

МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»
(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова)

Факультет Среднего профессионального образования

Дисциплина Основы алгоритмизации и программирования
наименование дисциплины

ЛАБОРАТОРНАЯ РАБОТА №7
номер (при наличии)

Структуры

при наличии указать тему лабораторной работы и (или) номер варианта

ОБУЧАЮЩИЙСЯ

группы 09С51

Куманов Д.В

подпись

фамилия и инициалы

дата сдачи

ПРОВЕРИЛ

Шарипова Э.Р.

подпись

фамилия и инициалы

Оценка / балльная оценка

дата проверки

г. Санкт-Петербург
20 26 г.

Задание 1. Учёт животных в приюте

1. Объявите структуру **Животное** со следующими полями:
 - кличка;
 - возраст;
 - вес в кг;
 - вид.
2. Объявите структуру **Запись**, содержащую:
 - идентификатор записи;
 - данные о животном;
 - дата поступления.
3. Создайте массив из трёх записей типа **Запись** и инициализируйте его произвольными данными.
4. Реализуйте вывод всех записей: для каждой записи выведите идентификатор, кличку, вид, возраст и дату поступления.
5. Выполните поиск и вывод всех животных старше 5 лет.

Код программы:

```
#include <stdio.h>

struct animal {
    char name[50];
    int age;
    float weight;
    char species[50];
};

struct record {
    int id;
    struct animal info;
    char arrival_date[20];
};

int main() {
    // инициализация массива записей
    struct record shelter[3] = {
        {1, {"Барсик", 6, 4.5, "кот"}, "01.10.2023"},
        {2, {"Вова", 3, 15.2, "собака"}, "15.11.2023"},
        {3, {"Буба", 8, 3.1, "кошка"}, "20.12.2023"}
    };

    printf("список всех записей:\n");
    for (int i = 0; i < 3; i++) {
```

```

        printf("id: %d | кличка: %s | вид: %s | возраст: %d | дата: %s\n",
               shelter[i].id,
               shelter[i].info.name,
               shelter[i].info.species,
               shelter[i].info.age,
               shelter[i].arrival_date);
    }

    printf("животные старше 5 лет:\n");
    for (int i = 0; i < 3; i++) {
        if (shelter[i].info.age > 5) {
            printf("кличка: %s, возраст: %d\n",
                   shelter[i].info.name,
                   shelter[i].info.age);
        }
    }

    return 0;
}

```

Результат работы программы прикреплен на рисунке 1.

Your Output

```

список всех записей:
id: 1 | кличка: барсик | вид: кот | возраст: 6 | дата: 01.10.2023
id: 2 | кличка: вова | вид: собака | возраст: 3 | дата: 15.11.2023
id: 3 | кличка: буба | вид: кошка | возраст: 8 | дата: 20.12.2023
животные старше 5 лет:
кличка: барсик, возраст: 6
кличка: буба, возраст: 8

```

Рисунок 1 – Результат работы программы

Задание 2. Ветеринарный осмотр

Объявите структуру **Осмотр** со следующими полями:

- дата осмотра;
 - температура тела;
 - диагноз;
 - признак здоровья.
1. Используйте ранее объявленную структуру **Animal** (см. задание 1).
 2. Объявите структуру **Мед.Карта**, содержащую:
 - animal (тип **Животное**);
 - last_checkup (тип **Осмотр**).

3. Создайте две медицинские карты и инициализируйте их данными.
4. Для каждой карты выведите: кличку животного, дату осмотра, температуру и заключение («здоров» / «требуется лечение») на основе поля **is_healthy**.
5. Выполните сравнение температур двух животных и выведите информацию о том, у кого она выше.

Код программы:

```
#include <stdio.h>

struct animal {
    char name[50];
    int age;
    float weight;
    char species[50];
};

// называем структуру осмотра по-другому
struct checkup {
    char date[20];
    float temperature;
    char diagnosis[100];
    int is_healthy;
};

// теперь мед.карта объединяет обе структуры
struct medcard {
    struct animal pet;
    struct checkup last_checkup;
};

int main() {
    // // инициализация первой карты
    struct medcard card1 = {
        {"шарик", 4, 12.0, "собака"},
        {"10.02.2024", 38.5, "норма", 1}
    };

    // // инициализация второй карты
    struct medcard card2 = {
        {"мурчало", 2, 3.5, "кошка"},
        {"12.02.2024", 39.8, "воспаление", 0}
    };

    struct medcard cards[2] = {card1, card2};

    printf("данные осмотра:\n");
    for (int i = 0; i < 2; i++) {
        printf("кличка: %s, дата: %s, темп.: %.1f, статус: %s\n",
            cards[i].pet.name,
            cards[i].last_checkup.date,
            cards[i].last_checkup.temperature,
            cards[i].last_checkup.is_healthy ? "здоров" : "Лечите
его");
    }
```

```

        printf("\nсравнение температуры:\n");
        if (card1.last_checkup.temperature > card2.last_checkup.temperature)
        {
            printf("у %s температура выше\n", card1.pet.name);
        } else if (card2.last_checkup.temperature >
card1.last_checkup.temperature) {
            printf("у %s температура выше\n", card2.pet.name);
        } else {
            printf("температура одинаковая\n");
        }

        return 0;
    }

```

Результат работы программы прикреплен на рисунке 2.

Your Output

```

данное осмотра:
кличка: шарик, дата: 10.02.2024, темп.: 38.5, статус: здоров
кличка: мурчало, дата: 12.02.2024, темп.: 39.8, статус: Лечите его

сравнение температуры:
у мурчало температура выше

```

Рисунок 2 – Результат работы программы