МИНОБРНАУКИ РОССИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Пермский государственный национальный
исследовательский университет»

Институт компьютерных наук и
технологий

**ОТЧЁТ**

по индивидуальной работе №1
по дисциплине «Языки программирования Python»
Вариант 8

Работу выполнила
студентка группы ИТ-7, 2025 1 курса
Злобина Дарья Сергеевна
13 января 2025 г.

Работу проверила
Шилина Алла Владимировна
«___» _____ 2025 г.

Пермь 2025

**СОДЕРЖАНИЕ**

**Постановка задачи**

Создать файл записей, в котором хранится информация о вакансиях в кадровом агентстве: должность, необходимый стаж работы, пол, образование, минимальный возраст, максимальный возраст, знание иностранных языков, минимальный оклад, наличие соцпакета, испытательный срок.

Разработать и реализовать программу "Кадровое агентство", которая считывает исходную информацию и позволяет на основе неё создавать следующие отчёты:

1) Полный список всех вакансий, который будет отсортирован следующему ключу: образование (по возрастанию) + должность (по возрастанию).

2) Список всех вакансий с испытательным сроком не менее 2 месяцев, отсортированный по следующему ключу: испытательный срок (по убыванию) + необходимый стаж работы (по убыванию) + максимальный возраст (по возрастанию).

3) Список всех вакансий, с окладом в диапазоне от N1 до N2 рублей, отсортированный по следующему ключу: наличие соцпакета (по возрастанию) + испытательный срок (по убыванию).

Создать базу вакансий, включающую не менее 25 записей и на основе неё сформировать все указанные списки. База должна содержать такие записи, чтобы во всех списках явно прослеживался заданный вид сортировки по всем ключам. Для сортировки записей использовать сортировку Хоара.

**Алгоритм решения**

**Структуры данных**

Для хранения вакансий используется список словарей. Каждая вакансия представлена словарём с ключами, соответствующими полям файла. Для образования определены уровни с рангами:

"Без образования" → 1

"Среднее" → 2

"Среднее специальное" → 3

"Высшее" → 4

Это позволяет сортировать текстовые значения образования как числовые.

**Алгоритм быстрой сортировки (Хоара)**

Реализована функция quick_sort(), которая принимает массив и функцию-ключ. Алгоритм**:**

Выбор опорного элемента.

Разделение массива на три части: меньше опорного, равные опорному, больше опорного.

Рекурсивная сортировка левой и правой частей.

Объединение отсортированных частей

**Функции-ключи для сортировки**

key_report1(): возвращает кортеж (ранг образования, название должности) для сортировки по возрастанию.

key_report2(): возвращает кортеж (-испытательный срок, -стаж, максимальный возраст). Отрицательные значения обеспечивают сортировку по убыванию.

key_report3(): возвращает кортеж (0/1 для соцпакета, -испытательный срок).

**Работа с файлами**

load_vacancies() – чтение данных из vacancies.txt с валидацией и преобразованием типов и обработкой ошибок.

save_vacancies() – сохранение данных обратно в файл с заголовком.

**Пользовательский интерфейс**

Реализовано циклическое консольное меню с 9 пунктами:

- Показать все вакансии
- Добавить вакансию
- Изменить вакансию
- Удалить вакансию
- Отчет 1: Все вакансии
- Отчет 2: Испытательный срок ≥ 2 мес
- Отчет 3: По диапазону зарплат
- Сохранить базу
- Выход

Каждое действие сопровождается проверкой ввода и обработкой ошибок.

**Архитектура программы**

Программа разделена на логические модули:

Работа с файлами — загрузка и сохранение.

Валидация — проверка всех типов данных.

Сортировка — алгоритм Хоара и функции-ключи.

Операции с вакансиями — добавление, изменение, удаление.

Отчеты — три типа отчетов с фильтрацией и сортировкой.

Интерфейс — главное меню и вспомогательные функции отображения.

