

Семестр 4 (2019), занятие 2

Случайные данные

Последовательность псевдослучайных целых чисел.

```
int i;
...
srand(0);
for(i = 0; i < 10; i++)
    printf("%d\n", rand());
```

Результат.

```
1804289383
846930886
1681692777
1714636915
1957747793
424238335
719885386
1649760492
596516649
1189641421
```

Последовательность псевдослучайных целых чисел в диапазоне $[0, 4]$.

```
int i;
...
srand(0);
for(i = 0; i < 50; i++)
    printf("%d ", rand() % 5);
```

Результат.

```
3 1 2 0 3 0 1 2 4 1 2 2 0 4 3 1 0 1 2 ...
```

В общем случае, генерировать последовательность псевдослучайных целых чисел в диапазоне $[0, N - 1]$ можно по формуле

```
rand() % N
```

Однако, более корректно использовать следующую формулу

```
(int)( N * rand() / (RAND_MAX + 1.0))
```

Псевдослучайная последовательность строчных латинских букв.

```
int i;
...
srand(0);
for(i = 0; i < 50; i++)
    printf("%c ", 'a' + rand() % 26);
```

Результат.

```
n w l r b b m q b h c d a r z o w k k ...
```

Псевдослучайная последовательность заглавных латинских букв.

```
int i;
...
srand(0);
for(i = 0; i < 50; i++)
    printf("%c ", 'A' + rand() % 26);
```

Результат.

```
N W L R B B M Q B H C D A R Z O W K K ...
```

Псевдослучайная последовательность символов цифр.

```
int i;
...
srand(0);
for(i = 0; i < 50; i++)
    printf("%c ", '0' + rand() % 10);
```

Результат.

```
3 6 7 5 3 5 6 2 9 1 2 7 0 9 3 6 0 6 2 ...
```

Псевдослучайная последовательность вещественных чисел в диапазоне $[0, 1]$.

```
int i;
...
srand(0);
for(i = 0; i < 50; i++)
    printf("%f\n", rand() / (double) RAND_MAX);
```

Результат.

```
0.840188
0.394383
0.783099
0.798440
0.911647
0.197551
0.335223
0.768230
0.277775
0.553970
```

Псевдослучайная последовательность строк длины меньше 10.

```
1 int i, j, l;
2 char s[1024];
3 ...
4 srand(0);
5 for(i = 0; i < 20; i++) {
6     l = rand() % 10;
7     for(j = 0; j < l; j++)
8         s[j] = 'a' + rand() % 26;
9     s[l] = 0;
10    printf("%d %s\n", l, s);
11 }
```

Результат.

```
3 wlr
3 bmq
9 hcdarzowk
2 yh
8 ddqscdxr
9 mowfrxsjy
9 ldbefsarc
3 yne
...
```

Псевдослучайная последовательность строк длины меньше 10.

```
1 int i, j, l;
2 char *p;
3 ...
```

```

4  srand(0);
5  for(i = 0; i < 20; i++) {
6      l = rand() % 10;
7
8      if(!(p = malloc(l + 1)))
9          ... // Error.
10
11     for(j = 0; j < l; j++)
12         p[j] = 'a' + rand() % 26;
13     p[l] = 0;
14     printf("%d %s\n", l, p);
15
16     free(p);
17 }

```

Результат.

```

3 wlr
3 bmq
9 hcdarzewk
2 yh
8 ddqscdxr
9 mowfrxsjy
9 ldbefsarc
3 yne
...

```

Псевдослучайная последовательность строк из заданного набора.

```

1  int i, j;
2  char *a[] = {"A", "AB", "ABC", "ABCD"};
3  ...
4  srand(0);
5  for(i = 0; i < 20; i++) {
6      j = rand() % 4;
7      printf("%d %s\n", j, a[j]);
8  }

```

Результат.

```

3 ABCD
2 ABC
1 AB
3 ABCD
1 AB
3 ABCD
2 ABC
0 A
1 AB
1 AB
2 ABC
3 ABCD
2 ABC
3 ABCD
2 ABC
0 A
2 ABC
0 A
0 A

```

Псевдослучайная последовательность копий строк из заданного набора.

```

1  int i, j, l;
2  char s[1024], *p;
3      *a[] = {"A", "AB", "ABC", "ABCD"};
4  ...
5  srand(0);
6  for(i = 0; i < 20; i++) {
7      l = rand() % 10;
8
9      if(!(p = malloc(l + 1)))
10         ... // Error.

