

---

## Генерация случайных данных (задача 01)

Требуется разработать две программы `w` и `r`. Программа `w` через аргументы командной строки получает имя файла и натуральное число `N`. Она генерирует `N` случайных экземпляров класса `A` (метод `setRand`) и последовательно записывает их кодирующие последовательности байтов в файл (метод `writeBin`). При этом кодирующие последовательности байтов полей класса записываются в том же порядке, в каком они объявляются в описании класса.

Программа `r` через аргумент командной строки получает имя файла, последовательно считывает (метод `readBin`) кодирующие последовательности байтов экземпляров класса `A` и печатает их текстовое представление (метод `print`) в стандартный поток вывода.

Описание класса `A` должно быть помещено в файл `A.hpp`, реализация его методов должна быть помещена в файл `A.cpp`. Текст программы `w` (`r`) должен быть помещен в файл `w.cpp` (`r.cpp`). Программы должны собираться с помощью утилиты `make`.

```
class A {
public:
    A();
    void setRand();
    void print();
    int readBin(int fd);
    int writeBin(int fd);
private:
    int    x_;
    char   a_[10];
    double y_;
};
```

Поле `x_` принимает значения из промежутка `[100, 200]`, элементы массива `a_` являются строчными латинскими буквами ('`a`' - '`z`'), поле `y_` принимает значения из интервала `(0, 1)`.

---

## Генерация случайных данных (задача 02)

Требуется разработать две программы `w` и `r`. Программа `w` через аргументы командной строки получает имя файла и натуральное число `N`. Она генерирует `N` случайных экземпляров класса `A` (метод `setRand`) и последовательно записывает их кодирующие последовательности байтов в файл (метод `writeBin`). При этом кодирующие последовательности байтов полей класса записываются в том же порядке, в каком они объявляются в описании класса.

Программа `r` через аргумент командной строки получает имя файла, последовательно считывает (метод `readBin`) кодирующие последовательности байтов экземпляров класса `A` и печатает их текстовое представление (метод `print`) в стандартный поток вывода.

Описание класса `A` должно быть помещено в файл `A.hpp`, реализация его методов должна быть помещена в файл `A.cpp`. Текст программы `w` (`r`) должен быть помещен в файл `w.cpp` (`r.cpp`). Программы должны собираться с помощью утилиты `make`.

```
class A {
public:
    A();
    void setRand();
    void print();
    int readBin(int fd);
    int writeBin(int fd);
private:
    int    x_;
    double y_;
    char   a_[10];
};
```

Поле `x_` принимает значения из промежутка `[0, 9]`, поле `y_` принимает значения из интервала `(1, 10)`, элементы массива `a_` являются заглавными латинскими буквами ('`A`' - '`Z`').

---

---

## Генерация случайных данных (задача 03)

Требуется разработать две программы `w` и `r`. Программа `w` через аргументы командной строки получает имя файла и натуральное число `N`. Она генерирует `N` случайных экземпляров класса `A` (метод `setRand`) и последовательно записывает их кодирующие последовательности байтов в файл (метод `writeBin`). При этом кодирующие последовательности байтов полей класса записываются в том же порядке, в каком они объявляются в описании класса.

Программа `r` через аргумент командной строки получает имя файла, последовательно считывает (метод `readBin`) кодирующие последовательности байтов экземпляров класса `A` и печатает их текстовое представление (метод `print`) в стандартный поток вывода.

Описание класса `A` должно быть помещено в файл `A.hpp`, реализация его методов должна быть помещена в файл `A.cpp`. Текст программы `w` (`r`) должен быть помещен в файл `w.cpp` (`r.cpp`). Программы должны собираться с помощью утилиты `make`.

```
class A {
public:
    A();
    void setRand();
    void print();
    int readBin(int fd);
    int writeBin(int fd);
private:
    char    a_[10];
    double  y_;
    int     x_;
};
```

Элементы массива `a_` являются символами цифр ('0' - '9'), поле `y_` принимает значения из интервала (10, 20), поле `x_` принимает значения из промежутка [1000, 5000].

