

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра ВТ**

**КУРСОВАЯ РАБОТА**  
**по дисциплине «Объектно-ориентированное программирование»**  
**Тема: «Разработка ПК для руководителя проектов»**

Студент гр. 1307

Таланков В.Р.

Преподаватель

---

---

Гречухин М.Н.

Санкт-Петербург

2023

## **ЗАДАНИЕ**

### **НА КУРСОВУЮ РАБОТУ (КУРСОВОЙ ПРОЕКТ)**

Разработать ПК для руководителя проектов. В ПК должны храниться сведения о сотрудниках отдела, о клиентах компании, о проектах, которые выполняются для клиентов, о заданиях, выданных сотрудникам.

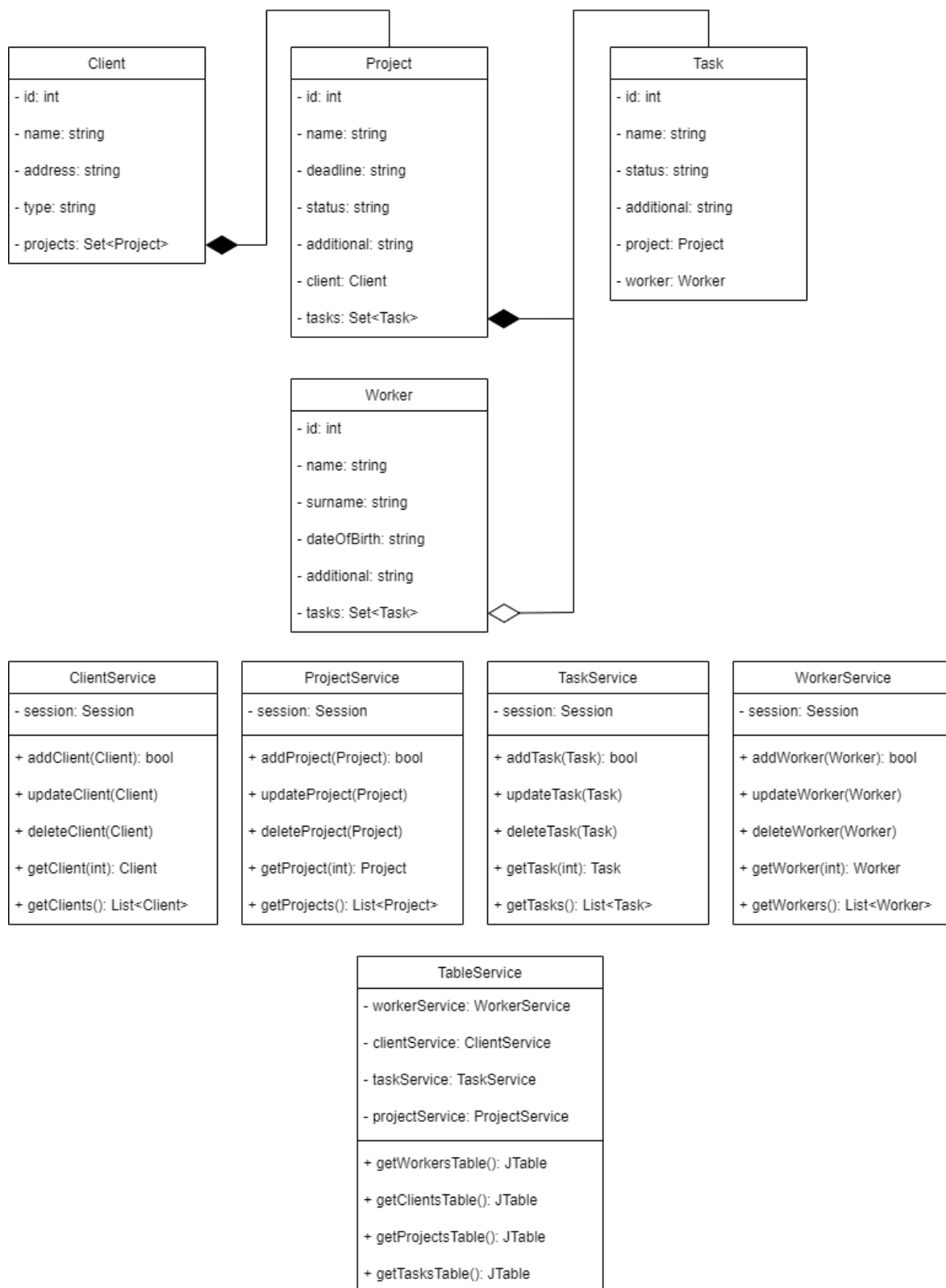
Руководитель проектов может добавлять, изменять и удалять эту информацию. Ему могут потребоваться следующие сведения:

- Перечень проектов, выполняемых для определённого клиента
- Перечень проектов, по которым нарушаются сроки выполнения заданий
- Список сотрудников, занятых на определённом проекте
- Загрузка выбранного сотрудника: в каких проектах и над какими задачами он в настоящее время работает

В итоге был разработан ПК с полной поддержкой требуемых выше функций, учетом клиентов, проектов, задач и работников.

# UML ДИАГРАММА КЛАССОВ

Классы данных и классы для работы с БД.



## ОПИСАНИЕ КЛАССОВ

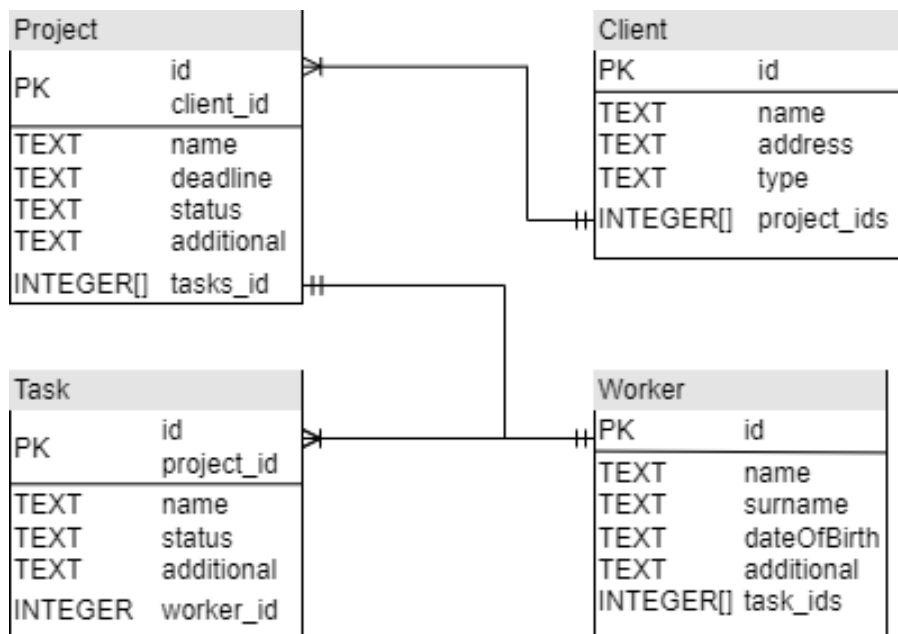
В работе используются 4 класса данных.

- **Client.** Представляет собой класс клиента отдела. Имеет поля `id` – уникальный идентификатор, `name` – имя клиента (компании), `address` – адрес (местоположение компании), `type` – тип компании (частная или государственная), `projects` – связанные с клиентом проекты.
- **Project.** Представляет собой класс проекта отдела. Имеет поля `id` – уникальный идентификатор, `name` – имя проекта, `deadline` – установленный срок завершения проекта, `status` – действующий статус проекта (в работе или срок выполнения истек), `additional` – дополнительная информация о проекте, `client` – связанный с проектом клиент, `tasks` – связанные с проектом задачи.
- **Task.** Представляет собой класс задачи проекта. Имеет поля `id` – уникальный идентификатор, `name` – имя задачи, `status` – действующий статус проекта (выполнен или не выполнен), `additional` – дополнительная информация о задаче, `project` – связанный с задачей проект, `worker` – связанный с задачей работник.
- **Worker.** Представляет собой класс работника. Имеет поля `id` – уникальный идентификатор, `name` – имя работника, `surname` – фамилия работника, `dateOfBirth` – дата рождения работника, `additional` – дополнительная информация о работнике, `tasks` – связанные с работником задачи.

