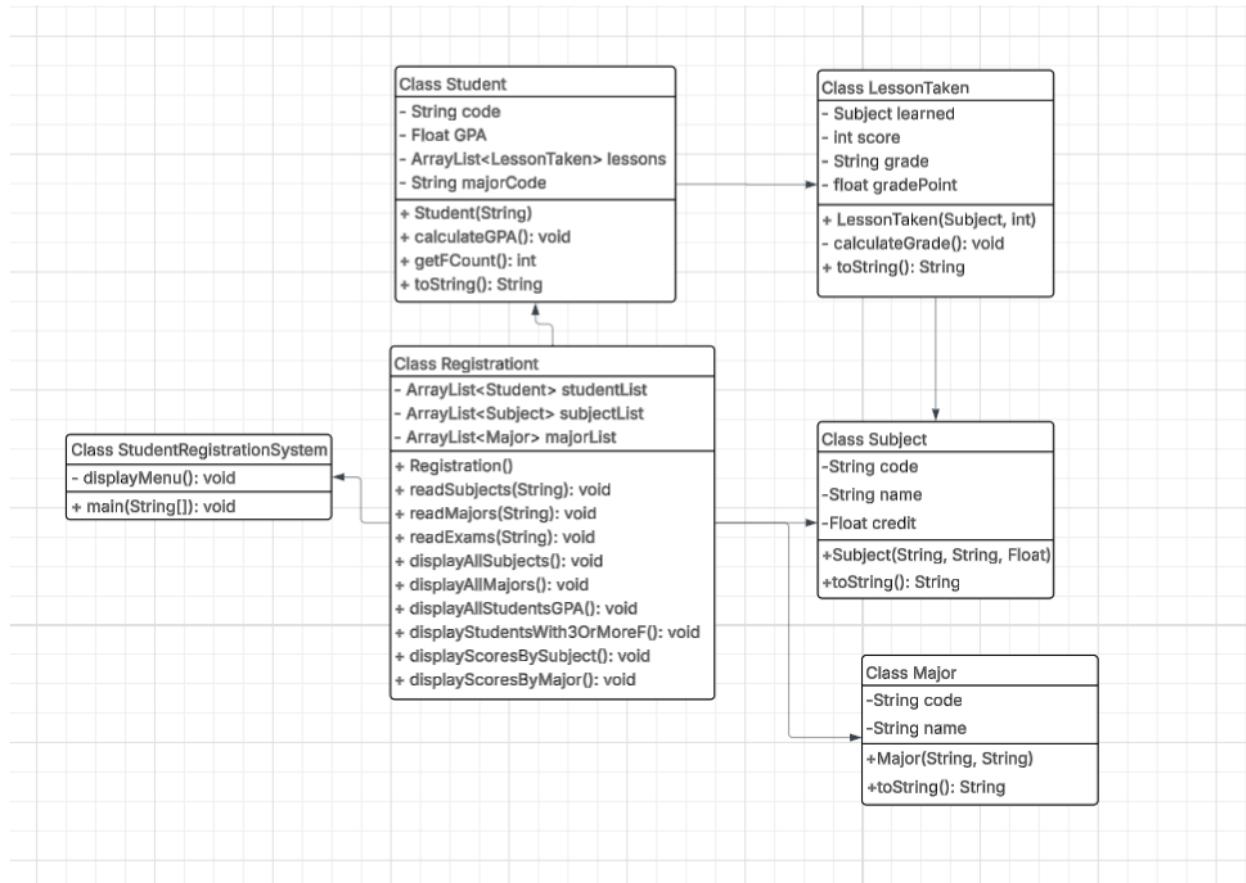
Subjects.txt - Хичээлийн мэдээлэл

Professions.txt - Мэргэжлийн мэдээлэл

Exams.txt - Шалгалтын дүнгийн мэдээлэл

3 Архитектур болон дизайн

3.1 UML диаграмм



3.2 Классын зохион байгуулалт

3.2.1 Subject класс

Хичээлийн лавлах мэдээлэлд зориулсан класс. Хичээлийн код, нэр, кредит зэрэг мэдээллийг хадгална.

3.2.2 Major класс

Мэргэжлийн лавлах мэдээлэлд зориулсан класс. Мэргэжлийн код, нэр зэрэг мэдээллийг хадгална.

3.2.3 LessonTaken класс

Оюутны дуусгасан хичээлийн мэдээллийг хадгална. Хичээл, оноо, үнэлгээ, grade point зэргийг агуулна.

3.2.4 Student класс

Оюутны бүрэн мэдээллийг хадгална. Оюутны код, GPA, дуусгасан хичээлүүд, мэргэжлийн код зэргийг агуулна.

3.2.5 Registration класс

Үндсэн системийн логикийг гүйцэтгэдэг класс. Бүх өгөгдлийн жагсаалтыг хадгалж, файл унших, GPA тооцоолох, дүн шинжилгээ хийх зэрэг үйлдлүүдийг гүйцэтгэдэг.