**Тестирование**

Загрузка вакансий из файла

```
Выбор (1-9): 1

ВСЕ ВАКАНСИИ
----------------------------------------------------------------------
№   | Должность           | Образование        |    Оклад | Стаж | Исп.срок | Соцпакет
----------------------------------------------------------------------
  1 | Программист Python  | Высшее             |   120000₽ |    3л |        3мес | Да
  2 | Менеджер по продажам | Среднее            |    80000₽ |    2л |        2мес | Нет
  3 | Бухгалтер           | Высшее             |    90000₽ |    5л |        1мес | Да
  4 | Системный администра | Среднее специально |    95000₽ |    4л |        3мес | Да
  5 | Дизайнер            | Высшее             |    85000₽ |    2л |        2мес | Нет
  6 | Аналитик данных     | Высшее             |   130000₽ |    3л |        3мес | Да
  7 | Юрист               | Высшее             |   150000₽ |    7л |        1мес | Да
  8 | Маркетолог          | Высшее             |    95000₽ |    3л |        2мес | Да
  9 | Тестировщик         | Среднее специально |    70000₽ |    1л |        3мес | Нет
 10 | DevOps инженер      | Высшее             |   180000₽ |    4л |        2мес | Да
 11 | Копирайтер          | Высшее             |    60000₽ |    2л |        1мес | Нет
 12 | HR-менеджер         | Высшее             |    85000₽ |    3л |        2мес | Да
 13 | Секретарь           | Среднее            |    45000₽ |    1л |        1мес | Нет
 14 | Продавец-консультант | Среднее            |    40000₽ |    0л |        2мес | Нет
 15 | Водитель            | Среднее            |    70000₽ |    5л |        1мес | Да
 16 | Повар               | Среднее специально |    60000₽ |    3л |        2мес | Нет
 17 | Врач-терапевт       | Высшее             |   120000₽ |   10л |        1мес | Да
 18 | Медсестра           | Среднее специально |    50000₽ |    3л |        1мес | Да
 19 | Уборщица            | Без образования    |    30000₽ |    0л |        1мес | Нет
 20 | Инженер             | Высшее             |   110000₽ |    5л |        3мес | Да
 21 | Архитектор          | Высшее             |   140000₽ |    6л |        2мес | Да
 22 | Переводчик          | Высшее             |    90000₽ |    4л |        2мес | Нет
 23 | Логист              | Высшее             |    95000₽ |    4л |        2мес | Да
 24 | Экономист           | Высшее             |   100000₽ |    5л |        3мес | Да
 25 | Библиотекарь        | Высшее             |    40000₽ |    2л |        1мес | Да
----------------------------------------------------------------------
Всего: 25
```

Добавление новой вакансии с корректными данными

```
Выбор (1-9): 2

ДОБАВЛЕНИЕ НОВОЙ ВАКАНСИИ

Образование:
1. Без образования
2. Среднее
3. Среднее специальное
4. Высшее
Должность: Медсестра
Стаж (лет): 1
Пол (М/Ж): Ж
Образование: 4
Мин.возраст: 23
Макс.возраст: 50
Языки (Enter - пропустить):
Оклад (руб.): 50000
Соцпакет (да/нет): да
Исп.срок (мес.): 3
Вакансия 'Медсестра' добавлена!
Сохранено вакансий: 26
```

Проверяем, что вакансия добавлена

```
 20 | Инженер              | Высшее              |   110000₽ |    5л |        3мес | Да
 21 | Архитектор           | Высшее              |   140000₽ |    6л |        2мес | Да
 22 | Переводчик           | Высшее              |    90000₽ |    4л |        2мес | Нет
 23 | Логист               | Высшее              |    95000₽ |    4л |        2мес | Да
 24 | Экономист            | Высшее              |   100000₽ |    5л |        3мес | Да
 25 | Библиотекарь         | Высшее              |    40000₽ |    2л |        1мес | Да
 26 | Медсестра            | Высшее              |    50000₽ |    1л |        3мес | Да
--------------------------------------------------------------------------
Всего: 26
```

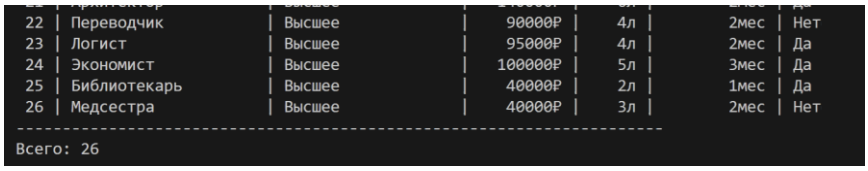
Добавление новой вакансии с некорректными данными

```
Должность: Юрист
Стаж (лет): десять
Ошибка: введите целое число
Стаж (лет): 10
Пол (М/Ж): 66
Ошибка: допустимые значения: 'М' или 'Ж'
Пол (М/Ж): М
Образование: есть
Ошибка: выберите из списка: 1.Без образования, 2.Среднее, 3.Среднее специальное, 4.Высшее
Образование: 4
Мин.возраст: 10
Ошибка: значение не может быть меньше 18
Мин.возраст: 28
Макс.возраст: 1009
Ошибка: значение не может быть больше 100
Макс.возраст: 70
Языки (Enter - пропустить):
Оклад (руб.): в
Ошибка: введите целое число
Оклад (руб.): 60000
Соцпакет (да/нет): есть
Ошибка: введите 'да' или 'нет'
Соцпакет (да/нет): да
Исп.срок (мес.): 16
Ошибка: значение не может быть больше 12
Исп.срок (мес.): 10
Вакансия 'Юрист' добавлена!
Сохранено вакансий: 27
```

