МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Южный Федеральный Университет»

Институт высоких технологий и пьезотехники



**Кафедра информационных и**

**измерительных технологий**

**Направление: 09.03.03 "Прикладная информатика"**

**Пояснительная записка к курсовой работе**

**на тему: «Бизнес-логика системы управления предприятием (метрология и финансы)»**

Выполнил студент 4 курса 6 группы:

Тезин Александр Михайлович

Ростов-на-Дону

2020

**Оглавление**

[1.](#_Toc61423332) Техническое задание 3

[2. UML моделирование](#_Toc61423336) 4

[2.1 Диаграмма вариантов использования](#_Toc61423337) 4

[2.2 Диаграмма классов 1](#_Toc61423338)7

[2.3 Диаграмма деятельности 1](#_Toc61423339)8

[2.4 Диаграмма компонентов](#_Toc61423340) 21

[3. Тестирование](#_Toc61423341) 22

**Техническое задание**

В качестве курсового проекта была выбрана разработка бизнес-логики системы управления предприятием (метрология и финансы). Проект представляет собой CLI (Command Line Interface) приложение на языке программирования Java, предоставляющее API (application programming interface). Данное приложение создано для начальников отделов метрологических предприятий.

Функциональные возможности приложения:

- Создать наряд

- Редактировать сотрудников в наряде

- Удалить наряд

- Посмотреть все наряды, где есть сотрудник с таким id

- Посмотреть все наряды, где есть исполнитель с таким id

- Добавить сотрудника

- Повысить в должности сотрудника

- Понизить в должности сотрудника

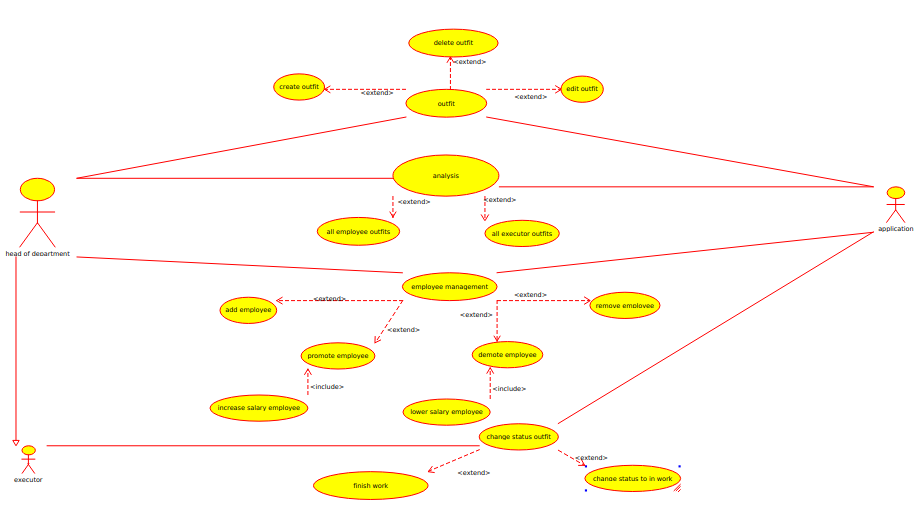
- Уволить рабочего

- Установить статус работы наряда как “в работе”

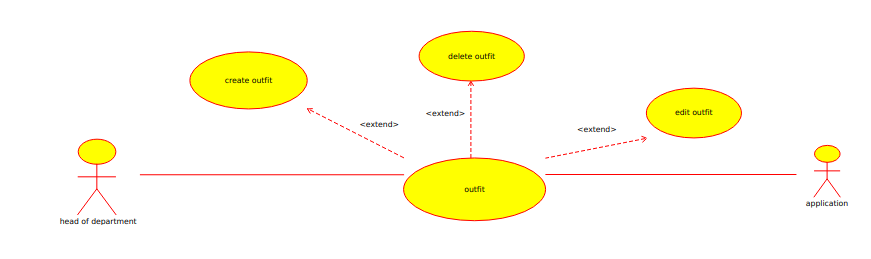
- Установить статус работы наряда как “завершен”

