

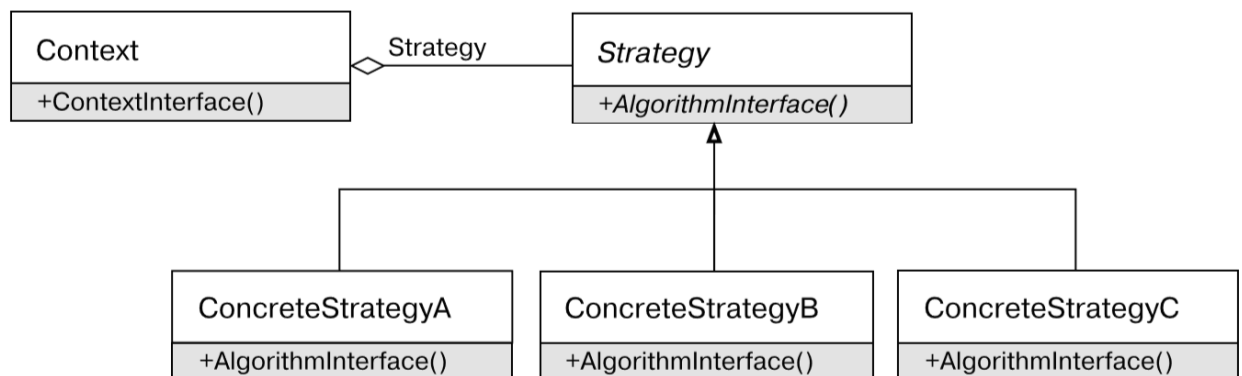
Лабораторная работа 6

Стрекнев Д. СКБ172

Задание:

Используя паттерн Strategy, реализовать генерацию массива случайных чисел (равномерное распределение) не менее, чем 3-мя разными алгоритмами. Параметры генерации: границы отрезка и объем выборки.

Паттерн Strategy:



Код:

```
#include <iostream>
#include <random>
#include <vector>
using namespace std;

// Иерархия классов, определяющая алгоритмы генерации чисел
class Generation {
public:
    vector<int> s;
    virtual void generate(int a, int b, int n) = 0;
    void print(){
        for (int i=0;i<s.size();i++){
            cout << s[i] << " ";
        }
        cout << endl;
    }
    virtual ~Generation(){};
};

class Linear_Congruential_Generation: public Generation {
public:
    minstd_rand0 x;
    double fraction = 1.0 / (static_cast<double>(x.max()) +
1.0);
    void generate(int a, int b, int n){
        for (int i=0;i<n;i++){
            int number = static_cast<int>(x() * fraction * (b -
a + 1) + a);
            s.push_back(number);
        }
    }
};

class Mersenne_Twister_Generation: public Generation {
public:
    mt19937 x;
    double fraction = 1.0 / (static_cast<double>(x.max()) +
1.0);
    void generate(int a, int b, int n){
        for (int i=0;i<n;i++){
            int number = static_cast<int>(x() * fraction * (b -
a + 1) + a);
```

```

        s.push_back(number);
    }
};

```

```

class Subtract_With_Carry_Generation: public Generation {
public:
    ranlux48_base x;
    double fraction = 1.0 / (static_cast<double>(x.max()) +
1.0);
    void generate(int a, int b, int n){
        for (int i=0;i<n;i++){
            int number = static_cast<int>(x() * fraction * (b -
a + 1) + a);
            s.push_back(number);
        }
    }
};

```

```

// Класс для использования
class Genarator {
public:
    Genarator(Generation* gen): g(gen){}
    ~Genarator(){delete g;}
    void generate(int a, int b, int n){
        g->generate(a,b,n);
        g->print();
    }
private:
    Generation* g;
};

```

```

int main() {
    Genarator* gen1 = new Genarator(new
Linear_Congruential_Generation);
    Genarator* gen2 = new Genarator(new
Mersenne_Twister_Generation);
    Genarator* gen3 = new Genarator(new
Subtract_With_Carry_Generation);

    cout << "Linear_Congruential_Generation:" << endl;
    gen1->generate(100, 1000, 10);

    cout << endl << "Mersenne_Twister_Generation:" << endl;
    gen2->generate(100, 1000, 10);
}

```

```
    cout << endl << "Subtract_With_Carry_Generation:" << endl;
    gen3->generate(100, 1000, 10);

    delete gen1;
    delete gen2;
    delete gen3;
}
```

Результат:

```
Linear_Congruential_Generation:
100 218 780 513 580 297 142 711 712 942

Mersenne_Twister_Generation:
834 222 916 852 214 972 922 299 669 377

Subtract_With_Carry_Generation:
175 191 986 519 370 766 330 876 923 277
Program ended with exit code: 0
```