

Сериализация и десериализация данных

Т.И. Комаров

НИЯУ МИФИ

2023

Определение

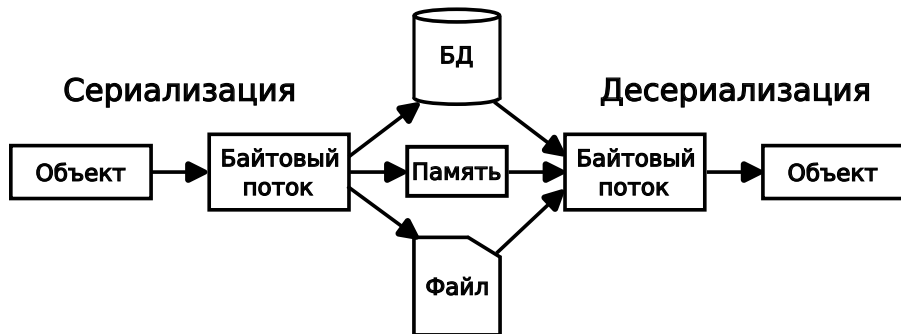
Сериализация — это процесс преобразования состояния объекта (или набора объектов) в форму, пригодную для сохранения или передачи, т. е. в последовательность байт.

Процессом, обратным по отношению к сериализации, является десериализация.

Цели применения сериализации:

- Сохранение объектов в файл
- Сохранение объектов в базу данных
- Передача объектов по сети

Сериализация и десериализация II



Форматы сериализации

Существует множество форматов сериализации:

Текстовые:

- XML
- JSON
- YAML
- TOML
- Собственные
- ...

Бинарные:

- Protobuf
- BSON
- MessagePack
- ASN.1
- Собственные
- ...

Важно

В рамках лабораторного практикума необходимо использовать собственные простые форматы сериализации данных.

В рамках реальных проектов рекомендуется использовать общепринятые форматы сериализации данных.

Пример сериализации I

Примеры

Пример набора атрибутов для описания человека:

- Имя — строка произвольной длины
- Номер паспорта — строка фиксированной длины формата «XXXX XXXXXX», где X — цифра
- Рост — целое беззнаковое число
- Вес — целое беззнаковое число

Реализация соответствующей структуры на языке C:

```
typedef struct Person {  
    char *name; char id[12];  
    int height; int weight;  
} Person;
```

Пример сериализации II

Примеры

JSON (текст):

```
{  
  "name": "Vasya",  
  "id": "1234 567890",  
  "height": 170,  
  "weight": 120  
}
```

XML (текст):

```
<?xml version="1.0"  
  encoding="UTF-8" ?>  
<root>  
  <name>Vasya</name>  
  <id>1234 567890</id>  
  <height>170</height>  
  <weight>120</weight>  
</root>
```

MessagePack (бинарные данные):

```
84 A4 6E 61 6D 65 A5 56 61 73 79 61 A2 69 64 AB  
31 32 33 34 20 35 36 37 38 39 30 A6 68 65 69 67  
68 74 CC AA A6 77 65 69 67 68 74 78
```

```
#include <stdio.h>
```

```
...
```

```
FILE *f = fopen(filename, "w");
```

```
...
```

```
fprintf(f, "%d\n", len);
```

```
for (int i = 0; i < len; ++i) {  
    fprintf(f, "%s\n", arr[i].name);  
    fprintf(f, "%s\n", arr[i].id);  
    fprintf(f, "%d\n", arr[i].height);  
    fprintf(f, "%d\n", arr[i].weight);  
}
```

```
...
```

```
fclose(f);
```

```
#include <stdio.h>
```

```
...
```

```
FILE *f = fopen(filename, "w");
```

```
...
```

```
fwrite(&len, sizeof(len), 1, f);
```

```
for (int i = 0; i < len; ++i) {
```

```
    size_t l = strlen(arr[i].name);
```

```
    fwrite(&l, sizeof(l), 1, f);
```

```
    fwrite(arr[i].name, len, sizeof(char), f);
```

```
    ...
```

```
}
```

```
...
```

```
fclose(f);
```