```
#include <iostream>
#include <vector>
#include <cmath>
#include <sstream>
#include <iomanip>
#include <limits>

#include "milling.h"

using namespace milling;

const int kNDay = 100000;

Logger::Logger() {
  log_file_.open("log.txt", std::ios_base::app);
}

Logger::~Logger() {
  log_file_.close();
}

void Logger::Log(std::string log) {
  log_file_ << log;
  log_file_ << std::endl;
}

template <class T>
void EventModel<T>::SetCumProb() {
  float cum;

  for(int i = 0; i < n_options_; i++) {
    cum = 0;
    for(int j = 0; j <= i; j++) {
      cum += probs_[j];
    }
    cum_prob_[i] = cum;
  }
}

template <class T>
void EventModel<T>::SetCumSum() {
  for(int i = 0; i < n_options_; i++) {
    cum_sum_[i] = std::pow(10, n_decimal_) * cum_prob_[i];
  }
}

template <class T>
EventModel<T>::EventModel(int n_decimal, std::vector<T> options,
std::vector<float> probs) {
  options_ = options;
  probs_ = probs;
  n_decimal_ = n_decimal;
  n_options_ = options_.size();
  cum_prob_.resize(n_options_);
  cum_sum_.resize(n_options_);
  SetCumProb();
  SetCumSum();
}

template <class T>
```

```
T EventModel<T>::GetEvent() {
  int r = std::rand();
  int range = std::pow(10, n_decimal_);
  r = r % range;

  if(r == 0) return options_.back();

  for(int i = 0; i < n_options_; i++) {
    if(r <= cum_sum_[i]) return options_[i];
  }

  throw (r);
}

Simulator::Simulator(EventModel<int>& life_model, EventModel<int>&
delay_model)
  : life_model_(life_model), delay_model_(delay_model) {total_delay_ =
total_life_ = 0;}

int Simulator::GetDelay() {
  return delay_model_.GetEvent();
}

int Simulator::GetLife() {
  return life_model_.GetEvent();
}

void OnDemandSimulator::StepSimulate(std::vector<int>& day) {
  int l, d;
  for(int i = 0; i < 3; i++) {
    l = GetLife();
    d = GetDelay();
    day[2*i] = l;
    day[2*i + 1] = d;
  }
}

void BroadcastSimulator::StepSimulate(std::vector<int>& day) {
  int l;
  int min = std::numeric_limits<int>::max();

  for(int i = 0; i < 3; i++) {
    l = GetLife();
    day[i] = l;

    if(min > l) min = l;
  }

  day[3] = min;
  day[4] = GetDelay();
}

void Simulator::UpdateTotals(int l, int d) {
  total_delay_ += d;
  total_life_ += l;
}

void OnDemandSimulator::UpdateTotals(std::vector<int>& day) {
  for(int i = 0; i < 3; i++)
    Simulator::UpdateTotals(day[2*i], day[2*i + 1]);
```

```
}

void BroadcastSimulator::UpdateTotals(std::vector<int>& day) {
  Simulator::UpdateTotals(day[3] * 3, day[4]);
}

void Simulator::Log(std::string s) {
  logger_.Log(s);
}

void Simulator::SetCosts(int n_cols) {
  cost_bearings_ = 3 * kNDay * 32;
  cost_delay_ = total_delay_ * 10;
  cost_downtime_ = n_cols == 6 ? 3 * kNDay * 20 * 10 : kNDay * 40 * 10;
  cost_repair_ = n_cols == 6 ? 3 * kNDay * 20 * 30 / 60 : kNDay * 40 * 30
/ 60;
  total_cost_ = cost_bearings_ + cost_delay_ + cost_downtime_ +
cost_repair_;
  total_cost_per_10k_hour = total_cost_ / ((float)total_life_ / 10000);
}

void Simulator::LogMetrics() {
  std::stringstream metrics;
  metrics << "Cost of bearings: " << cost_bearings_ << std::endl
          << "Cost of delay time: " << cost_delay_ << std::endl
          << "Cost of downtime during repair: " << cost_downtime_ <<
std::endl
          << "Cost of repair person: " << cost_repair_ << std::endl
          << "Total cost: " << total_cost_ << std::endl
          << "Total life of bearings: " << total_life_ << std::endl
          << "Total cost per 10k hour: " << total_cost_per_10k_hour <<
std::endl
     ;

  std::string metrics_string = metrics.str();
  Log(metrics_string);
