Бараненков Сергей 18 ПИ - 2

1
```
Class Train {
private:
    int id
    list <string> stations
public:
    explicit Train (int id, list<string>stations): id(id),
        stations (std::move (stations)) { };

    Train (const Train & copy) {
        if (this != &copy) {
            stations.clear();
            stations = copy. stations;
            id = copy. id;
        }
    }

    string & operator [ ] (unsigned int n) {
        auto it = stations. begin();
        std: advance (it, n);
        if (it == stations. end() ) {
            throw   invalid_argument (,,Выход за гран. места")
        }
        return *it

    void   remove (const string && to_remove) {
        Stations. remove (to_remove);
    }
```

(2)
```
using namespace std;
set <int> v = { ... }
```
a)
```
for (auto it = v.rbegin(); it != v.rend(); it++){
    cout << *it << endl;
}
```
б)
```
for (auto it = v.begin(); it != v.end; it++){
    if (*it % 2 == 1) {
        cout << *it << endl;
    }
```
в)
```
for_each (v.begin(), v.end(), [] (int x){
    if (*it % 2 == 1) { cout << *it }
}
```
```
mutex m;
```
(3)
```
int main () {
    string s;
    thread t1 (addIntToString, ref(s));
    thread t2 (addCharToString, ref(s));
    t1.join();
    t2.join();
    cout << s;
}
const char chars [] = {"ABC..."}
const ints[] = { 0, 1, 2 .. }
char genChar(){ return chars [random()%26];
int genInt () { return ints [random() %10];
```

```
void add Int To String (string & s){
  m.lock();
  for(int i=0; i<5; i++){
    S+= to_string(gen Int());
  }
  m.unlock;
}

void add Char To string(string & s){
  m.lock()
  for(int i=0; i<5; i++){
    S+= gen Char();
  }
  m.unlock;
}
```