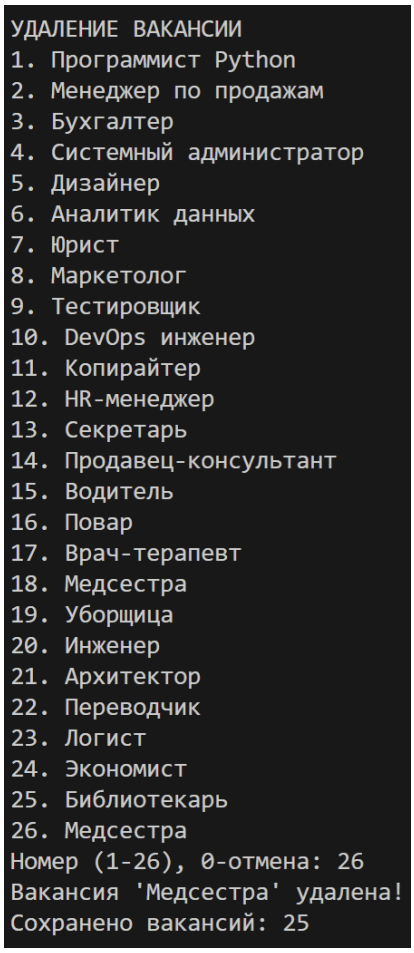
Изменение вакансии

```
Редактирование: Медсестра
(Enter - оставить текущее)
Должность [Медсестра]:
Стаж [1]: 3
Пол [Ж]: ж
Образование [Высшее]:
Мин.возраст [23]: 20
Макс.возраст [50]: 40
Языки [Не указано]:
Оклад [50000]: 40000
Соцпакет [Да] (да/нет): да
Исп.срок [3]: 2
✓ Вакансия изменена!
Сохранено вакансий: 26
```

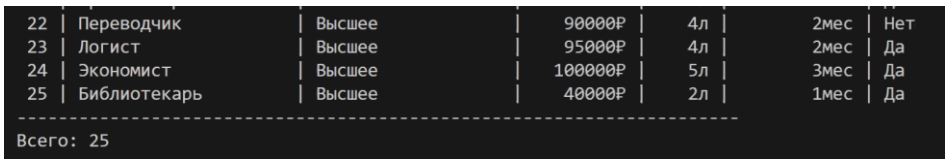
Проверяем, что вакансия изменилась



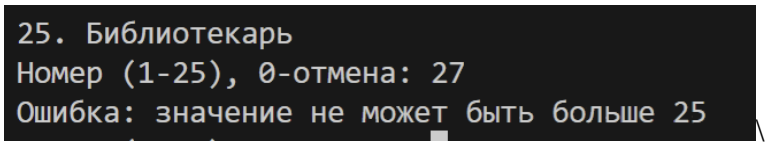

Удаление вакансии




Проверяем, что вакансия удалена.




При попытке удалить несуществующий номер - ошибка



Генерация 1 отчета: сортировка образования (по возрастанию) и должности (по возрастанию).

ОТЧЕТ 1: ВСЕ ВАКАНСИИ (Образование↑ + Должность↑)

