МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по практической работе №3
по дисциплине «Теория принятия решений»
Тема: Бесконечные антагонистические игры

Студентка гр. 7381 Алясова А.Н.

Преподаватель Попова Е.В.

Санкт-Петербург
2021

Цель работы.

Изучение различных инструментальных средств для решения задач поддержки принятия решения, а также овладение навыками принятия решения на основе задач астрономии.

Основные теоретические положения.

Используя соответствующие программы на языке C++ решить задачи принятия решения в астрономии.

Постановка задачи.

1. Скачать файлы по ссылке. Ликвидируйте баги и создайте приложение из интерфейсных и клиентских модулей (понадобятся `APC_Const, APC_Math, APC_PrecNut, APC_Spheric, APC_Sun, APC_Time, APC_VecMat3D, GNU_iomanip, Coco`). Укажите какие изменения произведены в файлах.

2. Провести циклические преобразования координат в зависимости от варианта. Выбрать `system X, format Y, coordinates Z, equinox K, origin L, epoch M`. Узнать координаты точки равноденствия в эпоху `N`. Перейти к `O` координатам. Перейти к `P` координатам. Вернуться к исходной точке. Определить погрешность.

3. Вывести значение данного угла `X` в 5 различных форматах, используемых в приложении. Сделать выводы.

Индивидуализация.

Вариант 11.

X	***Y***	***Z***	***K***	***L***	***M***	***N***	***O***	***P***
e	p	Все по 20, расстояние – половина астрономической единицы	1970.0	g	1990 первого января, 0 часов	2000.0	a	h

Задача №2.

По заданной модифицированной юлианской дате $MJD = 3432.1$ получить эклиптические и экваториальные координаты.

Выполнение работы.

1) Ликвидируем баги и создадим приложение из интерфейсных и клиентских модулей.

В `APC_Math` удалим следующую строчку:

```
os.setf(flags); // restore output stream format flags
```

Рисунок 1

В `APC_PrecNut` и `APC_Spheric` добавим следующие строчки:

```
Mat3D R_x(double RotAngle);
Mat3D R_y(double RotAngle);
Mat3D R_z(double RotAngle);
```

Рисунок 2

В `APC_Sun` добавим следующую строчку:

```
Mat3D R_x(double RotAngle);
```

Рисунок 3

В `GNU_iomanip` удалим следующие строчки:

```
namespace{
ostream& left (ostream& os){os.setf(ios::left ,ios::adjustfield); return os;};
ostream& right(ostream& os){os.setf(ios::right,ios::adjustfield); return os;};
ostream& fixed(ostream& os){os.setf(ios::fixed,ios::floatfield);  return os;};
ostream& showpos  (ostream& os){os.setf(ios::showpos); return os;};
ostream& noshowpos(ostream& os){os.unsetf(ios::showpos); return os;};
}
```

Рисунок 4

В `Coco.cpp` заменим тип возвращаемого значения функции `main` на `int`:

```
int main() {
```

Рисунок 5

Сборку приложения произведём следующим образом:

```
g++ -std=c++11 Coco.cpp APC_VecMat3D.cpp APC_Math.cpp
APC_PrecNut.cpp APC_Spheric.cpp APC_Sun.cpp APC_Time.cpp
```

Рисунок 6

2) Проведём циклические преобразования координат. Результаты представлены на рис. 7–11.