**UML моделирование**

**Общая диаграмма вариантов использования**

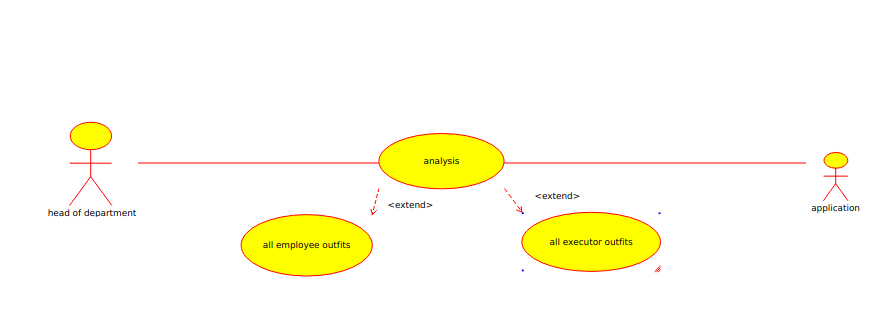


**Вариант использования: Outfit**



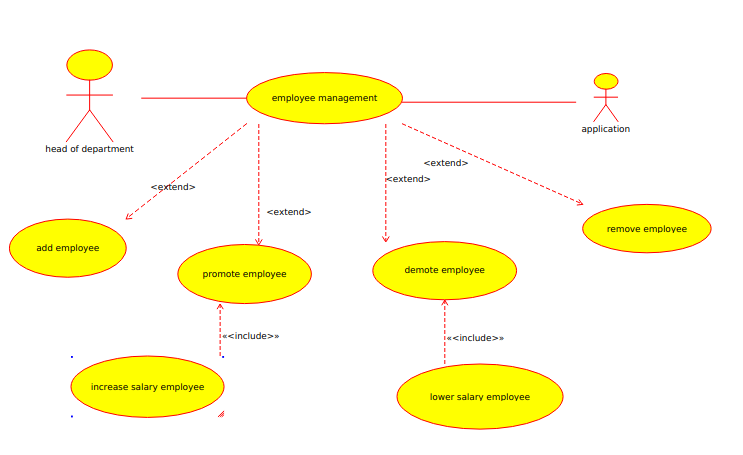
|  |  |
| --- | --- |
| **Вариант использования** | **Описание** |
| 1. outfit | **Вызов расширяющих методов:**  Расширяющий метод create outfit вызывается, когда в методе outfit параметр operation равно “create\_outfit”.  Расширяющий метод delete outfit вызывается, когда в методе outfit параметр operation равно “delete\_outfit”.  Расширяющий метод edit outfit вызывается, когда в методе outfit параметр operation равно “edit\_outfit”.  **Входные параметры:**  operation - операция которую хочет пользователь произвести (создать, удалить, редактировать наряд)  idOutfit - идентификатор наряда  idEmployees - список идентификаторов рабочих в наряде  dataStartWork - дата начала работы наряда  idCustomer - идентификатор покупателя  idHeadOfDepartment - идентификатор главы отдела  idPlaceWork - идентификатор места работы нарядами  typeWork - тип работы  typeInstrument - тип инструмента  idTypeMeasurementInstrument - идентификатор инструмента  statusWork - в каком статусе находится наряд (создан,в работе,завершен)  **Описание:**  Управление нарядами, возможность создавать, редактировать и удалять наряды.  **Возвращаемое значение:**  Если метод выполнился успешно:  Возвращает Status.*SUCCESS*  Если метод выполнился неудачно:  Возвращает Status.*FAIL* |
| 1.1 create outfit | **Вызов метода:**  Вызываем метод если в методе outfit значение operation равно create\_outfite  **Входные параметры:**  idOutfit - идентификатор наряда  idEmployees - список идентификаторов рабочих в наряде  dataStartWork - дата начала работы наряда  idCustomer - идентификатор покупателя  idHeadOfDepartment - идентификатор главы отдела  idPlaceWork - идентификатор места работы нарядами  typeWork - тип работы  typeInstrument - тип инструмента  idTypeMeasurementInstrument - идентификатор инструмента  statusWork - в каком статусе находится наряд(создан) **Описание:** Создается новый наряд на поверку/калибровку.  **Возвращаемое значение:**  Если метод выполнился успешно:  Возвращает список из двух параметров 1 параметр true, 2 – созданный наряд  Если метод выполнился неудачно:  Возвращает список из двух параметров 1 параметр false, 2 - null |
| 1.2 delete outfit | **Вызов метода:**  Вызываем метод если в методе outfit значение operation равно delete\_outfit.  **Входные параметры:**  idOutfit - идентификатор наряда  idHeadOfDepartment - идентификатор главы отдела **Описание:**  Удаляет наряд **Возвращаемое значение:**  Если метод выполнился успешно:  Возвращает True  Если метод выполнился неудачно:  Возвращает False |
| 1.3 edit outfit | **Вызов метода:**  Вызываем метод если в методе outfit значение operation равно edit\_outfit.  **Входные параметры:**  idOutfit - идентификатор наряда  idHeadOfDepartment - идентификатор главы отдела  idEmployees - список идентификаторов рабочих в наряде  **Описание:**  Редактирует идентификаторы рабочих которые в наряде  **Возвращаемое значение:**  Если метод выполнился успешно:  Возвращает список из двух параметров 1 параметр true, 2 – отредактированный наряд  Если метод выполнился неудачно:  Возвращает список из двух параметров 1 параметр false, 2 - null |