| № | Должность | Образование | Оклад | Стаж | Исп.срок | Соцпакет |
|---|---|---|---|---|---|---|
| 1 | Уборщица | Без образования | 30000₽ | 0л | 1мес | Нет |
| 2 | Водитель | Среднее | 70000₽ | 5л | 1мес | Да |
| 3 | Менеджер по продажам | Среднее | 80000₽ | 2л | 2мес | Нет |
| 4 | Продавец-консультант | Среднее | 40000₽ | 0л | 2мес | Нет |
| 5 | Секретарь | Среднее | 45000₽ | 1л | 1мес | Нет |
| 6 | Медсестра | Среднее специально | 50000₽ | 3л | 1мес | Да |
| 7 | Повар | Среднее специально | 60000₽ | 3л | 2мес | Нет |
| 8 | Системный администра | Среднее специально | 95000₽ | 4л | 3мес | Да |
| 9 | Тестировщик | Среднее специально | 70000₽ | 1л | 3мес | Нет |
| 10 | DevOps инженер | Высшее | 180000₽ | 4л | 2мес | Да |
| 11 | HR-менеджер | Высшее | 85000₽ | 3л | 2мес | Да |
| 12 | Аналитик данных | Высшее | 130000₽ | 3л | 3мес | Да |
| 13 | Архитектор | Высшее | 140000₽ | 6л | 2мес | Да |
| 14 | Библиотекарь | Высшее | 40000₽ | 2л | 1мес | Да |
| 15 | Бухгалтер | Высшее | 90000₽ | 5л | 1мес | Да |
| 16 | Врач-терапевт | Высшее | 120000₽ | 10л | 1мес | Да |
| 17 | Дизайнер | Высшее | 85000₽ | 2л | 2мес | Нет |
| 18 | Инженер | Высшее | 110000₽ | 5л | 3мес | Да |
| 19 | Копирайтер | Высшее | 60000₽ | 2л | 1мес | Нет |
| 20 | Логист | Высшее | 95000₽ | 4л | 2мес | Да |
| 21 | Маркетолог | Высшее | 95000₽ | 3л | 2мес | Да |
| 22 | Переводчик | Высшее | 90000₽ | 4л | 2мес | Нет |
| 23 | Программист Python | Высшее | 120000₽ | 3л | 3мес | Да |
| 24 | Экономист | Высшее | 100000₽ | 5л | 3мес | Да |
| 25 | Юрист | Высшее | 150000₽ | 7л | 1мес | Да |

Всего: 25

Генерация 2 отчета: список всех вакансий с испытательным сроком не менее 2 месяцев, отсортированный по следующему ключу: испытательный срок (по убыванию) + необходимый стаж работы (по убыванию) + максимальный возраст (по возрастанию).

ОТЧЕТ 2: ИСПЫТАТЕЛЬНЫЙ СРОК ≥ 2 МЕСЯЦЕВ

| № | Должность | Образование | Оклад | Стаж | Исп.срок | Соцпакет |
|---|---|---|---|---|---|---|
| 1 | Инженер | Высшее | 110000₽ | 5л | 3мес | Да |
| 2 | Экономист | Высшее | 100000₽ | 5л | 3мес | Да |
| 3 | Системный администра | Среднее специально | 95000₽ | 4л | 3мес | Да |
| 4 | Программист Python | Высшее | 120000₽ | 3л | 3мес | Да |
| 5 | Аналитик данных | Высшее | 130000₽ | 3л | 3мес | Да |
| 6 | Тестировщик | Среднее специально | 70000₽ | 1л | 3мес | Нет |
| 7 | Архитектор | Высшее | 140000₽ | 6л | 2мес | Да |
| 8 | Переводчик | Высшее | 90000₽ | 4л | 2мес | Нет |
| 9 | DevOps инженер | Высшее | 180000₽ | 4л | 2мес | Да |
| 10 | Логист | Высшее | 95000₽ | 4л | 2мес | Да |
| 11 | Маркетолог | Высшее | 95000₽ | 3л | 2мес | Да |
| 12 | HR-менеджер | Высшее | 85000₽ | 3л | 2мес | Да |
| 13 | Повар | Среднее специально | 60000₽ | 3л | 2мес | Нет |
| 14 | Дизайнер | Высшее | 85000₽ | 2л | 2мес | Нет |
| 15 | Менеджер по продажам | Среднее | 80000₽ | 2л | 2мес | Нет |
| 16 | Продавец-консультант | Среднее | 40000₽ | 0л | 2мес | Нет |