Также в работе использованы 4 класса для работы с БД и 1 класс для создания таблиц с данными из БД.

- **ClientService.** Класс сервиса для работы с таблицей Client БД. Имеет методы: `addClient(Client client)` – добавление записи в таблицу Client, `updateClient(Client client)` – обновление записи, `deleteClient(Client client)` – удаление записи, `getClient(int id)` – получение объекта класса Client по его id, `getClients()` – получение списка объектов Client (всех записей хранящихся в таблице Client).
- **ProjectService.** Класс сервиса для работы с таблицей Project БД. Имеет методы: `addProject(Project project)` – добавление записи в таблицу Project, `updateProject(Project project)` – обновление записи, `deleteProject(Project project)` – удаление записи, `getProject(int id)` – получение объекта класса Project по его id, `getProjects` – получение списка объектов Project (всех записей хранящихся в таблице Project).
- **TaskService.** Класс сервиса для работы с таблицей Task БД. Имеет методы: `addTask` – добавление записи в таблицу Task, `updateTask` – обновление записи, `deleteTask` – удаление записи, `getTask` – получение записи по id, `getTasks` – получение всех записей.
- **WorkerService.** Класс сервиса для работы с таблицей Worker БД. Имеет методы: `addWorker` – добавление записи в таблицу Worker, `updateWorker` – обновление записи, `deleteWorker` – удаление записи, `getWorker` – получение записи по id, `getWorkers` – получение всех записей.
- **TableService.** Класс сервиса для создания таблиц класса JTable, использующихся в графическом интерфейсе. Имеет методы: `getClientsTable` – формирование графической таблицы из таблицы Client БД, `getProjectTable` – формирование графической таблицы из таблицы Project, `getTaskTable` – формирование графической таблицы из таблицы Task, `getWorkerTable` – формирование графической таблицы из таблицы Worker.

## СХЕМА ДАННЫХ



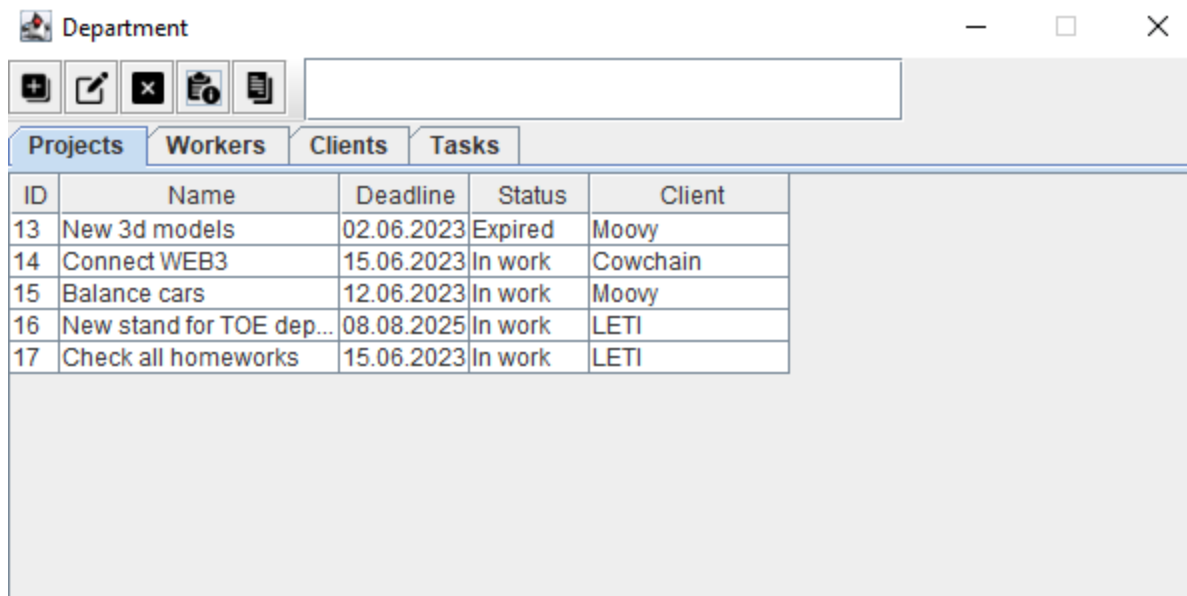
В работе представлены 4 таблицы БД:

- Таблица Client представляет собой класс клиента. Имеет поля: id – индивидуальный идентификатор, name – имя клиента, address – адрес (местоположении компании), type – тип компании (частная или государственная), project\_ids – список идентификаторов проектов, связанных с клиентом.
- Таблица Project представляет собой класс проекта. Имеет поля: id – индивидуальный идентификатор, client\_id – идентификатор клиента, связанного с данным проектом, name – название проекта, deadline – срок выполнения проекта, status – статус проекта (просрочен срок выполнения или нет), additional – дополнительная информация о проекте, tasks\_id – список идентификаторов задач, связанных с проектом.
- Таблица Task представляет собой класс задачи. Имеет поля: id – индивидуальный идентификатор, project\_id – идентификатор проекта, связанного с данной задачей, name – название задачи, status – статус задачи (выполнена или нет), additional – дополнительная информация о проекте, worker\_id – идентификатор работника, связанного с задачей.

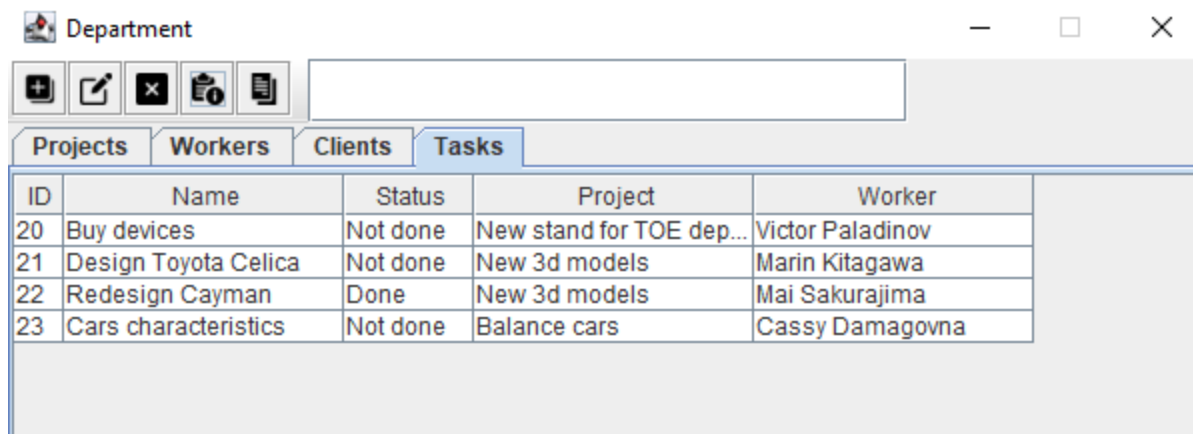
- Таблица `Worker` представляет собой класс работника. Имеет поля: `id` – индивидуальный идентификатор, `name` – имя работника, `surname` – фамилия работника, `dateOfBirth` – дата рождения работника, `additional` – дополнительная информация о работнике.