**Вариант использования: analysis**



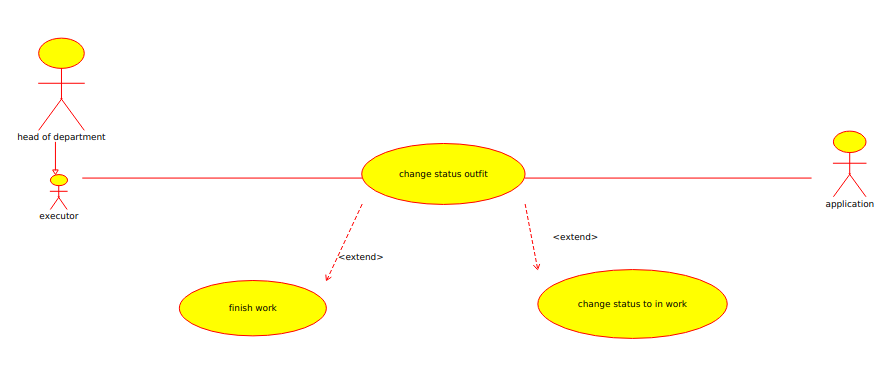
|  |  |
| --- | --- |
| **Вариант использования** | **Описание** |
| 1. analysis | **Вызов расширяющих методов:**  Расширяющий метод all employee outfits вызывается, когда в методе analysis параметр operation равно “all\_employee\_outfits”.  Расширяющий метод all\_executor\_outfits вызывается, когда в методе analysis параметр operation равно “all\_executor\_outfits”.  **Входные параметры:**  operation - операция которую хочет пользователь произвести (посмотреть все наряды где есть такой индентификатор сотрудника, посмотреть все наряды где есть такой индентификатор исполнителя)  idAnalysis - идентификатор сотрудника/ исполнителя  idHeadOfDepartment - идентификатор главы отдела  **Описание:**  Анализ информации о сотруднике/исполнителе  **Возвращаемое значение:**  Если метод выполнился успешно:  Возвращает Status.*SUCCESS*  Если метод выполнился неудачно:  Возвращает Status.*FAIL* |
| * 1. all employee outfits | **Вызов метода:**  Вызываем метод если в методе analysis значение operation равно all\_employee\_outfits.  **Входные параметры:**  idAnalysis - табельный номер сотрудника  idHeadOfDepartment - идентификатор главы отдела  **Описание:**  Просмотр нарядов, где есть такой идентификатор сотрудника  **Возвращаемое значение:**  Если метод выполнился успешно:  Возвращает список из двух параметров 1 параметр true, 2 – список нарядов где был сотрудник  Если метод выполнился неудачно:  Возвращает список из двух параметров 1 параметр false, 2 – null |
| 2.2 all\_executor\_outfits | **Вызов метода:**  Вызываем метод если в методе analysis значение operation равно all\_executor\_outfits.  **Входные параметры:**  idAnalysis - идентификатор исполнителя  idHeadOfDepartment - идентификатор главы отдела  **Описание:**  Просмотр нарядов, где есть такой идентификатор исполнителя  **Возвращаемое значение:**  Если метод выполнился успешно:  Возвращает список из двух параметров 1 параметр true, 2 – список нарядов где был исполнитель с данным id  Если метод выполнился неудачно:  Возвращает список из двух параметров 1 параметр false, 2 – null |