Всего: 16

Генерация 3 отчета: список всех вакансий, с окладом в диапазоне от N1 до N2 рублей, отсортированный по следующему ключу: наличие соцпакета (по возрастанию) + испытательный срок (по убыванию).

```
Выбор (1-9): 7
Мин.оклад: 20000
Макс.оклад: 50000

ОТЧЕТ 3: ОКЛАД ОТ 20000 ДО 50000 РУБ.
------------------------------------------------------------------------
№   | Должность           | Образование        |    Оклад | Стаж | Исп.срок | Соцпакет
------------------------------------------------------------------------
  1 | Медсестра           | Среднее специально |    50000₽ |    3л |       1мес | Да
  2 | Библиотекарь        | Высшее             |    40000₽ |    2л |       1мес | Да
  3 | Продавец-консультант | Среднее            |    40000₽ |    0л |       2мес | Нет
  4 | Секретарь           | Среднее            |    45000₽ |    1л |       1мес | Нет
  5 | Уборщица            | Без образования    |    30000₽ |    0л |       1мес | Нет
------------------------------------------------------------------------
Всего: 5
```

При попытке ввести минимальный оклад больше максимального, возникает ошибка

```
Выбор (1-9): 7
Мин.оклад: 50000
Макс.оклад: 10000
Ошибка: значение не может быть меньше 50000
```

9) Сохранение базы данных

```
Выбор (1-9): 8
Сохранено вакансий: 25
```

10) Выход из базы данных

```
Выбор (1-9): 9

До свидания!
```

11) Попытка запустить код без файла с базой данных

```
============================================================
Ошибка: файл vacancies.txt не найден!

Создайте файл vacancies.txt и перезапустите программу.
```