```

```

11
12     for(j = 0; j < l; j++)
13         p[j] = 'a' + rand() % 26;
14     p[l] = 0;
15     printf("%d %s\n", l, p);
16
17     free(p);
18 }

```

Класс Date

```

class Date {
public:
    Date();
    Date(int y, int m, int d);

    const char *str() const;
    void setRand();

    bool operator< (const Date &d) const;

private:
    int y_,
        m_,
        d_;
    char s_[11];

    void set(int y, int m, int d);
};

#include <stdexcept>
...
void Date::set(int y, int m, int d) {
    bool c = (1900 <= y && y <= 2100)
        || (1 <= m && m <= 12)
        || (1 <= d && d <= 30);

    if(!c)
        throw std::invalid_argument("...");

    y_ = y;
    m_ = m;
    d_ = d;

    snprintf(s_, 11, "%4d-%02d-%02d",
        y_, m_, d_);
}

Date::Date() {
    set(1900, 1, 1);
}

Date::Date(int y, int m, int d) {
    set(y, m, d);
}

try {
    Date(1899, 10, 10);
}
catch(const std::exception &e) {
    fprintf(stderr, "%s\n", e.what());
}

const char *Date::str() {
    return s_;
}

void Date::setRand() {
    int y = 1900 + rand() % 200,
        m = 1 + rand() % 12,
        d = 1 + rand() % 30;

```

```

        set(y, m, d);
    }

    bool Date::operator< (const Date &d) const {
        return y_ < d.y_
            || (y_ == d.y_ && m_ < d.m_)
            || (y_ == d.y_ && m_ == d.m_ && d_ < d.d_);
    }

    Date d;
    printf("%s\n\n", d.str());

    srand(0);
    for(int i = 0; i < 50; i++) {
        d.setRand();
        printf("%s <(%c) 2005-10-16 \n",
            d.str(),
            d < Date(2005, 10, 16) ? '+' : '-');
    }

```

Результат.

```

2083-11-28 <(-) 2005-10-16
2015-06-26 <(-) 2005-10-16
2086-01-10 <(-) 2005-10-16
1921-03-08 <(+) 2005-10-16
1990-08-24 <(+) 2005-10-16
2026-01-07 <(-) 2005-10-16
2072-05-12 <(-) 2005-10-16
2068-04-10 <(-) 2005-10-16
2082-03-03 <(-) 2005-10-16
2023-08-26 <(-) 2005-10-16
2029-07-13 <(-) 2005-10-16
1958-06-28 <(+) 2005-10-16
2093-05-02 <(-) 2005-10-16
1942-10-04 <(+) 2005-10-16
1921-12-15 <(+) 2005-10-16
2037-03-15 <(-) 2005-10-16
...

```

Стандартный вектор

```

#include<vector>
...
std::vector<Date> v(5);

for(int j = 0; j < 5; j++) {
    Date d;
    d.setRand();
    v.push_back(d);
}

for(size_t i = 0; i < v.size(); i++)
    printf("%s\n", v[i].str());

```

Результат.

```

1900-01-01
1900-01-01
1900-01-01
1900-01-01
1900-01-01
2083-11-28
2015-06-26
2086-01-10
1921-03-08
1990-08-24

int x = 15;
double y = .5;
const char *s = "abcd";
size_t l;

std::vector<char> v;

```

```

l = v.size();
v.resize(l + sizeof x);
memcpy(&v[0] + l, &x, sizeof x);

l = v.size();
v.resize(l + sizeof y);
memcpy(&v[0] + l, &y, sizeof y);

l = v.size();
v.resize(l + 5);
memcpy(&v[0] + l, s, 5);

int tx;
double ty;
const char *ts;

l = 0;
memcpy(&tx, &v[0] + l, sizeof tx);

l += sizeof tx;
memcpy(&ty, &v[0] + l, sizeof ty);

l += sizeof ty;
ts = &v[0] + l;

printf("%d %f %s\n", tx, ty, ts);