**Вариант использования: employee management**



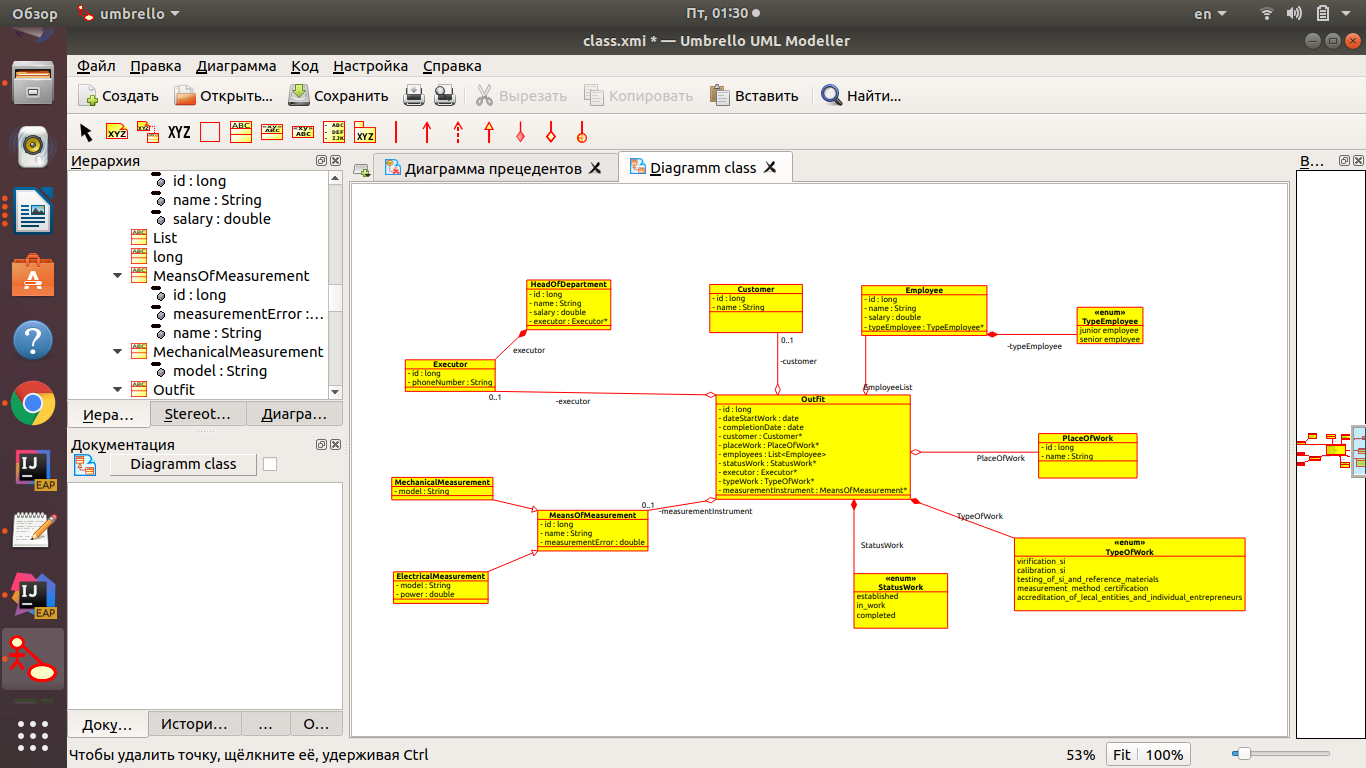
|  |  |
| --- | --- |
| **Вариант использования** | **Описание** |
| 3. employee management | **Вызов расширяющих методов:**  Расширяющий метод add employee вызывается, когда в методе employee management параметр operation равно “add\_employee”.  Расширяющий метод promote employee вызывается, когда в методе employee management параметр operation равно “promote\_employee”.  Расширяющий метод demote employee вызывается, когда в методе employee management параметр operation равно “demote\_employee”.  Расширяющий метод remove employee вызывается, когда в методе employee management параметр operation равно “remove\_employee”.  **Входные параметры:**  operation - действие которое производится(добавить сотрудника, удалить сотрудника, повысить или понизить сотрудника в должности)  idEmployee - табельный номер сотрудника  name - имя сотрудника  idHeadOfDepartment - идентификатор главы отдела  **Описание:**  Управление сотрудниками (возможность добавления сотрудника, удаления сотрудника(увольнения), повышения сотрудника, понижения сотрудника в должности)  **Возвращаемое значение:**  Если метод выполнился успешно:  Возвращает Status.*SUCCESS*  Если метод выполнился неудачно:  Возвращает Status.*FAIL* |
| 3.1. add employee | **Вызов метода:**  Вызываем метод если в методе employee management значение operation равно add\_employee.  **Входные параметры:**  idEmployee - табельный номер сотрудника  name - имя сотрудника  idHeadOfDepartment - идентификатор главы отдела  **Описание:**  Добавление нового сотрудника в штат  **Возвращаемое значение:**  Если метод выполнился успешно:  Возвращает список из двух параметров 1 параметр true, 2 – созданный сотрудник  Если метод выполнился неудачно:  Возвращает список из двух параметров 1 параметр false, 2 – null |
| 3.2 promote employee | **Вызов метода:**  Вызываем метод если в методе employee management значение operation равно promote\_employee.  **Входные параметры:**  idEmployee - табельный номер сотрудника  idHeadOfDepartment - идентификатор главы отдела  **Описание:**  Повысить сотрудника в должности  **Возвращаемое значение:**  Если метод выполнился успешно:  Возвращает список из двух параметров 1 параметр true, 2 – сотрудник  Если метод выполнился неудачно:  Возвращает список из двух параметров 1 параметр false, 2 – null |
| 3.2.1 increase salary employee | **Входные параметры:**  employee – сотрудник  **Описание:**  Повысить зарплату сотруднику  **Возвращаемое значение:**  employee – сотрудник |
| 3.3 demote employee | **Вызов метода:**  Вызываем метод если в методе employee management значение operation равно demote\_employee.  **Входные параметры:**  idEmployee - табельный номер сотрудника  idHeadOfDepartment - идентификатор главы отдела  **Описание:**  Понизить сотрудника в должности  **Возвращаемое значение:**  Если метод выполнился успешно:  Возвращает список из двух параметров 1 параметр true, 2 – сотрудник  Если метод выполнился неудачно:  Возвращает список из двух параметров 1 параметр false, 2 – null |
| 3.3.1 lower salary employee | **Входные параметры:**  employee – сотрудник  **Описание:**  Понизить зарплату сотруднику  **Возвращаемое значение:**  employee – сотрудник |
| 3.4 remove employee | **Вызов метода:**  Вызываем метод если в методе employee management значение operation равно remove\_employee.  **Входные параметры:**  idEmployee - табельный номер сотрудника  idHeadOfDepartment - идентификатор главы отдела  **Описание:**  Уволить сотрудника  **Возвращаемое значение:**  Если метод выполнился успешно:  Возвращает True  Если метод выполнился неудачно:  Возвращает False |

**Вариант использования: change status outfit**



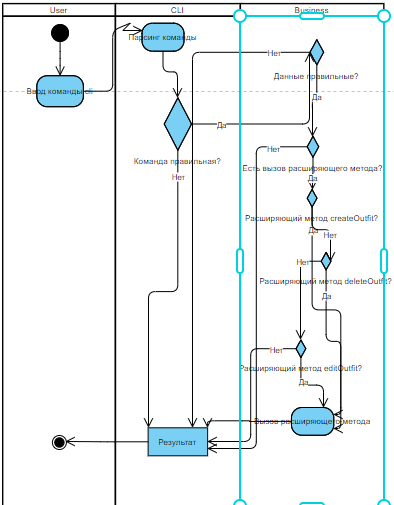
|  |  |
| --- | --- |
| **Вариант использования** | **Описание** |
| change status outfit | **Вызов расширяющих методов:**  Расширяющий метод finish work вызывается, когда в методе change status outfit параметр operation равно “finish\_work”.  Расширяющий метод change status to in work  вызывается, когда в методе change status outfit параметр operation равно “change\_status\_to\_in\_work”.  **Входные параметры:**  operation – действие которое производится(закрыть наряд, наряд в работе)  idOutfit – идентификатор наряда  idExecutor – табельный номер исполнителя  **Описание:**  Изменение статуса наряда  **Возвращаемое значение:**  Если метод выполнился успешно:  Возвращает Status.*SUCCESS*  Если метод выполнился неудачно:  Возвращает Status.*FAIL* |
| 4.1. finish work | **Вызов метода:**  Вызываем метод если в методе change status outfit значение operation равно finish\_work.  **Входные параметры:**  idOutfit – идентификатор наряда  idExecutor – табельный номер исполнителя  **Описание:**  Закрыть наряд как выполненный  **Возвращаемое значение:**  Если метод выполнился успешно:  Возвращает список из двух параметров 1 параметр true, 2 – наряд с изменным статусом  Если метод выполнился неудачно:  Возвращает список из двух параметров 1 параметр false, 2 – null |
| change status to in work | **Вызов метода:**  Вызываем метод если в методе change status outfit значение operation равно change\_status\_to\_in\_work.  **Входные параметры:**  idOutfit – номер наряда  idExecutor – табельный номер исполнителя  **Описание:**  Изменить статус наряда на «в работе»  **Возвращаемое значение:**  Если метод выполнился успешно:  Возвращает список из двух параметров 1 параметр true, 2 – наряд с изменным статусом  Если метод выполнился неудачно:  Возвращает список из двух параметров 1 параметр false, 2 – null |