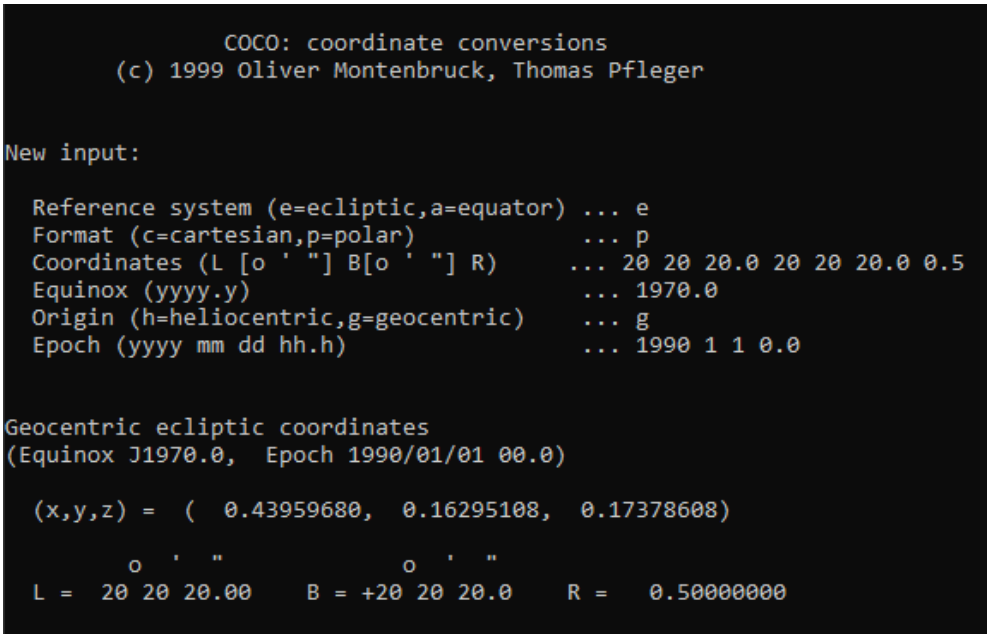

Рисунок 7 – Исходная точка

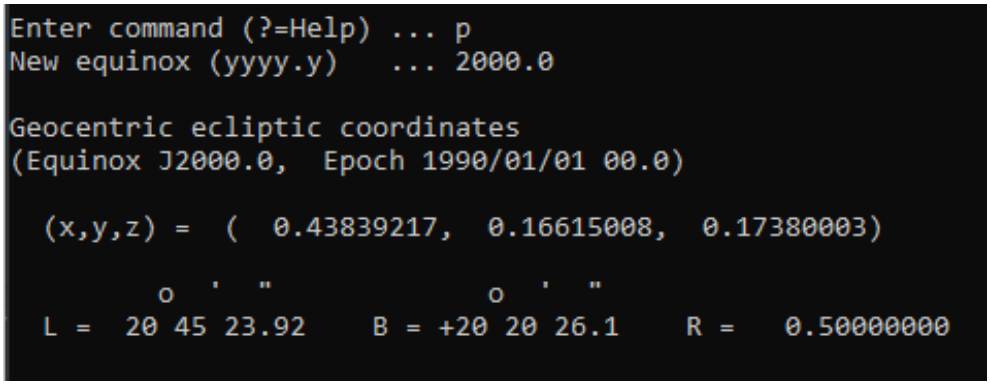

Рисунок 8 – Координаты точки равноденствия в эпоху N.

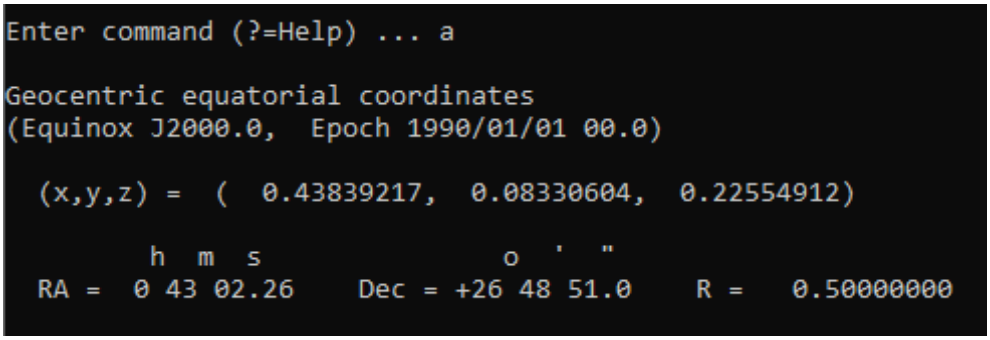

Рисунок 9 – Переход к О координатам.

```
Enter command (?=Help) ... h

Heliocentric equatorial coordinates
(Equinox J2000.0,  Epoch 1990/01/01 00.0)

  (x,y,z) =  (  0.26010919,  0.97053782,  0.61023476)

         h  m  s              o  '  "
  RA =  4 59 59.28    Dec = +31 16 17.1    R =   1.17557940
```