**Код программы**

Ссылка на GitHub: https://github.com/dashazlobina/Individual-Task

```
#КАДРОВОЕ АГЕНТСТВО - ПРОГРАММА ДЛЯ УПРАВЛЕНИЯ БАЗОЙ ВАКАНСИЙ, 
вариант 8

vacancies_db = []

#Список уровней образования
EDUCATION_LEVELS = ["Без образования", "Среднее", "Среднее 
специальное", "Высшее"]

#Словарь с номерами для каждого уровня
EDUCATION_RANKS = {
    "Без образования": 1,
    "Среднее": 2,
    "Среднее специальное": 3,
    "Высшее": 4
}

#ФУНКЦИИ ДЛЯ РАБОТЫ С ФАЙЛАМИ

def load_vacancies():
    """Загрузка вакансий из файла vacancies.txt"""
    global vacancies_db
    try:
        with open("vacancies.txt", 'r', encoding='utf-8') as f:
            next(f)  #Пропускаем заголовок
            vacancies_db = []
            for line in f:
                if line.strip():
                    parts = line.strip().split(',')
                    if len(parts) >= 10:
                        vacancy = {
                            'position': parts[0].strip(),
                            'experience': int(parts[1].strip()),
                            'gender': parts[2].strip(),
                            'education': parts[3].strip(),
                            'min_age': int(parts[4].strip()),
                            'max_age': int(parts[5].strip()),
                            'languages': parts[6].strip() or "Не 
указано",
                            'min_salary': int(parts[7].strip()),
                            'social_package': 
parts[8].strip().lower() in ['true', 'да', 'yes', '1'],
                            'probation_period': 
int(parts[9].strip())
                        }
                        vacancies_db.append(vacancy)
```

```
        print(f"Загружено вакансий: {len(vacancies_db)}")
        return True
    except FileNotFoundError:
        print("Ошибка: файл vacancies.txt не найден!")
        return False
    except Exception as e:
        print(f"Ошибка при загрузке файла: {e}")
        return False

def save_vacancies():
    """Сохранение вакансий в файл vacancies.txt"""
    try:
        with open("vacancies.txt", 'w', encoding='utf-8') as f:

f.write("Должность,Стаж,Пол,Образование,Мин.возраст,Макс.возраст
,Языки,Оклад,Соцпакет,Исп.срок\n")
            for v in vacancies_db:

f.write(f"{v['position']},{v['experience']},{v['gender']},"

f"{v['education']},{v['min_age']},{v['max_age']},"
                        f"{v['languages']},{v['min_salary']},"
                        f"{'True'  if  v['social_package']  else
'False'},"
                        f"{v['probation_period']}\n")
        print(f"Сохранено вакансий: {len(vacancies_db)}")
        return True
    except Exception as e:
        print(f"Ошибка при сохранении: {e}")
        return False

#ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ ДЛЯ ВАЛИДАЦИИ

def get_input(prompt, validator=None, default=None):
    """Универсальная функция ввода с валидацией"""
    while True:
        value = input(prompt).strip()
        if not value and default is not None:
            return default
        if validator:
            result, error = validator(value)
            if result:
                return result
            print(f"Ошибка: {error}")
        elif value:
            return value
        else:
            print("Ошибка: поле не может быть пустым!")

def validate_gender(value):
    """Валидация пола"""
    value = value.upper()
```

```
    if value in ['М', 'М', 'МУЖ', 'MALE']:
        return 'М', None
    elif value in ['Ж', 'F', 'ЖЕН', 'FEMALE']:
        return 'Ж', None
    return None, "допустимые значения: 'М' или 'Ж'"

def validate_bool(value):
    """Валидация булевого значения"""
    value = value.lower()
    if value in ['да', 'д', 'yes', 'y', 'true', '1']:
        return True, None
    elif value in ['нет', 'н', 'no', 'n', 'false', '0']:
        return False, None
    return None, "введите 'да' или 'нет'"

def validate_int(value, min_val=None, max_val=None):
    """Валидация целого числа"""
    try:
        num = int(value.replace(" ", ""))
        if min_val is not None and num < min_val:
            return None, f"значение не может быть меньше
{min_val}"
        if max_val is not None and num > max_val:
            return None, f"значение не может быть больше
{max_val}"
        return num, None
    except ValueError:
        return None, "введите целое число"

def validate_str(value, min_len=1, allow_numbers=True):
    """Валидация строки"""
    if len(value) < min_len:
        return None, f"должно быть не менее {min_len} символов"
    if not allow_numbers and any(c.isdigit() for c in value):
        return None, "нельзя использовать цифры"
    return value, None

def validate_education(value):
    """Валидация образования"""
    if value.isdigit() and 1 <= int(value) <=
len(EDUCATION_LEVELS):
        return EDUCATION_LEVELS[int(value)-1], None
    for edu in EDUCATION_LEVELS:
        if value.lower() == edu.lower():
            return edu, None
    return None, f"выберите из списка: {', '.join(f'{i+1}.{e}' for
i,e in enumerate(EDUCATION_LEVELS))}"

#ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ ДЛЯ ПРОВЕРКИ ВВОДА