---

## Генерация случайных данных (задача 04)

Требуется разработать две программы `w` и `r`. Программа `w` через аргументы командной строки получает имя файла и натуральное число `N`. Она генерирует `N` случайных экземпляров класса `A` (метод `setRand`) и последовательно записывает их кодирующие последовательности байтов в файл (метод `writeBin`). При этом кодирующие последовательности байтов полей класса записываются в том же порядке, в каком они объявляются в описании класса.

Программа `r` через аргумент командной строки получает имя файла, последовательно считывает (метод `readBin`) кодирующие последовательности байтов экземпляров класса `A` и печатает их текстовое представление (метод `print`) в стандартный поток вывода.

Описание класса `A` должно быть помещено в файл `A.hpp`, реализация его методов должна быть помещена в файл `A.cpp`. Текст программы `w` (`r`) должен быть помещен в файл `w.cpp` (`r.cpp`). Программы должны собираться с помощью утилиты `make`.

```
class A {
public:
    A();
    void setRand();
    void print();
    int readBin(int fd);
    int writeBin(int fd);
private:
    char    c_;
    int     a_[5];
    double  y_;
};
```

Значением поля `c_` является строчная латинская буква ('a' - 'z'), элементы массива `a_` принимают значения из промежутка [0, 9], поле `y_` принимает значения из интервала (0, 10).

---

---

## Генерация случайных данных (задача 05)

Требуется разработать две программы `w` и `r`. Программа `w` через аргументы командной строки получает имя файла и натуральное число `N`. Она генерирует `N` случайных экземпляров класса `A` (метод `setRand`) и последовательно записывает их кодирующие последовательности байтов в файл (метод `writeBin`). При этом кодирующие последовательности байтов полей класса записываются в том же порядке, в каком они объявляются в описании класса.

Программа `r` через аргумент командной строки получает имя файла, последовательно считывает (метод `readBin`) кодирующие последовательности байтов экземпляров класса `A` и печатает их текстовое представление (метод `print`) в стандартный поток вывода.

Описание класса `A` должно быть помещено в файл `A.hpp`, реализация его методов должна быть помещена в файл `A.cpp`. Текст программы `w` (`r`) должен быть помещен в файл `w.cpp` (`r.cpp`). Программы должны собираться с помощью утилиты `make`.

```
class A {
public:
    A();
    void setRand();
    void print();
    int readBin(int fd);
    int writeBin(int fd);
private:
    char    c_;
    double  y_;
    int     a_[5];
};
```

Значением поля `c_` является заглавная латинская буква ('A' - 'Z'), поле `y_` принимает значения из интервала (1, 5), элементы массива `a_` принимают значения из промежутка [10, 99].

---

## Генерация случайных данных (задача 06)

Требуется разработать две программы `w` и `r`. Программа `w` через аргументы командной строки получает имя файла и натуральное число `N`. Она генерирует `N` случайных экземпляров класса `A` (метод `setRand`) и последовательно записывает их кодирующие последовательности байтов в файл (метод `writeBin`). При этом кодирующие последовательности байтов полей класса записываются в том же порядке, в каком они объявляются в описании класса.

Программа `r` через аргумент командной строки получает имя файла, последовательно считывает (метод `readBin`) кодирующие последовательности байтов экземпляров класса `A` и печатает их текстовое представление (метод `print`) в стандартный поток вывода.

Описание класса `A` должно быть помещено в файл `A.hpp`, реализация его методов должна быть помещена в файл `A.cpp`. Текст программы `w` (`r`) должен быть помещен в файл `w.cpp` (`r.cpp`). Программы должны собираться с помощью утилиты `make`.

```
class A {
public:
    A();
    void setRand();
    void print();
    int readBin(int fd);
    int writeBin(int fd);
private:
    int     a_[5];
    double  y_;
    char    c_;
};
```

Элементы массива `a_` принимают значения из промежутка  $[-10, -1]$ , поле `y_` принимает значения из интервала  $(-5, -1)$ , значением поля `c_` является цифра ('0' - '9').