**Диаграмма классов**

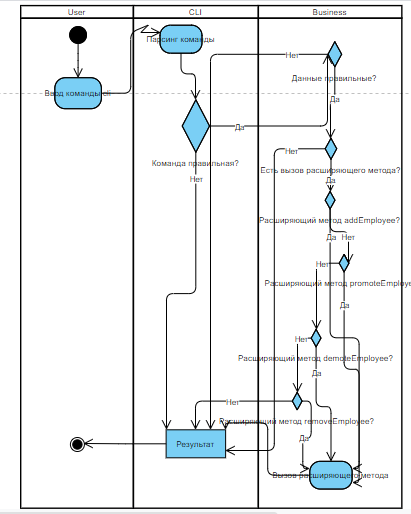


**Диаграмма деятельности**

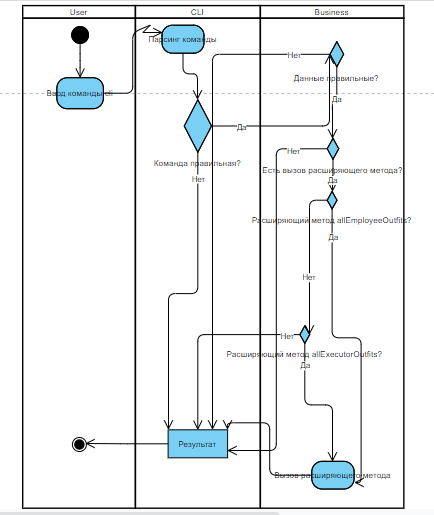
**Диаграмма варианта активности outfit**



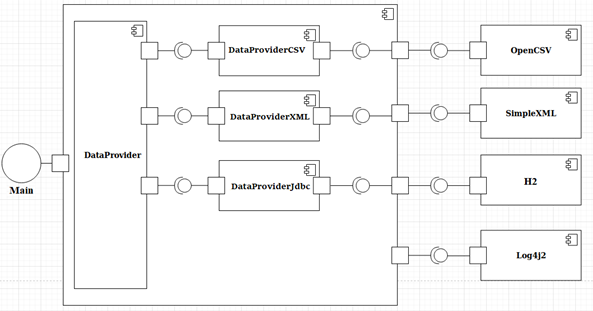
**Диаграмма варианта активности employee management**



**Диаграмма варианта активности analysis**



**Диаграмма компонентов**



**Тестирование**

Тестирование предполагало написание двух тестов для каждого метода API: позитивного и негативного.

Таблица – Сценарии тестирования и результаты тестов.