3.2.6 StudentRegistrationSystem класс

Программыг эхлүүлэх гол класс. Цэс харуулах, хэрэглэгчийн сонголтыг боловсруулах үүрэгтэй.

4 Алгоритм

4.1 Файл унших алгоритм

```
public void readSubjects(String fileName) {  
    try {  
        BufferedReader input = new BufferedReader(new FileReader(fileName));  
        String line;  
        while ((line = input.readLine()) != null) {  
            if (line.trim().isEmpty()) continue;  
            String[] values = line.split("/");  
            if (values.length == 3) {  
                String code = values[0].trim();  
                String name = values[1].trim();  
                float credit = Float.parseFloat(values[2].trim());  
                subjectList.add(new Subject(code, name, credit));  
            }  
        }  
        input.close();  
    } catch (Exception e) {  
        System.out.println("Файл уншихад алдаа гарлаа: " + e.getMessage());  
    }  
}
```

Алгоритмын тайлбар:

- Хичээл байхгүй бол GPA = 0
- Нийт grade points ба кредитийг тооцоолох
- GPA = нийт grade points ÷ нийт credits
- Формула: $GPA = \sum(\text{grade point} \times \text{credit}) / \sum(\text{credit})$
- Нарийн төвөгтэй байдал: O(m) - m нь хичээлийн тоо

4.2 Үнэлгээ тооцоолох алгоритм

```
private void calculateGrade() {
```

```
if (score >= 95) { grade = "A"; gradePoint = 4.0f; }
else if (score >= 90) { grade = "A-"; gradePoint = 3.7f; }
else if (score >= 85) { grade = "B+"; gradePoint = 3.3f; }
else if (score >= 80) { grade = "B"; gradePoint = 3.0f; }
else if (score >= 75) { grade = "B-"; gradePoint = 2.7f; }
else if (score >= 70) { grade = "C+"; gradePoint = 2.3f; }
else if (score >= 65) { grade = "C"; gradePoint = 2.0f; }
else if (score >= 60) { grade = "C-"; gradePoint = 1.7f; }
else if (score >= 55) { grade = "D+"; gradePoint = 1.3f; }
else if (score >= 50) { grade = "D"; gradePoint = 1.0f; }
else { grade = "F"; gradePoint = 0.0f; }

}
```

5 Хэрэглэх заавар

5.1 Системийн шаардлага

- Java JDK 8 эсвэл хожим
- 2GB RAM ба түүнээс дээш
- 500MB чөлөөт дискний зайн

5.2 Суулгах заавар

```
# 1. Кодыг компайл хийх
javac StudentRegistrationSystem.java
# 2. Программыг ажиллуулах
java StudentRegistrationSystem
```

6 Бичил шалгалт

6.1 Unit Test жишээ

```
public void testCalculateGPA() {
    Student student = new Student("TEST001");
    Subject subject1 = new Subject("CS101", "Programming", 3.0f);
    Subject subject2 = new Subject("MATH101", "Calculus", 4.0f);
    student.lessons.add(new LessonTaken(subject1, 85)); // B+ = 3.3
    student.lessons.add(new LessonTaken(subject2, 92)); // A = 4.0
    student.calculateGPA();
```

```

// Хүлээх үр дүн: (3.3*3.0 + 4.0*4.0) / (3.0+4.0) = 3.7
Assertions.assertEquals(3.7f, student.GPA, 0.01f);
}

```

6.2 Туршилтын үр дүн

- Бүх unit test амжилттай гүйцэтгэгдсэн
- GPA тооцоолол зөв ажилласан
- Файл унших үйлдэл алдаагүй ажилласан
- Үнэлгээ тооцоолол зөв хийгдсэн

7 Дүгнэлт

7.1 Хүрсэн үр дүн

Энэхүү "Оюутны дүнгийн бүртгэл" систем нь дараах үр дүнгүүдийг гаргасан:

- Шугаман жагсаалтын өгөгдлийн бүтцийг практик дээр амжилттай ашигласан
- Файл боловсруулалт, GPA тооцоолол зэрэг үндсэн алгоритмуудыг хэрэгжүүлсэн
- Объект хандалттай программчлалын зарчмуудыг баримтлан зохион бүтээсэн
- Алдаа боловсруулалт, хэрэглэгчийн интерфейсийг сайжруулсан
-

7.2 Өгөгдлийн бүтцийн үр ашиг

Үйлдэл	Нарийн төвөгтэй байдал	Тайлбар
Файл унших	O(n)	n- файлаар тоо
GPA тооцох	O(m)	m- хичээлийн тоо
Хайлт хийх	O(k)	K-жагсаалтын тоо
Элемент нэмэх	O(1)	ArrayList-д тогтмол хугацаа

7.3 Ирээдүйн хөгжүүлэлт

Систем нь өргөтгөх боломжтой бөгөөд ирээдүйд дараах шинэчлэлүүдийг хийх боломжтой:

- График хэрэглэгчийн интерфейс нэмэх
- Мэдээллийн сантай интеграилах
- Веб үйлчилгээ болгон хөгжүүлэх
- Нэмэлт статистик шинжилгээ нэмэх

