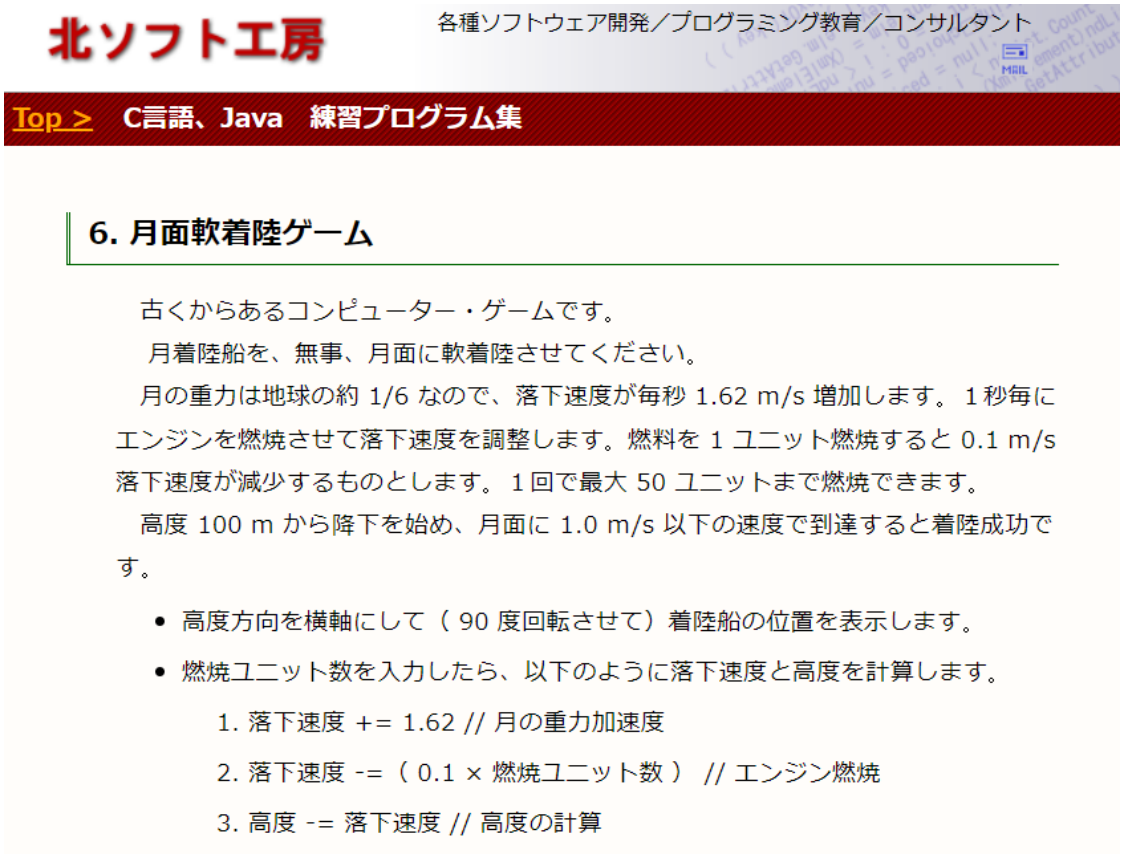
作成者: mitsuya.85u@ray.ocn.ne.jp

6.2. スレッド対応（月面着陸）

月面着陸シミュレーションプログラムをスレッド対応で作成します。
以下のサイトの問題から実装します。

https://kitako.tokyo/lib/CTask.aspx?id=6

