**Описание классов**

Класс **Database** представляет собой простую базу данных для хранения информации о сотрудниках и задачах. Вот краткое описание каждого метода:

1. **addEmployeeToDatabaseFile()**: Добавляет нового сотрудника в базу данных.
2. **addTaskToDatabaseFile()**: Добавляет новую задачу в базу данных.
3. **removeEmployeeFromFile(const std::string& name)**: Удаляет сотрудника из базы данных по имени.
4. **removeTaskFromFile(const std::string& taskName)**: Удаляет задачу из базы данных по названию.
5. **displayEmployeesFromFile()**: Отображает всех сотрудников, хранящихся в базе данных.
6. **displayTasksFromFile()**: Отображает все задачи, хранящиеся в базе данных.
7. **advanceTime(int days)**: Продвигает время вперёд на указанное количество дней.
8. **employeeHasAssignedTask(const std::string& employeeName)**: Проверяет, есть ли у сотрудника назначенная задача.
9. **assignTask(const std::string& employeeName, const std::string& taskDescription)**: Назначает задачу определенному сотруднику.
10. **getTaskStatus(const std::string& taskDescription)**: Получает статус задачи по её описанию.

В классе также имеются приватные переменные **employees** и **tasks**, которые представляют собой векторы сотрудников и задач соответственно, используемые для хранения данных в базе.

Класс **Employee** представляет сотрудника компании. Его методы позволяют получать информацию о сотруднике и его назначенной задаче:

1. **Employee(const std::string& name, int hoursPerDay)**: Конструктор класса, который инициализирует объект сотрудника с указанным именем и количеством рабочих часов в день.
2. **getName() const**: Возвращает имя сотрудника.
3. **hasTaskAssigned() const**: Проверяет, назначена ли сотруднику задача.
4. **assignTask(Task\* task)**: Назначает сотруднику задачу.
5. **getAssignedTask()**: Возвращает указатель на назначенную сотруднику задачу.
6. **getHoursPerDay() const**: Возвращает количество рабочих часов в день для сотрудника.

В классе **Employee** есть три приватные переменные:

1. **std::string name**: Строка, содержащая имя сотрудника.
2. **int hoursPerDay**: Целочисленное значение, представляющее количество рабочих часов в день для сотрудника.
3. **Task\* assignedTask**: Указатель на объект класса **Task**, представляющий назначенную сотруднику задачу.

Класс **Menu** представляет меню для взаимодействия с базой данных. Его методы обеспечивают отображение меню и взаимодействие с базой данных:

1. **Menu(Database& db)**: Конструктор класса, принимающий ссылку на объект базы данных и инициализирующий член **database**.
2. **showMenu()**: Метод для отображения меню, предоставляющий пользователю различные опции для взаимодействия с базой данных.

В классе **Menu** есть одна приватная переменная:

1. **Database& database**: Ссылка на объект класса **Database**, который представляет базу данных, с которой будет взаимодействовать меню.

Класс **Task** представляет задачу, которая может быть назначена сотруднику. Его методы обеспечивают доступ к информации о задаче и управление её статусом:

1. **Task(const std::string& description, const std::string& deadline, int timeNeeded)**: Конструктор класса, инициализирующий объект задачи с указанным описанием, крайним сроком выполнения и необходимым временем для выполнения.
2. **getStatus() const**: Возвращает текущий статус задачи.
3. **setStatus(const std::string& newStatus)**: Устанавливает новый статус задачи.
4. **getDescription() const**: Возвращает описание задачи.
5. **getDeadline() const**: Возвращает крайний срок выполнения задачи.
6. **getTimeNeeded() const**: Возвращает необходимое время для выполнения задачи.

В классе **Task** есть четыре приватные переменные:

1. **std::string description**: Строка, содержащая описание задачи.
2. **std::string status**: Строка, представляющая статус задачи. Изначально устанавливается в "Not Assigned".
3. **std::string deadline**: Строка, содержащая крайний срок выполнения задачи.
4. **int timeNeeded**: Целочисленное значение, представляющее необходимое количество времени для выполнения задачи.