}

void Simulator::RunSimulation(int n_cols) {
  if(kNDay == 0) return;
  std::stringstream initial_log;
  std::string title = n_cols == 6 ? "###On Demand Simulation###" :
"###Broadcast Simulation###";
  initial_log << title << std::endl
        << "Days: " << kNDay << std::endl
    ;
  Log(initial_log.str());
  std::vector<int> day(n_cols, 0);

  for(int i = 0; i < kNDay; i++) {
    StepSimulate(day);
    UpdateTotals(day);
  }

  SetCosts(n_cols);
  LogMetrics();
}

int main() {
```

```
  std::vector<int> life_options {1000, 1100, 1200, 1300, 1400, 1500,
1600, 1700, 1800, 1900};
  std::vector<float> life_probs {0.1, 0.13, 0.25, 0.13, 0.09, 0.12, 0.02,
0.06, 0.05, 0.05};
  std::vector<int> delay_options {5, 10, 15};
  std::vector<float> delay_probs {0.6, 0.3, 0.1};

  EventModel<int> life_model(2, life_options, life_probs), delay_model(1,
delay_options, delay_probs);
  OnDemandSimulator on_demand_simulator(life_model, delay_model);
  BroadcastSimulator broadcast_simulator(life_model, delay_model);

  on_demand_simulator.RunSimulation(6);
  broadcast_simulator.RunSimulation(5);

  return 0;
}
```

```
###On Demand Simulation###
Days: 15

Cost of bearings: 1440
Cost of delay time: 3500
Cost of downtime during repair: 9000
Cost of repair person: 450
Total cost: 14390
Total life of bearings: 62400
Total cost per 10k hour: 2306

###Broadcast Simulation###
Days: 15

Cost of bearings: 1440
Cost of delay time: 1200
Cost of downtime during repair: 6000
Cost of repair person: 300
Total cost: 8940
Total life of bearings: 49500
Total cost per 10k hour: 1806

###On Demand Simulation###
Days: 100

Cost of bearings: 9600
Cost of delay time: 23450
Cost of downtime during repair: 60000
Cost of repair person: 3000
Total cost: 96050
Total life of bearings: 403000
Total cost per 10k hour: 2383

###Broadcast Simulation###
Days: 100

Cost of bearings: 9600
Cost of delay time: 7500
Cost of downtime during repair: 40000
Cost of repair person: 2000
Total cost: 59100
Total life of bearings: 349500
Total cost per 10k hour: 1690

###On Demand Simulation###
Days: 1000

Cost of bearings: 96000
Cost of delay time: 228750
Cost of downtime during repair: 600000
Cost of repair person: 30000
Total cost: 954750
Total life of bearings: 4046100
Total cost per 10k hour: 2359

###Broadcast Simulation###
Days: 1000

Cost of bearings: 96000
Cost of delay time: 75350
```

```
Cost of downtime during repair: 400000
Cost of repair person: 20000
Total cost: 591350
Total life of bearings: 3452400
Total cost per 10k hour: 1712

###On Demand Simulation###
Days: 10000

Cost of bearings: 960000
Cost of delay time: 2248850
Cost of downtime during repair: 6000000
Cost of repair person: 300000
Total cost: 9508850
Total life of bearings: 40186700
Total cost per 10k hour: 2366

###Broadcast Simulation###
Days: 10000

Cost of bearings: 960000
Cost of delay time: 749000
Cost of downtime during repair: 4000000
Cost of repair person: 200000
Total cost: 5909000
Total life of bearings: 34317300
Total cost per 10k hour: 1721

###On Demand Simulation###
Days: 100000

Cost of bearings: 9600000
Cost of delay time: 22490550
Cost of downtime during repair: 60000000
Cost of repair person: 3000000
Total cost: 95090550
Total life of bearings: 401845400
Total cost per 10k hour: 2366

###Broadcast Simulation###
Days: 100000

Cost of bearings: 9600000
Cost of delay time: 7508950
Cost of downtime during repair: 40000000
Cost of repair person: 2000000
Total cost: 59108950
Total life of bearings: 343568100
Total cost per 10k hour: 1720
```