Рисунок 10 – Переход к P координатам.5

```
Enter command (?=Help) ... e

Heliocentric ecliptic coordinates
(Equinox J2000.0,  Epoch 1990/01/01 00.0)

  (x,y,z) =  (  0.26010919,  1.13318849,  0.17382167)

         o  '  "             o  '  "
  L =  77 04 20.67    B = + 8 30 10.6    R =   1.17557940


Enter command (?=Help) ... p
New equinox (yyyy.y)   ... 1970.0

Heliocentric ecliptic coordinates
(Equinox J1970.0,  Epoch 1990/01/01 00.0)

  (x,y,z) =  (  0.26839147,  1.13126758,  0.17374292)

         o  '  "             o  '  "
  L =  76 39 12.33    B = + 8 29 56.7    R =   1.17557940


Enter command (?=Help) ... g

Geocentric ecliptic coordinates
(Equinox J1970.0,  Epoch 1990/01/01 00.0)

  (x,y,z) =  (  0.43959680,  0.16295108,  0.17378608)

         o  '  "             o  '  "
  L =  20 20 20.00    B = +20 20 20.0    R =   0.50000000
```

Рисунок 11 – Возвращение к исходной точке.

Сравним значения с рис. 7. Все значения совпали.

3) Решим задачу №2. По заданной модифицированной юлианской дате MJD получим эклиптические и экваториальные координаты. Значение $MJD = 3432.1$.

Для решения задачи возьмем вектор (4, 3, 5) и допишем следующие строчки кода в Coco.cpp:

```
double MJD = 3432.1;
double T = (MJD - 2400000.5)/36525;
Vec3D m_R(4,3,5);
cout << " (x,y,z) = " << fixed << setprecision(8) << setw(12) << m_R << endl;
cout << "ecliptical coordinates" << endl;
Vec3D m_R1 = Equ2EclMatrix(T) * m_R; // Ecliptic;
cout << " (x,y,z) = " << fixed << setprecision(8) << setw(12) << m_R1 << endl;
cout << " o ' \" o ' \"" << endl;
cout << " L = " << setprecision(2) << setw(12) << Angle(Deg*m_R1[phi],DMMSSs);
cout << " B = " << setprecision(1) << showpos << setw(11)
<< Angle(Deg*m_R1[theta],DMMSSs) << noshowpos;
cout << " R = " << setprecision(8) << setw(12) << m_R1[r] << endl;
cout << endl << endl;
cout << "equatorial coordinates" << endl;
Vec3D m_R2 = Ecl2EquMatrix(T) * m_R; // Equator;
cout << " (x,y,z) = " << fixed << setprecision(8) << setw(12) << m_R2 << endl;
cout << " h m s o ' \"" << endl;
cout << " RA = " << setprecision(2) << setw(11)
<< Angle(Deg*m_R2[phi]/15.0,DMMSSs);
cout << " Dec = " << setprecision(1) << showpos << setw(11)
<< Angle(Deg*m_R2[theta],DMMSSs) << noshowpos;
cout << " R = " << setprecision(8) << setw(12) << m_R2[r] << endl;
cout << endl << endl;
```

Рисунок 12 – Добавленный код

Результат работы программы представлен на рис. 14.

```
 (x,y,z) =  (  4.00000000,  3.00000000,  5.00000000)

ecliptical coordinates
 (x,y,z) =  (  4.00000000,  4.78304274,  3.33504155)

 o ' " o ' "
 L =  50 05 40.83 B = +28 08 28.3 R =   7.07106781


equatorial coordinates
 (x,y,z) =  (  4.00000000,  0.69184008,  5.78976315)

 h m s o ' "
 RA =  0 39 15.07 Dec = +54 57 52.1 R =   7.07106781
```

Рисунок 13 – Полученные эклиптические и экваториальные координаты

Выводы.

В ходе выполнения лабораторной работы были изучены различные инструментальные средства для решения задач поддержки принятия решения, а также были освоены навыки принятия решений на основе задач астрономии. Используя соответствующие программы на языке C++ были решены задачи принятия решения в астрономии.