|  |  |  |  |
| --- | --- | --- | --- |
| **Тестовый метод** | **Тип сценария** | **Тестируемый метод** | **Что**  **контролируется?** |
| Наряд  outfitSuccess | Позитивный | provider.outfit  1)Проверка метода  при правильном названии  расширяющего метода  для вызова createOutfit  2) Проверка метода  при правильном названии  расширяющего метода  для вызова deleteOutfit  3) Проверка метода  при правильном названии  расширяющего метода  для вызова editOutfit | 1)Статус выполнения  метода:  assertEquals(Status.SUCCESS, provider.outfit("create\_outfit", 5, listEmployeeId, new Date(), 11,50,33,"CALIBRATION", "ELECTRICAL", 44, StatusWork.ESTABLISHED));  2)Статус выполнения  метода:  assertEquals(Status.SUCCESS, provider.outfit("delete\_outfit",1,  null, null, 0, 50, 0, null, null, 0,  null));  3)Статус выполнения  метода:  assertEquals(Status.SUCCESS, provider.outfit("edit\_outfit", 5, listEmployeeId, null, 0, 50, 0, null, null, 0, null)); |
| Наряд  outfitFail | Негативный | provider.outfit  1)Проверка метода  при неправильном  названии расширяющего  метода для вызова  createOutfit  2) Проверка метода при неправильном названии  расширяющего метода  для вызова deleteOutfit  3) Проверка метода при неправильном названии  расширяющего метода  для вызова editOutfit | 1)Статус выполнения  метода:  assertEquals(Status.FAIL, provider.outfit("create\_outfit", 5, listEmployeeId, new Date(), 21,50,33,"CALIBRATION", "ELECTRICAL", 70, StatusWork.ESTABLISHED));  2)Статус выполнения  метода:  assertEquals(Status.FAIL, provider.outfit("delete\_outfit",10,  null, null, 0, 50, 0, null, null, 0,  null));  3)Статус выполнения  метода  assertEquals(Status.FAIL, provider.outfit("edit\_outfit", 5, listEmployeeId, null, 0, 60, 0, null, null, 0, null)); |
| Создание наряда  createOutfitSuccess | Позитивный | provider.createOutfit  1)Проверка метода при правильно введенных  всех данных | 1)True значение:  assertTrue((boolean)provider.createOutfit  (6, new Date(), 13, 53,32,"CALIBRATION",  "ELECTRICAL",46, listEmployeeId, StatusWork.ESTABLISHED).get(0)); |
| Создание наряда  createOutfitSuccess | Негативный | provider.createOutfit  1)Проверка метода если  в наряде будет 1 сотруд-  -ник  2)Проверка метода если  будет несуществующий  id одного из сотрудников  3)Проверка метода если  будет введен  несуществующий id измерительного  инструмента  4)Проверка метода если  будет введен существу  ющий id наряда  5) Проверка метода если  будет введен несущест  вующий id покупателя  6) Проверка метода если  будет введен несущест  вующий id главы  отдела.  7) Проверка метода если  будет введен несущест  вующий id места работы  8) Проверка метода если  будет введен неправи  льно тип работ.  9) Проверка метода если  будет введен неправи  льно тип инструмента | 1)False значение:  assertFalse((boolean)provider.createOutfit  (6, new Date(), 13, 53,32,"CALIBRATION",  "ELECTRICAL", 46, listEmployeeId, StatusWork.ESTABLISHED).get(0));  2) False значение:  assertFalse((boolean) provider.createOutfit(6,  new Date(),13,53,32,"CALIBRATION",  "ELECTRICAL",46, listEmployeeId, StatusWork.ESTABLISHED).get());  3) False значение:  assertFalse((boolean)provider.createOutfit(50,  new Date(),13,53,32,"CALIBRATION",  "ELECTRICAL",50, listEmployeeId, StatusWork.ESTABLISHED).get(0));  4) False значение:  assertFalse((boolean)provider.createOutfit(2,  new Date(),13,53,32,"CALIBRATION",  "ELECTRICAL",46, listEmployeeId, StatusWork.ESTABLISHED).get(0));  5) False значение:  assertFalse((boolean)provider.createOutfit(8,  new Date(), 43,53,32,"CALIBRATION",  "ELECTRICAL",46, listEmployeeId, StatusWork.ESTABLISHED).get(0));  6) False значение:  assertFalse((boolean)provider.createOutfit(9,  new Date(), 13,63,32,"CALIBRATION",  "ELECTRICAL",46, listEmployeeId, StatusWork.ESTABLISHED).get(0));  7) False значение:  assertFalse((boolean)provider.createOutfit(10,  new Date(),13,53,92,"CALIBRATION",  "ELECTRICAL",46, listEmployeeId, StatusWork.ESTABLISHED).get(0));  8) False значение:  assertFalse((boolean)provider.createOutfit(11,  new Date(), 13,53,32,"CALIBRATIONS",  "ELECTRICAL",46, listEmployeeId, StatusWork.ESTABLISHED).get(0));  9) False значение:  assertFalse((boolean)provider.createOutfit(12,  new Date(), 13,53,32,"CALIBRATION",  "ELECTRICALS",46, listEmployeeId, StatusWork.ESTABLISHED).get(0)); |
| Удалить наряд  deleteOutfitSuccess | Позитивный | provider.deleteOutfit  1)Проверка метода при правильно введенных  всех данных | 1)True значение:  assertTrue(provider.deleteOutfit  (1, 50)); |
| Удалит наряд  deleteOutfitFail | Негативный | provider.deleteOutfit  1) Проверка метода если  будет введен несущест  вующий id наряда.  2) Проверка метода если  будет введен несущест  вующий id начальника  отдела | 1) False значение:  assertFalse(provider.deleteOutfit  (10, 50));  2) False значение:  assertFalse(provider.deleteOutfit  (2, 80)); |
| Редактировать  наряд  editOutfitSuccess | Позитивный | provider.editOutfit  1) Проверка метода при правильно введенных  всех данных | 1) True значение:  assertTrue((boolean)provider.editOutfit(2, listEmployeeId, 53).get(0)); |