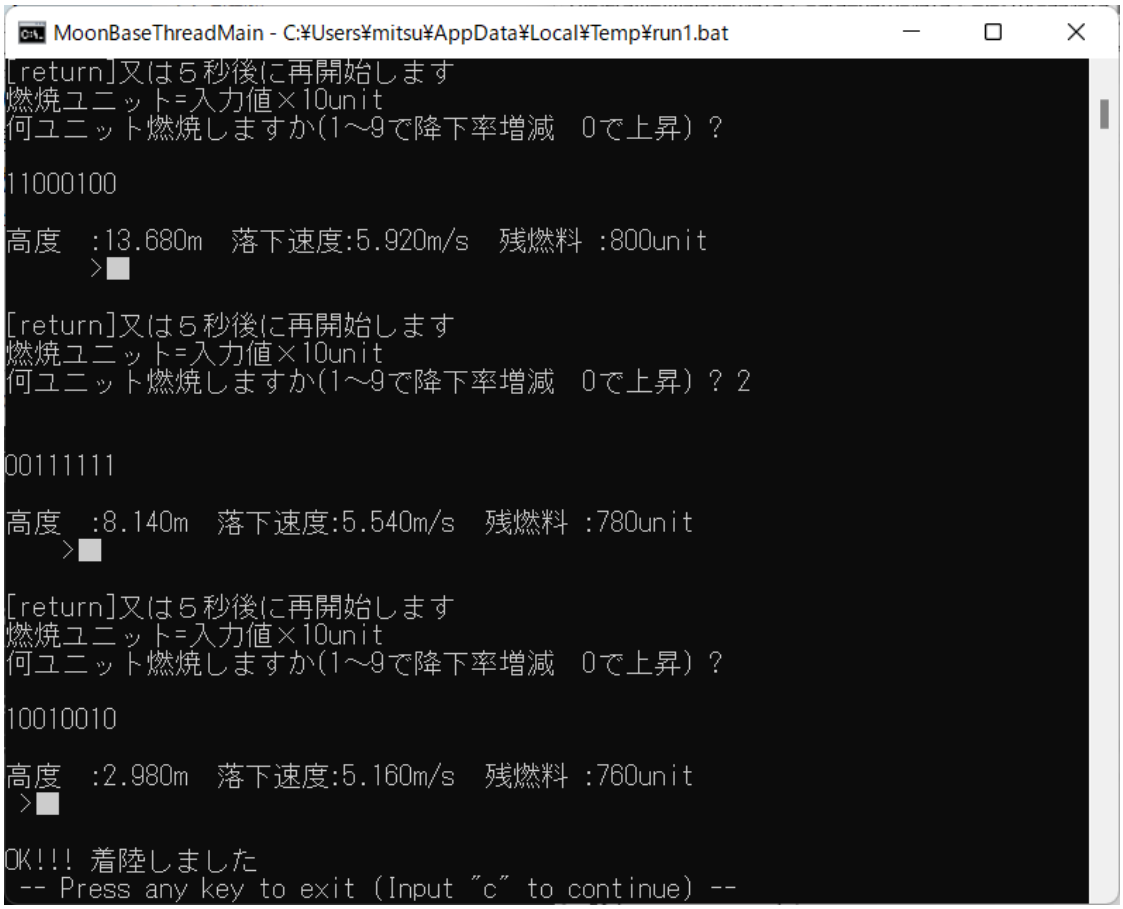


このプログラムが実体化されると、8ビットの数字列の回転表示を行います。別スレッドからインタラプトをかけることで回転を止め、その時点での月着陸船の状態を計算して表示したのち、さらに回転を続けます。条件を多少緩和するために着陸条件を以下のように変えます。