## ПРИМЕРЫ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА

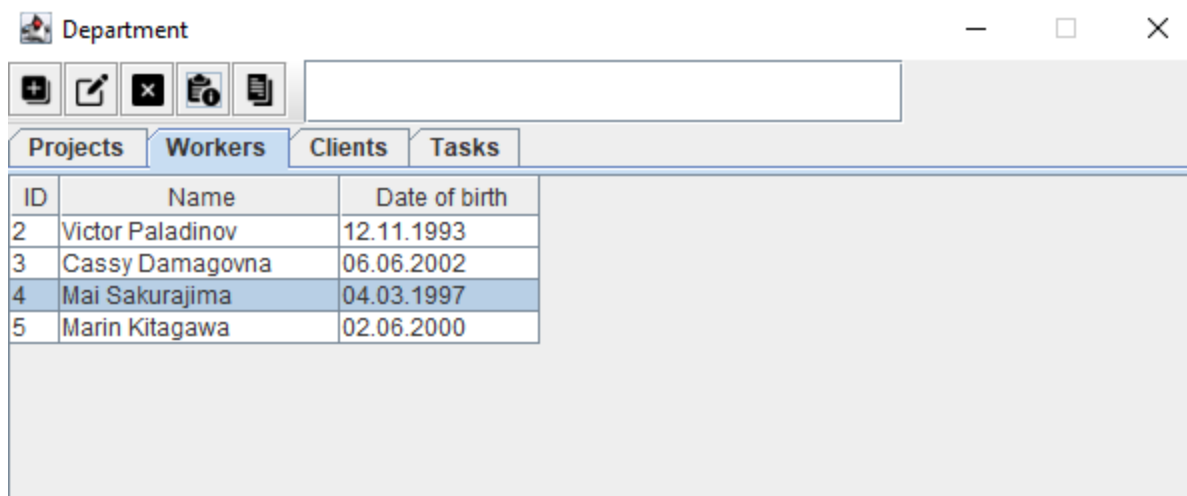
### Начальный экран программы



### Вкладка задач



### Вкладка работников





## Вкладка клиентов

Department

+ ✎ ✕ 📋 📄

Projects Workers **Clients** Tasks

ID	Name	Address	Type
7	Moovy	Spain	Private
8	Cowchain	Poland	Private
9	LETI	Russia, St. Petersburg	Government

## Дополнительная информация (проект)

About Project

**Name:** New 3d models

**Additional:** Design new models and 2d sprites

**Deadline:** 02.06.2023

**Status:** In work

**Client:** Moovy

**Tasks:** Task: Design Toyota Celica  
Worker: Marin Kitagawa  
Task: Redesign Cayman  
Worker: Mai Sakurajima

OK

## Пример добавления записи (проекта)

Add Task ×

Name:

Additional:

Deadline:  ?

Client:  ▼

- Moovy
- Cowchain
- LETI

Add Task ×

Name:

Additional:

Deadline:  ?

