

Projektbeskrivning

<Projekttitel>

2042-03-22

Projektmedlemmar:

Noah Ljungberg <noalj314@student.liu.se>

Handledare:

Hampus Holm <hamho999@student.liu.se>

Innehåll

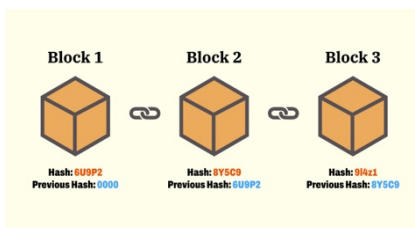
1. Introduktion till projektet.....	2
2. Ytterligare bakgrundsinformation	2
3. Milstolpar.....	2
4. Övriga implementationsförberedelser	4
5. Utveckling och samarbete	4
6. Implementationsbeskrivning	6
6.1. Milstolpar.....	6
6.2. Dokumentation för programstruktur, med UML-diagram.....	6
7. Användarmanual	7

Projektplan

1. Introduktion till projektet

Mitt projekt går ut på att skapa en egen blockkedja som kommer hantera en egen kryptovaluta. Den kommer använda konceptet Proof of Work vilket egentligen betyder att datorn utför beräkningar för att lösa blocket, så en dator blir en miner, eller grävare. Genom att använda din dator för att lösa en beräkning får du betalt helt enkelt.

Programmet kommer vara något som användare kan ladda ner för att delta i mining-processen. Om du är bekant med bitcoin så kommer detta vara ungefär likadant. Så programmet kommer att använda datorns beräkningskraft för att lösa algoritmer. Sedan när datorn har löst en algoritm så får man en vinst i form av en viss kryptovaluta. När det sedan är löst så skickar den datorn (eller noden) lösning till resten av datorerna i blockkedjan för att få den bekräftelse att den är rätt.



2. Ytterligare bakgrundsinformation

Blockkedjan kommer ha massor med block. Ett block kommer innehålla information, inklusive ett hash. Hashet beror på all annan data i blocket, det kommer också innehålla det föregående blockets hash. Så eftersom alla block innehåller det tidigare blockets hash, så blir det en kedja, en blockkedja.

Proof of Work går ut på att datorn arbetar för att lösa en algoritm. Säg att vi vet om x , men inte y och vi ska ta reda på z så $z * y = x$ så vi kommer gissa detta till vi hittar ett svar som blir lika med x , bara ett exempel, algoritmen kommer vara svårare.

3. Milstolpar

#	Beskrivning
1	Block-klassen ska skapas den ska innehålla information om den aktuella hashen och den föregående hashen för det tidigare blocket Testprogrammet kan skapa ett enkelt block med en statisk data
2	Skapa blockkedjan som kommer vara en klass som lagrar flera block där varje block är länkat till det tidigare blocket genom dess hash. Detta kommer testas genom att skapa flera block och jämföra hashar för att säkerställa

	integriteten av kedjan.
3	Implementera proof of work, lägga till proof of work mekanismen i board-klassen för att se till att block inte kan skapas utan arbete.
4	Skapa en mineBlock metod. Testen ska vara ett program som kräver beräkningsarbete för att lägga till block i kedjan.
5	Skapa en transaktions klass som hanterar överföringar mellan olika valutaadresser Testet innefattar ett program som kan lagra och skapa transaktioner i blocken
6	Skapa möjligtvis en GUI för en plånbok
	...

4. Övriga implementationsförberedelser.

De klasser som kommer finnas är transaktioner, block, blockkedja, testprogram.