【着陸条件】
1．高度０メートル以上4m未満
2．落下速度０m/s以上10m/s以下
3．初期燃料1000ユニット

（初期燃料は起動クラスから指定、また残燃料の確認も起動クラスで実施）

【実行例】



【ソースコード】MoonBaseThread.java

```
public class MoonBaseThread extends Thread{↓
  double fallSpeed,altitude;↓
  int fuelUnit,burningUnit;↓
↓
public MoonBaseThread(){↓
  fallSpeed=0.0;↓
  fuelUnit=250;↓
  burningUnit=0;↓
  altitude=100;↓
}↓
public void run(){↓
    while(true)[↓
      try[↓
          rolling();↓
          Thread.sleep(70);↓
      } catch (InterruptedException ie) {↓
          calc();                    ↓
          result();↓
          if( (altitude < 4 && altitude >= 0) && (fallSpeed <= 10 && fallSpeed >= 0) )[↓
            position();↓
            System.out.println("¥nOK!!! 着陸しました");↓
//          break;↓
            System.exit(0);↓
          ]↓
          if( altitude < 0 )[↓
            System.out.println("¥nNG... 激突しました");↓
//          break;↓
            System.exit(0);↓
          ]↓
          try [↓
            position();↓
                       ↓
            System.out.print("¥n[return]又は５秒後に再開始します¥n");↓
            System.out.print("燃焼ユニット=入力値×10unit¥n");↓
            System.out.print("何ユニット燃焼しますか(1～9で降下率増減□0で上昇）？ ");↓
            Thread.sleep(5000);↓
            System.out.println("¥n");↓
          ]catch(InterruptedException e)[↓
//          e.printStackTrace();↓
//          System.out.println("リターンキー連打でここに入る");↓
          ]↓
      ]↓
  ]↓
]↓
public void rolling(){↓
          int numA=new java.util.Random().nextInt(2);↓
          int numB=new java.util.Random().nextInt(2);↓
          int numC=new java.util.Random().nextInt(2);↓
          int numD=new java.util.Random().nextInt(2);↓
          int numE=new java.util.Random().nextInt(2);↓
          int numF=new java.util.Random().nextInt(2);↓
          int numG=new java.util.Random().nextInt(2);↓
          int numH=new java.util.Random().nextInt(2);↓
↓
          System.out.printf("¥b¥b¥b¥b¥b¥b¥b¥b%d%d%d%d%d%d%d%d",numA,numB,numC,numD,numE,
]↓
public void result(){↓
          System.out.printf("高度  :%.3fm□",altitude);↓
          System.out.printf("落下速度:%.3fm/s□",fallSpeed);↓
          System.out.printf("残燃料 :%dunit",fuelUnit);↓
          System.out.println("");↓
]↓
```

```
public void position(){
    for(int i=1;i<=(int)altitude/2;i++){
        System.out.print(" ");
    }
    System.out.println(">■");
}
public void calc(){
  fallSpeed += 1.62;
  fallSpeed -= (0.1*burningUnit);
  altitude -= fallSpeed;
  fuelUnit -= burningUnit;
}
public double getAltitude() {
 return altitude;
}
public double getFallSpeed() {
 return fallSpeed;
}
public int getFuelUnit() {
 return fuelUnit;
}
public void setBurningUnit(int b) {
 this.burningUnit=b;
}
public void setFuelUnit(int f) {
 this.fuelUnit=f;
}
}
```

【ソースコード】MoonBaseThreadMain.java

```
public class MoonBaseThreadMain{
    public static void main(String[] args){
        MoonBaseThread mbt = new MoonBaseThread();
        mbt.setFuelUnit(1000);  //燃料の量はメインメソッドで設定
        mbt.start();
        while(true){
         String s=new java.util.Scanner(System.in).nextLine();
         if (s.equals("0")){mbt.setBurningUnit(0);}
         if (s.equals("1")){mbt.setBurningUnit(10);}
         if (s.equals("2")){mbt.setBurningUnit(20);}
         if (s.equals("3")){mbt.setBurningUnit(30);}
         if (s.equals("4")){mbt.setBurningUnit(40);}
         if (s.equals("5")){mbt.setBurningUnit(50);}
         if (s.equals("6")){mbt.setBurningUnit(60);}
         if (s.equals("7")){mbt.setBurningUnit(70);}
         if (s.equals("8")){mbt.setBurningUnit(80);}
         if (s.equals("9")){mbt.setBurningUnit(90);}
         if (s.equals("")){
            System.out.println("");
            mbt.interrupt();
            if(mbt.getFuelUnit()<=0){
                System.out.println("<<<燃料がありません>>>");
                break;
            }
         }
        }
        mbt.stop();
    }
}
```