

## MonteCarloPiKernel.java

Implementiert Kernel-interface und stellt Konstruktor und private Member-Variablen zur Verfügung

## ...Kernel.java

Weitere benötigte Kernelimplementationen

## MonteCarloPi.scala

- Initialisiert Rootbeer-Kontext und bindet dieses an eine GPU
- Startet Kernel

## MonteCarloPiSpark.scala

- erstellt SparkContext, RDD und führt darauf map und reduce aus

```
javac <name>.class -classpath  
$(ROOTBEER_ROOT)/Rootbeer.jar:..
```

```
jar -cvfm gpu.tmp.jar  
*Kernel.class
```

## gpu.tmp.jar

enthält ...Kernel.class

```
java -jar $(ROOTBEER_ROOT)/  
Rootbeer.jar MontePiGPUtmp.jar  
MontePiGPU.jar -64bit -  
computecapability=sm_30
```

## gpu.jar

Fertiggestellte MontePi-Klasse, welche auf einer GPU ausgeführt wird

## Rootbeer.jar

## scala-library.jar

```
scalac *.scala -classpath  
"$SPARK_ROOT"/core/target/spark-  
core_2.10-1.5.2.jar:"$ROOTBEER_ROOT"/  
Rootbeer.jar:/gpu.jar:..
```

```
MonteCarloPiSpark$$anonfun$1.class  
MonteCarloPiSpark$$anonfun$2.class  
MonteCarloPiSpark$$anonfun$main$1.class  
MonteCarloPiSpark.classTestMonteCarloPi$.class
```

```
jar -cvfm cpu.jar manifest.txt  
MonteCarloPiSpark*.class
```

## cpu.jar

enthält alle Klassen, die nicht mit Rootbeer kompiliert werden müssen

## MontePi.jar

Fertiggestellte MontePi-Klasse, die mit Spark ausgeführt werden kann

```
zipmerge MontePi.jar /usr/share/java/  
scala-library.jar gpu/MontePiGPU.jar  
MonteCarloPiSpark.jar
```