

Цалин бодох ERP системийн модуль

Багийн гишүүд:

- Мягмарсүрэн
- Эрдэнэтунгалаг
- Билгүүн

Төслийн зорилго

- Манай төслийн гол зорилго бол байгууллагын ажилчдын цалинг шударгаар, ил тод, автоматжуулсан байдлаар тооцоолох системийг Java хэл дээр бүтээх явдал юм.
- Томьёо:
- $\text{Цэвэр цалин} = \text{Үндсэн} + \text{Илүү цаг} + \text{Бонус} - \text{Тасалт} - \text{Татвар} - \text{Шимтгэл}$

Классуудын бүтэц

1. Employee

- Зорилго: Ажилчдын мэдээллийг хадгална
- Үндсэн өгөгдөл / Үйлдэл: id, name, salary, contract

2. Contract

- Зорилго: Ажилтны гэрээний дэлгэрэнгүй тайлбарыг хадгална
- Үндсэн өгөгдөл / Үйлдэл: details

3. Attendance

- Зорилго: Ирцийн мэдээлэл бүртгэнэ, ажилчдыг ирцэд бүртгэнэ
- Үндсэн өгөгдөл / Үйлдэл: daysPresent, presentEmployees, markPresent(), isPresent()

4. Bonus

- Зорилго: Ажилтны бонус (нэмэгдэл) мэдээлэл
- Үндсэн өгөгдөл / Үйлдэл: amount

5. Deduction

- Зорилго: Ажилтны суутгалын мэдээлэл
- Үндсэн өгөгдөл / Үйлдэл: amount

6. Salary

- Зорилго: Тооцоологдсон цалингийн үр дүн
- Үндсэн өгөгдөл / Үйлдэл: employee, netSalary

7. PayrollSystem

- Зорилго: Цалинг тооцоолох гол модулиудыг удирдана
- Үндсэн өгөгдөл / Үйлдэл: generateSalary(), getSalaries(), нэмэгдэл, суутгал, ирц зэргийг харгалзан тооцоолно

8. PayrollSystemTest

- Зорилго: JUnit тест хариуцна
- Үндсэн өгөгдөл / Үйлдэл: testCalculateSalaryAndAttendance()

9. log4j2.xml

- Зорилго: Лог бичих тохиргоо
- Үндсэн өгөгдөл / Үйлдэл: Console болон файл руу бичих лог формат тодорхойлсон

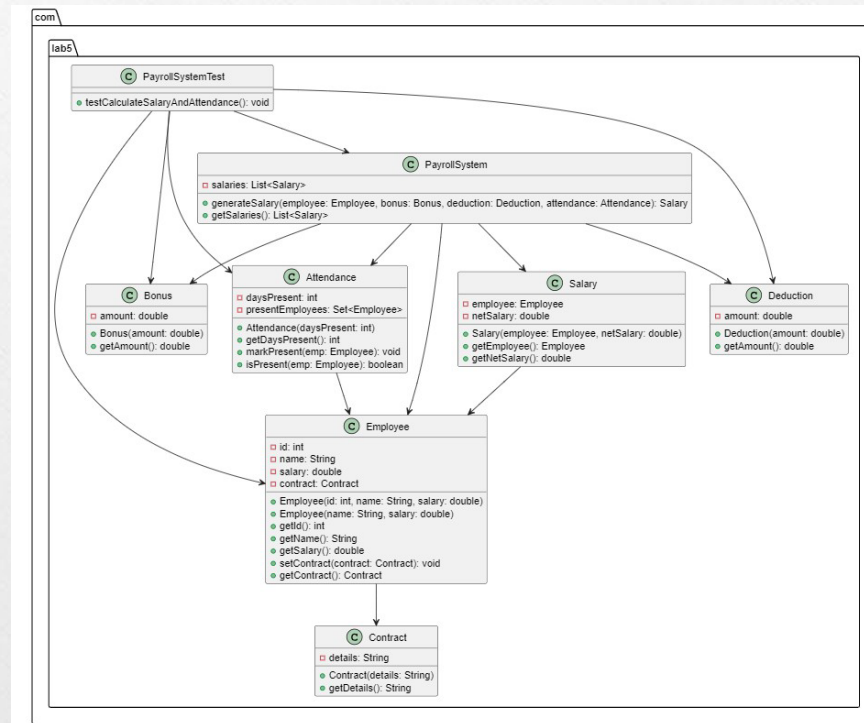
10. pom.xml

- Зорилго: Төслийн хамаарлууд болон билд тохиргоог удирдана
- Үндсэн өгөгдөл / Үйлдэл: Java 17, Maven, JUnit 5, Log4j 2

Цалин бодох логик

- • Үндсэн цалин
- • Илүү цаг (1.5х)
- • Бонус
- • Тасалт, хоцролт (торгууль)
- • Нийгмийн даатгал – 10%
- • Эрүүл мэндийн даатгал – 5%
- • Компанийн шимтгэл

UML диаграмм



Тест ба демо

- • JUnit тест нэмсэн
- • Алдааг засаж, лог файл нэмсэн
- • Үр дүн: тогтвортой, найдвартай ажиллаж байна.

- Системийн онцлог болон давуу талууд
- **Объект хандалт загвар (ООР):** Encapsulation, Inheritance, Polymorphism, Abstraction зарчмуудыг бүрэн ашигласан.
- **Алдаа зохицуулалт:** Хэрэглэгчийн буруу оролтод IllegalArgumentException ашиглан хариу өгдөг.
- **Хэрэглэгчийн оролт шалгалт:** Нэр хоосон, цалин сөрөг, бонус хэт өндөр зэргээс сэргийлсэн.
- **Тестүүд:** JUnit ашиглан зөвхөн зөв оролт төдийгүй, алдаатай (edge case) нөхцлүүдийг шалгасан.
- **Лог бичилт:** Log4j ашиглан системийн үйлдлийг log файл болон console руу хөтөлдөг.
- **UML диаграм:** Бүтэц, харилцааг ойлгомжтойгоор дүрсэлсэн.
- Бодитоор хэрэгжүүлсэн зүйлс
 - Цалинг бонус, суутгал, ирц дээр үндэслэн бодох логик
 - Хувь хүн бүрийн мэдээллийг хадгалах ангилсан бүтэц
 - Юнит тестүүдээр баталгаажсан тооцоолол
 - Төслийн бүх мэдээлэл файлаар хадгалагддаг
 - Алдаатай өгөгдлийг автоматаар илрүүлж, системийг хамгаалдаг

Манай төслийн дүгнэлт

Энэхүү зорилгыг амжилттай хэрэгжүүлж, объект хандалтат загварыг баримталсан, модулиудыг салгаж зохион байгуулсан, логикын нарийвчилсан бодлого шийдлүүдийг оруулсан болно.