def check_position(value):
    """Проверка должности"""
```

```
    return validate_str(value, 2, False)

def check_experience(value):
    """Проверка стажа"""
    return validate_int(value, 0, 50)

def check_age_min(value):
    """Проверка минимального возраста"""
    return validate_int(value, 18, 100)

def check_salary(value):
    """Проверка оклада"""
    return validate_int(value, 0, 1000000)

def check_probation(value):
    """Проверка испытательного срока"""
    return validate_int(value, 0, 12)

def check_menu_choice(value):
    """Проверка выбора в меню"""
    return validate_int(value, 1, 9)

def check_vacancy_choice(value, max_num):
    """Проверка выбора вакансии"""
    return validate_int(value, 0, max_num)

def check_max_age(value, min_age):
    """Проверка максимального возраста"""
    return validate_int(value, min_age, 100)

def check_max_salary(value, min_salary):
    """Проверка максимального оклада"""
    return validate_int(value, min_salary, 1000000)

#СОРТИРОВКА ХОАРА И ФУНКЦИИ ДЛЯ КЛЮЧЕЙ

def quick_sort(arr, key_func=None):
    """Сортировка Хоара"""
    if len(arr) <= 1:
        return arr

    pivot = arr[len(arr)//2]
    pivot_key = key_func(pivot) if key_func else pivot

    left = [x for x in arr if (key_func(x) if key_func else x) <
pivot_key]
    middle = [x for x in arr if (key_func(x) if key_func else x)
== pivot_key]
    right = [x for x in arr if (key_func(x) if key_func else x) >
pivot_key]
```

```
    return quick_sort(left, key_func) + middle + quick_sort(right,
key_func)

def key_report1(v):
    """Ключ для отчета 1: образование (↑) + должность (↑)"""
    return        (EDUCATION_RANKS.get(v['education'],        0),
v['position'].lower())

def key_report2(v):
    """Ключ для отчета 2: исп.срок (↓) + стаж (↓) + возраст (↑)"""
    return       (-v['probation_period'],       -v['experience'],
v['max_age'])

def key_report3(v):
    """Ключ для отчета 3: соцпакет (↑) + исп.срок (↓)"""
    return    (0    if    v['social_package']    else    1,    -
v['probation_period'])

#ФУНКЦИИ ДЛЯ РАБОТЫ С ВАКАНСИЯМИ

def add_vacancy():
    """Добавление новой вакансии"""
    print("\nДОБАВЛЕНИЕ НОВОЙ ВАКАНСИИ")

    print("\nОбразование:")
    for i in range(4):  #4 уровня образования
        print(f"{i+1}. {EDUCATION_LEVELS[i]}")

    position = get_input("Должность: ", check_position)
    experience = get_input("Стаж (лет): ", check_experience)
    gender = get_input("Пол (М/Ж): ", validate_gender)
    education = get_input("Образование: ", validate_education)

    min_age = get_input("Мин.возраст: ", check_age_min)
    max_age    =    get_input("Макс.возраст:    ",    lambda    v:
check_max_age(v, min_age))

    languages = input("Языки (Enter - пропустить): ").strip() or
"Не указано"
    min_salary = get_input("Оклад (руб.): ", check_salary)
    social = get_input("Соцпакет (да/нет): ", validate_bool)
    probation = get_input("Исп.срок (мес.): ", check_probation)

    vacancy = {
        'position': position,
        'experience': experience,
        'gender': gender,
        'education': education,
        'min_age': min_age,
        'max_age': max_age,
        'languages': languages,
        'min_salary': min_salary,
```

```
        'social_package': social,
        'probation_period': probation
    }

    vacancies_db.append(vacancy)
    print(f"Вакансия '{position}' добавлена!")
    save_vacancies()

def edit_vacancy():
    """Изменение вакансии"""
    if not vacancies_db:
        print("Нет вакансий!")
        return

    print("\nИЗМЕНЕНИЕ ВАКАНСИИ")

    #Показать список вакансий
    for i, v in enumerate(vacancies_db, 1):
        print(f"{i}. {v['position']} - {v['min_salary']}₽")

    #Выбрать вакансию
    choice = get_input(f"Номер (1-{len(vacancies_db)}), 0-отмена:
",
                       lambda      v:      check_vacancy_choice(v,
len(vacancies_db)))

    if choice == 0:
        return

    vacancy = vacancies_db[choice-1]
    print(f"\nРедактирование: {vacancy['position']}")
    print("(Enter - оставить текущее)")

    #Редактировать каждое поле
    def edit_field(field_name, check_func, current):
        value = input(f"{field_name} [{current}]: ").strip()
        if value:
            ok, result = check_func(value)
            if ok:
                return result
        return current

    #Функции для проверки при редактировании
    def check_position_edit(value):
        return validate_str(value, 2, False)

    def check_experience_edit(value):
        return validate_int(value, 0, 50)

    def check_min_age_edit(value):
        return validate_int(value, 18, vacancy['max_age'])
```

```
    def check_max_age_edit(value):
        return validate_int(value, vacancy['min_age'], 100)

    def check_salary_edit(value):
        return validate_int(value, 0, 1000000)

    def check_probation_edit(value):
        return validate_int(value, 0, 12)

    vacancy['position']         =         edit_field("Должность",
check_position_edit, vacancy['position'])
    vacancy['experience']           =           edit_field("Стаж",
check_experience_edit, vacancy['experience'])