Client:  ▼

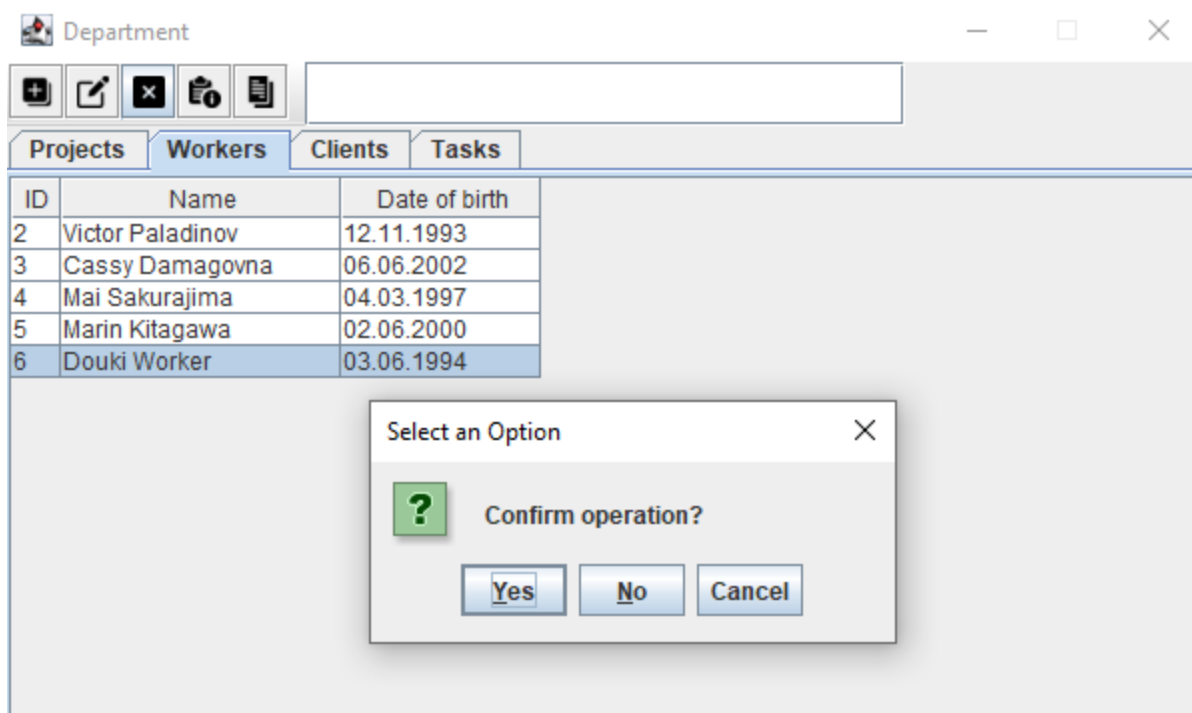
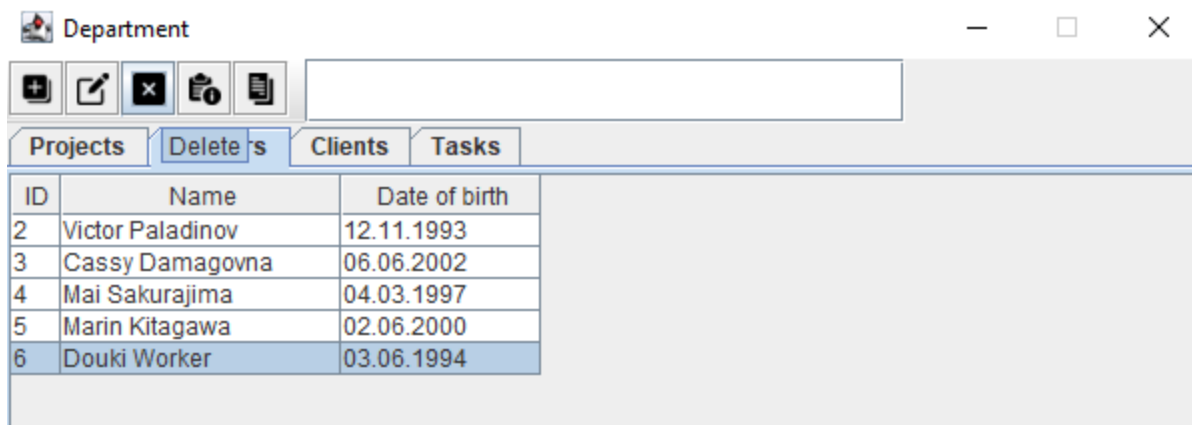
Calendar overlay:

< June > 2023

Sun	Mon	Tue	Wed	Thu	Fri	Sat
X	5	6	7	8	X	X
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Today is 05 June 2023

## Пример удаления записи (работника)



## **КРАТКАЯ ИНСТРУКЦИЯ ДЛЯ ПОЛЬЗОВАТЕЛЯ**

Взаимодействие пользователя с записями производится посредством панели инструментов в левом верхнем углу экрана. Для добавления записи необходимо нажать кнопку добавления и внести данные, после чего запись появится в таблице. Для изменения или удаления записи требуется сначала выделить ее, а после использовать кнопки из панели инструментов. Для получения полной информации о записи необходимо выделить ее и нажать кнопку информации, после чего будет отображена вся имеющаяся информация. Так же присутствует строка поиска с помощью, которой можно находить требуемую информацию по всем таблицам. Связи между проектами, задачами и работниками можно получить в дополнительной информации об этих записях.

## **ЗАКЛЮЧЕНИЕ**

Было разработано приложение для работы с записями, хранящимися в базе данных. В ходе работы были применены знания и навык, полученные в ходе курса, объектно-ориентированного программирование, в частности, применялись основные концепции ООП, такие как полиморфизм и инкапсуляция. В качестве основного языка программирования был выбран Java в связке с библиотекой Swing для создания графического интерфейса, для базы данных использовался язык SQL. В конечном итоге был разработан ПК полностью соответствующий заявленному техническому заданию.