---

---

## Генерация случайных данных (задача 07)

Требуется разработать две программы `w` и `r`. Программа `w` через аргументы командной строки получает имя файла и натуральное число `N`. Она генерирует `N` случайных экземпляров класса `A` (метод `setRand`) и последовательно записывает их кодирующие последовательности байтов в файл (метод `writeBin`). При этом кодирующие последовательности байтов полей класса записываются в том же порядке, в каком они объявляются в описании класса.

Программа `r` через аргумент командной строки получает имя файла, последовательно считывает (метод `readBin`) кодирующие последовательности байтов экземпляров класса `A` и печатает их текстовое представление (метод `print`) в стандартный поток вывода.

Описание класса `A` должно быть помещено в файл `A.hpp`, реализация его методов должна быть помещена в файл `A.cpp`. Текст программы `w` (`r`) должен быть помещен в файл `w.cpp` (`r.cpp`). Программы должны собираться с помощью утилиты `make`.

```
class A {
public:
    A();
    void setRand();
    void print();
    int readBin(int fd);
    int writeBin(int fd);
private:
    char a_[10];
    int b_[5];
    char c_;
};
```

Элементы массива `a_` являются строчными латинскими буквами ('a' - 'z'), элементы массива `b_` принимают значения из промежутка [1, 5], значением поля `c_` является цифра ('0' - '9').

---

## Генерация случайных данных (задача 08)

Требуется разработать две программы `w` и `r`. Программа `w` через аргументы командной строки получает имя файла и натуральное число `N`. Она генерирует `N` случайных экземпляров класса `A` (метод `setRand`) и последовательно записывает их кодирующие последовательности байтов в файл (метод `writeBin`). При этом кодирующие последовательности байтов полей класса записываются в том же порядке, в каком они объявляются в описании класса.

Программа `r` через аргумент командной строки получает имя файла, последовательно считывает (метод `readBin`) кодирующие последовательности байтов экземпляров класса `A` и печатает их текстовое представление (метод `print`) в стандартный поток вывода.

Описание класса `A` должно быть помещено в файл `A.hpp`, реализация его методов должна быть помещена в файл `A.cpp`. Текст программы `w` (`r`) должен быть помещен в файл `w.cpp` (`r.cpp`). Программы должны собираться с помощью утилиты `make`.

```
class A {
public:
    A();
    void setRand();
    void print();
    int readBin(int fd);
    int writeBin(int fd);
private:
    char a_[5];
    double y_;
    char b_[5];
};
```

Элементы массива `a_` являются заглавными латинскими буквами ('A' - 'Z'), поле `y_` принимает значения из интервала (1, 10), элементы массива `b_` являются цифрами ('0' - '9').

---

---

## Генерация случайных данных (задача 09)

Требуется разработать две программы `w` и `r`. Программа `w` через аргументы командной строки получает имя файла и натуральное число `N`. Она генерирует `N` случайных экземпляров класса `A` (метод `setRand`) и последовательно записывает их кодирующие последовательности байтов в файл (метод `writeBin`). При этом кодирующие последовательности байтов полей класса записываются в том же порядке, в каком они объявляются в описании класса.

Программа `r` через аргумент командной строки получает имя файла, последовательно считывает (метод `readBin`) кодирующие последовательности байтов экземпляров класса `A` и печатает их текстовое представление (метод `print`) в стандартный поток вывода.

Описание класса `A` должно быть помещено в файл `A.hpp`, реализация его методов должна быть помещена в файл `A.cpp`. Текст программы `w` (`r`) должен быть помещен в файл `w.cpp` (`r.cpp`). Программы должны собираться с помощью утилиты `make`.

```
class A {
public:
    A();
    void setRand();
    void print();
    int readBin(int fd);
    int writeBin(int fd);
private:
    char a_[10];
    int x_;
    int y_;
};
```

Элементы массива `a_` являются латинскими буквами ('a' - 'z', 'A' - 'Z'), поле `x_` принимает значения из промежутка [1,9], поле `y_` принимает значения из промежутка [101,109].