| Редактировать  наряд  editOutfitFail | Негативный | provider.editOutfit  1)Проверка метода если  редактактируем сотруд  ников наряда на других  сотрудников которых  несуществует(нет id)  2) Проверка метода если  редактактируем сотруд  ников наряда на другого  сотрудника которого  несуществует(нет id)  3) Проверка метода если  будет введен несущест  вующий id наряда  4) Проверка метода если  будет введен несущест  вующий id начальника  отдела | 1) False значение:  assertFalse((boolean)provider.editOutfit(2, listEmployeeId, 53).get(0));  2) False значение:  assertFalse((boolean)provider.editOutfit(2, listEmployeeId, 53).get(0));  3) False значение:  assertFalse((boolean)provider.editOutfit(31, listEmployeeId, 53).get(0));  4) False значение:  assertFalse((boolean)provider.editOutfit(2, listEmployeeId, 98).get(0)); |
| Анализ информации  о сотруднике  /исполнителе  analysisSuccess | Позитивный | provider.analysis  1)Проверка метода  при правильном названии  расширяющего метода  для вызова  allEmployeeOutfit  2)Проверка метода  при правильном названии  расширяющего метода  для вызова  allExecutorOutfits | 1)Статус выполнения  метода:  assertEquals(Status.  SUCCESS,provider.analysis  ("all\_employee\_outfits",  22, 52));  2)Статус выполнения  метода:  assertEquals(Status.  SUCCESS,provider.analysis  ("all\_executor\_outfits", 63, 52)); |
| Анализ информации  о сотруднике  /исполнителе  analysisFail | Негативный | provider.analysis  1) Проверка метода  при неправильном  названии расширяющего метода для вызова  2)Проверка метода  при неправильном  названии расширяющего метода для вызова  allExecutorOutfits | 1)Статус выполнения  метода:  assertEquals(Status.FAIL,  provider.analysis(  "all\_employee\_outfit", 22, 52));  2)Статус выполнения  метода:  assertEquals(Status.FAIL,  provider.analysis  ("all\_executor\_outfit", 63, 52)); |
| Просмотр нарядов, где есть такой идентификатор сотрудника  allEmployeeOutfits  Success | Позитивный | provider.allEmployee  Outfits  1) Проверка метода при правильно введенных  всех данных | 1) True значение:  assertTrue((boolean)provider.allEmployee  Outfits(22, 53).get(0)); |
| Просмотр нарядов, где есть такой идентификатор сотрудника  allEmployeeOutfits  Fail | Негативный | provider.allEmployee  Outfits  1) Проверка метода если  будет введен несущест  вующий id наряда  2) Проверка метода если  будет введен несущест  вующий id начальника  отдела | 1) False значение:  assertFalse((boolean)provider.  AllEmployeeOutfits(12, 53).get(0));  2) False значение:  assertFalse((boolean)provider.  AllEmployeeOutfits(22, 80).get(0)); |
| Просмотр нарядов, где есть такой идентификатор исполнителя  allExecutor  OutfitsSuccess | Позитивный | provider.allExecutor  Outfits  1) Проверка метода при правильно введенных  всех данных | 1) True значение:  assertTrue((boolean)provider.  AllExecutorOutfits(63, 53).get(0)); |
| Просмотр нарядов, где есть такой идентификатор исполнителя  allExecutor  OutfitsFail | Негативный | provider.allExecutor  Outfits  1) Проверка метода если  будет введен несущест  вующий id наряда  2) Проверка метода если  будет введен несущест  вующий id исполни  теля | 1) False значение:  assertFalse((boolean)provider.  AllExecutorOutfits(30, 53).get(0));  2) False значение:  assertFalse((boolean)provider.  AllExecutorOutfits(63, 63).get(0)); |
| Управление  сотрудниками  employeeManagement  Success | Позитивный | provider. employeeManagement  1) Проверка метода  при правильном названии  расширяющего метода  для вызова  addEmployee  2) Проверка метода  при правильном названии  расширяющего метода  для вызова  promoteEmployee  3) Проверка метода  при правильном названии  расширяющего метода  для вызова  demoteEmployee  4) Проверка метода  при правильном названии  расширяющего метода  для вызова  removeEmployee | 1)Статус выполнения  метода:  assertEquals(Status.  SUCCESS, provider.  employeeManagement  ("add\_employee",27, "Alen", 52));  2)Статус выполнения  метода:  assertEquals(Status.  SUCCESS,provider.  employeeManagement  ("promote\_employee",20, null, 52));  3)Статус выполнения  метода:  assertEquals(Status.  SUCCESS,provider.  employeeManagement  ("demote\_employee",21, null,52));  4)Статус выполнения  метода:  assertEquals(Status.  SUCCESS,provider.  employeeManagement  ("remove\_employee",27, null,52)); |
| Управление  сотрудниками  employeeManagement  Fail | Негативный | provider. employeeManagement  1) Проверка метода  при неправильном  названии расширяющего метода для вызова  addEmployee  2) Проверка метода  при неправильном  названии расширяющего метода для вызова  promoteEmployee  3) Проверка метода  при неправильном  названии расширяющего метода для вызова  demoteEmployee  4) Проверка метода  при неправильном  названии расширяющего метода для вызова  removeEmployee | 1)Статус выполнения  метода:  assertEquals(Status.  FAIL, provider.employee  Management("add\_employe",27,  "Alen", 52));  2)Статус выполнения  метода:  assertEquals(Status.  FAIL, provider.employee  Management("promote\_employe",  27, null, 52));  3)Статус выполнения  метода:  assertEquals(Status.  FAIL, provider.employee  Management("demote\_employee",  27, null, 52));  4)Статус выполнения  метода:  assertEquals(Status.  FAIL, provider.employee  Management("remove\_employee",  27, null, 52)); |
| Добавить  сотрудника  addEmployeeSuccess | Позитивный | provider.addEmployee  1) Проверка метода при правильно введенных  всех данных | 1) True значение:  assertTrue((boolean)provider.  addEmployee(28, "Kiko", 52).get(0)); |
| Добавить  сотрудника  addEmployeeFail | Негативный | provider.addEmployee  1) Проверка метода если  будет введено имя не  в правильном  формате  2) Проверка метода если  будет введен несущест  вующий id начальника  отдела | 1) False значение:  assertFalse((boolean)provider.  addEmployee  (28, "Kiko123", 52).get(0));  2) False значение:  assertFalse((boolean)provider.  addEmployee  (28, "Kiko", 78).get(0)); |