    vacancy['gender']    =    edit_field("Пол",    validate_gender,
vacancy['gender'])
    vacancy['education']        =       edit_field("Образование",
validate_education, vacancy['education'])
    vacancy['min_age']         =        edit_field("Мин.возраст",
check_min_age_edit, vacancy['min_age'])
    vacancy['max_age']         =       edit_field("Макс.возраст",
check_max_age_edit, vacancy['max_age'])

    lang_input    =    input(f"Языки    [{vacancy['languages']}]:
").strip()
    if lang_input:
        vacancy['languages'] = lang_input

    vacancy['min_salary']           =           edit_field("Оклад",
check_salary_edit, vacancy['min_salary'])

    social_input      =      input(f"Соцпакет      [{'Да'      if
vacancy['social_package'] else 'Нет'}] (да/нет): ").strip()
    if social_input:
        ok, result = validate_bool(social_input)
        if ok:
            vacancy['social_package'] = result

    vacancy['probation_period']     =     edit_field("Исп.срок",
check_probation_edit, vacancy['probation_period'])

    print("✓ Вакансия изменена!")
    save_vacancies()

def delete_vacancy():
    """Удаление вакансии"""
    if not vacancies_db:
        print("Нет вакансий!")
        return

    print("\nУДАЛЕНИЕ ВАКАНСИИ")

    #Показать список
```

```
    for i, v in enumerate(vacancies_db, 1):
        print(f"{i}. {v['position']}")

    #Выбрать для удаления
    choice = get_input(f"Номер (1-{len(vacancies_db)})",
                       lambda      v:      check_vacancy_choice(v,
len(vacancies_db)))

    if choice > 0:
        deleted = vacancies_db.pop(choice-1)
        print(f"Вакансия '{deleted['position']}' удалена!")
        save_vacancies()

def show_vacancies(vacancies, title=""):
    """Показать вакансии"""
    if not vacancies:
        print("Нет данных")
        return

    print(f"\n{title if title else 'ВАКАНСИИ'}")
    print("-" * 70)
    print(f"{'№':3}  |  {'Должность':20}  |  {'Образование':18}  |
{'Оклад':>8} | Стаж | Исп.срок | Соцпакет")
    print("-" * 70)

    for i, v in enumerate(vacancies, 1):
        social = "Да" if v['social_package'] else "Нет"
        print(f"{i:3}      |      {v['position'][:20]:20}      |
{v['education'][:18]:18} | "
              f"{v['min_salary']:>8}₽ | {v['experience']:4}л | "
              f"{v['probation_period']:8}мес | {social}")

    print("-" * 70)
    print(f"Всего: {len(vacancies)}")

#ФУНКЦИИ ДЛЯ ОТЧЕТОВ

def generate_report1():
    """Отчет 1: Все вакансии по образованию и должности"""
    if vacancies_db:
        sorted_vac          =          quick_sort(vacancies_db,
key_func=key_report1)
        show_vacancies(sorted_vac,   "ОТЧЕТ   1:   ВСЕ   ВАКАНСИИ
(Образование↑ + Должность↑)")

def generate_report2():
    """Отчет 2: Вакансии с испытательным сроком ≥ 2 месяцев"""
    filtered = [v for v in vacancies_db if v['probation_period']
>= 2]
    if filtered:
        sorted_vac = quick_sort(filtered, key_func=key_report2)
```

```
        show_vacancies(sorted_vac, "ОТЧЕТ 2: ИСПЫТАТЕЛЬНЫЙ СРОК ≥
2 МЕСЯЦЕВ")
    else:
        print("Нет вакансий с испытательным сроком ≥ 2 месяцев")

def generate_report3():
    """Отчет 3: Вакансии по диапазону зарплат"""

    def check_min_salary(value):
        return validate_int(value, 0, 1000000)

    n1 = get_input("Мин.оклад: ", check_min_salary)

    def check_max_salary_for_report(value):
        return check_max_salary(value, n1)

    n2 = get_input("Макс.оклад: ", check_max_salary_for_report)

    filtered = [v for v in vacancies_db if n1 <= v['min_salary']
<= n2]
    if filtered:
        sorted_vac = quick_sort(filtered, key_func=key_report3)
        show_vacancies(sorted_vac, f"ОТЧЕТ 3: ОКЛАД ОТ {n1} ДО
{n2} РУБ.")
    else:
        print(f"Нет вакансий с окладом от {n1} до {n2} рублей")

#ГЛАВНОЕ МЕНЮ

def main_menu():
    """Главное меню программы"""
    options = [
        ("Показать все вакансии", lambda:
show_vacancies(vacancies_db, "ВСЕ ВАКАНСИИ")),
        ("Добавить вакансию", add_vacancy),
        ("Изменить вакансию", edit_vacancy),
        ("Удалить вакансию", delete_vacancy),
        ("Отчет 1: Все вакансии", generate_report1),
        ("Отчет 2: Испытательный срок ≥ 2 мес", generate_report2),
        ("Отчет 3: По диапазону зарплат", generate_report3),
        ("Сохранить базу", save_vacancies),
        ("Выход", None)
    ]

    while True:
        print("\n" + "="*60)
        print(f"КАДРОВОЕ АГЕНТСТВО (вакансий:
{len(vacancies_db)})")
        print("="*60)

        for i, (text, _) in enumerate(options, 1):
            print(f"{i}. {text}")
```

```
        choice = get_input("\nВыбор (1-9): ", check_menu_choice)

        if choice == 9:
            print("\nДо свидания!")
            break

        _, action = options[choice-1]
        if action:
            action()
            if choice != 9:
                input("\nEnter для продолжения...")

#ЗАПУСК ПРОГРАММЫ

if __name__ == "__main__":
    print("\n" + "="*60)
    print("КАДРОВОЕ АГЕНТСТВО")
    print("="*60)

    if load_vacancies():
        main_menu()
    else:
        print("\nСоздайте файл vacancies.txt и перезапустите
программу.")
```