```

Результат.

```
15 0.500000 abcd
```

Класс Student1

Объявление класса.

```

class Student1 {
public:
    ...
    void setRand();

    int readBin(int fd);
    int writeBin(int fd);

    int readTxt(const char *str);
    void writeTxt(FILE *fout);

private:
    char name_[64];
    int group_;
    double rating_;
    char phone_[14];
};

```

Инициализация случайными данными.

```

1 void Student1::setRand() {
2     // Random Name.
3     int j, l = 2 + rand() % 17;
4     name_[0] = 'A' + rand() % 26;
5     for(j = 1; j < l; j++)
6         name_[j] = 'a' + rand() % 26;
7     name_[l] = '\0';
8
9     // Random Group.
10    group_ = (1 + rand() % 5) * 100
11            + 1 + rand() % 9;
12
13    // Random Rating.
14    double r = 2. * (rand() / (double)RAND_MAX)
15            + 3.;
16
17    rating_ = round(r * 10.) / 10.;
18
19    // Random Phone.
20    int p1 = 1 + rand() % 999,

```

```

21         p2 = 1 + rand() % 999,
22         p3 = 1 + rand() % 99,
23         p4 = 1 + rand() % 99;
24     snprintf(phone_, 14, "%03d-%03d-%02d-%02d",
25              p1, p2, p3, p4);
26 }

```

Печать текстового представления.

```

1 void Student1::writeTxt(FILE *fout) {
2     fprintf(fout, "%-20s %03d %3.1f %s\n",
3             name_, group_, rating_, phone_);
4 }

```

Запись в бинарный файл.

```

1 int Student1::writeBin(int fd) {
2     ssize_t r;
3
4     r = write(fd, name_, sizeof name_);
5     if(r != sizeof name_)
6         return -1;
7
8     r = write(fd, &group_, sizeof group_);
9     if(r != sizeof group_)
10        return -1;
11
12    r = write(fd, &rating_, sizeof rating_);
13    if(r != sizeof rating_)
14        return -1;
15
16    r = write(fd, phone_, sizeof phone_);
17    if(r != sizeof phone_)
18        return -1;
19
20    return 0;
21 }

```

Чтение из бинарного файла.

```

1 int Student1::readBin(int fd) {
2     ssize_t r;
3
4     r = read(fd, name_, sizeof name_);
5     if(r == 0)
6         return 0;
7     if(r != sizeof name_)
8         return -1;
9

```

```

10    r = read(fd, &group_, sizeof group_);
11    if(r != sizeof group_)
12        return -1;
13
14    r = read(fd, &rating_, sizeof rating_);
15    if(r != sizeof rating_)
16        return -1;
17
18    r = read(fd, phone_, sizeof phone_);
19    if(r != sizeof phone_)
20        return -1;
21
22    return 1;
23 }

```

Запись случайных записей в файл.

```

1 for(int i = 0; i < n; i++) {
2     Student1 s;
3     s.setRand();
4     s.writeBin(fd);
5 }

```

Чтение всех записей из файла и их печать на экран.

```

1 for(;;) {
2     Student1 s;
3
4     int r = s.readBin(fd);
5     if(r == 0 || r == -1)
6         break;
7
8     s.writeTxt(stdout);
9 }

```

Результат.

Wlrbbmqbheda	506	4.8	088-352-80-56
Idd	309	3.3	023-389-11-33
Owfrxsjybldbefsar	103	3.2	281-675-63-73
Gg	207	4.9	501-493-84-54
Llnmpapqfwkh	108	3.3	340-018-15-76
Nwnkue	402	4.8	719-037-94-21
Uqcljjivswm	203	3.4	245-016-33-20
Mvtr	303	4.4	390-442-86-22
Nfwzqfjmafadrrwso	105	4.5	745-714-87-38
Hffbsaqxwpqca	508	3.0	117-394-82-36