| Повысить сотру  дника в должности  promoteEmployee  Success | Позитивный | provider.  promoteEmployee  1) Проверка метода при правильно введенных  всех данных | 1) True значение:  assertTrue((boolean)provider.  PromoteEmployee(20, 53).get(0)); |
| Повысить сотру  дника в должности  promoteEmployee  Fail | Негативный | provider.  promoteEmployee  1)Проверка метода  при попытке повы  сить сотрудника  в должности кото  рый занимает  должность Senior  2) Проверка метода если  будет введен несущест  вующий id начальника  отдела | 1) False значение:  assertFalse((boolean)provider.  PromoteEmployee(21, 53).get(0));  2) False значение:  assertFalse((boolean)provider.  PromoteEmployee(20, 93).get(0)); |
| Повысить заработ  ную плату сотруд  нику  increaseSalary  EmployeeSuccess | Позитивный | 1)Проверка метода  при правильном  изменении  заработной  платы сотруднику | 1) Заработная плата:  assertEquals(employee1.  getSalary(),provider.increase  SalaryEmployee(employee2).  getSalary()); |
| Повысить заработ  ную плату сотруд  нику  increaseSalary  EmployeeFail | Негативный | 1)Проверка метода  при неправильном  изменении  заработной  платы сотруднику | 1) Заработная плата:  assertEquals(employee1.  getSalary(),provider.increase  SalaryEmployee(employee2).  getSalary(), 1000); |
| Понизить сотру  дника в должности  demoteEmployee  Success | Позитивный | provider.  demoteEmployee  1) Проверка метода при правильно введенных  всех данных | 1) True значение:  assertTrue((boolean)provider.  DemoteEmployee(21, 53).get(0)); |
| Понизить сотру  дника в должности  demoteEmployee  Fail | Негативный | provider.  demoteEmployee  1)Проверка метода  при попытке понизить сотрудника  в должности который занимает должность  Junior  2) Проверка метода  Если будет введен  несуществующий  id начальника  отдела | 1) False значение:  assertFalse((boolean)provider.  DemoteEmployee(20, 53).get(0));  2) False значение:  assertFalse((boolean)provider.  DemoteEmployee(20, 90).get(0)); |
| Понизить заработ  ную плату сотруд  нику lowerSalary  EmployeeSuccess | Позитивный | 1)Проверка метода  при правильном  изменении  заработной  платы сотруднику | 1) Заработная плата:  assertEquals(employee1.  getSalary(),provider.lower  SalaryEmployee(employee2).  getSalary()); |
| Понизить заработ  ную плату сотруд  нику lowerSalary  EmployeeFail | Негативный | 1)Проверка метода  при неправильном  изменении  заработной  платы сотруднику | 1) Заработная плата:  assertEquals(employee1.  getSalary(),provider.lower  SalaryEmployee(employee2).  getSalary(), 1000); |
| removeEmployee  Success | Позитивный | provider.  removeEmployee  1) Проверка метода при правильно введенных  всех данных | 1) True значение:  assertTrue(provider.  removeEmployee(24, 52)); |
| removeEmployee  Fail | Негативный | provider.  removeEmployee  1) Проверка метода если  будет введен  несуществующий id  сотрудника  2) Проверка метода  Если будет введен  несуществующий  id начальника  отдела | 1) False значение:  assertFalse(provider.  removeEmployee(30, 52));  2) False значение:  assertFalse(provider.  removeEmployee(23, 69)); |
| changeStatusOutfit  Success | Позитивный | provider.  changeStatusOutfit  1) Проверка метода  при правильном  названии расширяющего метода для вызова  changeStatusToInWorks  2) Проверка метода  при правильном  названии расширяющего метода для вызова  finish\_works | 1)Статус выполнения  метода:  assertEquals(Status.SUCCESS, provider.changeStatus  Outfit("change\_status\_to\_in\_work"  , 2, 63));  2)Статус выполнения  метода:  assertEquals(Status.SUCCESS, provider.changeStatus  Outfit("finish\_work", 2, 63)); |
| changeStatusOutfit  Fail | Негативный | provider.  changeStatusOutfit  1) Проверка метода  при неправильном  названии расширяющего метода для вызова  changeStatusToInWorks  2) Проверка метода  при неправильном  названии расширяющего метода для вызова  finish\_works | 1)Статус выполнения  метода:  assertEquals(Status.FAIL,  provider.change  StatusOutfit("change\_status\_to\_in  \_works", 2, 63));  2)Статус выполнения  метода:  assertEquals(Status.FAIL, provider.changeStatus  Outfit("finish\_works", 2, 63)); |
| changeStatusToIn  WorkSuccess | Позитивный | provider.  changeStatusToInWork  1) Проверка метода при правильно введенных  всех данных | 1) True значение:  assertTrue((boolean)provider.  ChangeStatusToInWork( 2, 63).get(0)); |
| changeStatusToIn  WorkFail | Негативный | provider.  changeStatusToInWork  1) Проверка метода если  будет введен несущест  вующий id наряда  2) Проверка метода если  будет введен id испол  нителя не этого наряда | 1) False значение:  assertFalse((boolean)provider.change  StatusToInWork( 3, 63).get(0));  2) False значение:  assertFalse((boolean)provider.  ChangeStatusToInWork( 2, 62).get(0)); |
| finishWorkSuccess | Позитивный | provider.  finishWork  1) Проверка метода при правильно введенных  всех данных | 1) True значение:  assertTrue((boolean)provider.  FinishWork(2, 63).get(0)); |
| finishWorkFail | Негативный | provider.  finishWork  1) Проверка метода если  будет введен несущест  вующий id наряда  2) Проверка метода если  будет введен id испол  нителя не этого наряда | 1) False значение:  assertFalse((boolean)provider.finish  Work( 3, 63).get(0));  2) False значение:  assertFalse((boolean)provider.finish  Work( 2, 62).get(0)); |