---

## Генерация случайных данных (задача 10)

Требуется разработать две программы `w` и `r`. Программа `w` через аргументы командной строки получает имя файла и натуральное число `N`. Она генерирует `N` случайных экземпляров класса `A` (метод `setRand`) и последовательно записывает их кодирующие последовательности байтов в файл (метод `writeBin`). При этом кодирующие последовательности байтов полей класса записываются в том же порядке, в каком они объявляются в описании класса.

Программа `r` через аргумент командной строки получает имя файла, последовательно считывает (метод `readBin`) кодирующие последовательности байтов экземпляров класса `A` и печатает их текстовое представление (метод `print`) в стандартный поток вывода.

Описание класса `A` должно быть помещено в файл `A.hpp`, реализация его методов должна быть помещена в файл `A.cpp`. Текст программы `w` (`r`) должен быть помещен в файл `w.cpp` (`r.cpp`). Программы должны собираться с помощью утилиты `make`.

```
class A {
public:
    A();
    void setRand();
    void print();
    int readBin(int fd);
    int writeBin(int fd);
private:
    int x_;
    int y_;
    char a_[10];
};
```

Поле `x_` принимает значения из промежутка [1001,1009], поле `y_` принимает значения из промежутка [101,109], элементы массива `a_` являются строчными латинскими буквами или цифрами ('a' - 'z', '0' - '9').

---

---

## Генерация случайных данных (задача 11)

Требуется разработать две программы `w` и `r`. Программа `w` через аргументы командной строки получает имя файла и натуральное число `N`. Она генерирует `N` случайных экземпляров класса `A` (метод `setRand`) и последовательно записывает их кодирующие последовательности байтов в файл (метод `writeBin`). При этом кодирующие последовательности байтов полей класса записываются в том же порядке, в каком они объявляются в описании класса.

Программа `r` через аргумент командной строки получает имя файла, последовательно считывает (метод `readBin`) кодирующие последовательности байтов экземпляров класса `A` и печатает их текстовое представление (метод `print`) в стандартный поток вывода.

Описание класса `A` должно быть помещено в файл `A.hpp`, реализация его методов должна быть помещена в файл `A.cpp`. Текст программы `w` (`r`) должен быть помещен в файл `w.cpp` (`r.cpp`). Программы должны собираться с помощью утилиты `make`.

```
class A {
public:
    A();
    void setRand();
    void print();
    int readBin(int fd);
    int writeBin(int fd);
private:
    int    a_[5];
    char   c_;
    double y_;
};
```

Элементы массива `a_` принимают значения из множества  $[1, 9] \cup [101, 109]$ , значением поля `c_` является цифра ('0' - '9'), поле `y_` принимает значения из интервала (1, 5).

---

## Генерация случайных данных (задача 12)

Требуется разработать две программы `w` и `r`. Программа `w` через аргументы командной строки получает имя файла и натуральное число `N`. Она генерирует `N` случайных экземпляров класса `A` (метод `setRand`) и последовательно записывает их кодирующие последовательности байтов в файл (метод `writeBin`). При этом кодирующие последовательности байтов полей класса записываются в том же порядке, в каком они объявляются в описании класса.

Программа `r` через аргумент командной строки получает имя файла, последовательно считывает (метод `readBin`) кодирующие последовательности байтов экземпляров класса `A` и печатает их текстовое представление (метод `print`) в стандартный поток вывода.

Описание класса `A` должно быть помещено в файл `A.hpp`, реализация его методов должна быть помещена в файл `A.cpp`. Текст программы `w` (`r`) должен быть помещен в файл `w.cpp` (`r.cpp`). Программы должны собираться с помощью утилиты `make`.

```
class A {
public:
    A();
    void setRand();
    void print();
    int readBin(int fd);
    int writeBin(int fd);
private:
    double y_;
    int    a_[5];
    char   c_;
};
```

Поле `y_` принимает значения из интервала (10, 50), элементы массива `a_` принимают значения из множества  $[11, 19] \cup [1001, 1009]$ , значением поля `c_` является заглавная латинская буква ('A' - 'Z').

---