**Enterprise Information Systems –**

**Business- & Management-Prozesse**

**Gruppe XY**

**Michael Kirchknopf (1126331 / 033 526)**

**Matthes Koenen (1127522 / 033 526)**

**Sebastian Scheickl (1127548 / 033 526)**

**Wien, Mai 2014**

**Übungsteil: Kurzbeschreibung**

1. Beschreibung der CPPI-Strategie – Max. 5 Punkte
2. Zuordnung eines PDCA-Management-Infrastruktur – Max. 10 Punkte
3. Implementierung & Dokumentation der Implementierung – Max. 15 Punkte

Inhaltsverzeichnis

[1 Beschreibung der CPPI-Strategie 3](#_Toc387315869)

[1.0 Beschreibung der Aufgabenstellung 3](#_Toc387315870)

[1.1 Lösung 3](#_Toc387315871)

[2 Zuordnung eines PDCA-Management-Infrastruktur 3](#_Toc387315872)

[2.0 Beschreibung der Aufgabenstellung 3](#_Toc387315873)

[2.1 MGT-Aktivitätsdiagramm inkl. Begründung 3](#_Toc387315874)

[2.2 Aktivitäten und Informationen 3](#_Toc387315875)

[3 Dokumentation der Implementierung 3](#_Toc387315876)

[3.0 Beschreibung der Aufgabenstellung 3](#_Toc387315877)

[3.1 Lösung 3](#_Toc387315878)

# Beschreibung der CPPI-Strategie

## Beschreibung der Aufgabenstellung

Beschreibung der CPPI-Strategie mit eigenen Worten. Identifizieren und beschreiben Sie dabei die beteiligten Aktivitäten und Datenflüsse.

## Lösung

CPPI steht für [Constant Proportion Portfolio Insurance und beschreibt eine Portfolio Sicherungsstrategie damit ein bestimmtes Renditeniveau garantiert werden kann.](http://de.wikipedia.org/wiki/Constant_Proportion_Portfolio_Insurance) Ziel der Portfolioabsicherungsstrategien ist es, das Verlustrisiko im Falle sinkender Kurse an den Wertpapiermärkten zu begrenzen und gleichzeitig eine Partizipation an steigenden Wertpapiermärkten zu ermöglichen.

Zu Anfang der Strategie bestimmt man einen Wert, den man am Ende mindestens erreichen will. Nun sucht man sich eine möglichst sichere Kapitalanlage und zinst den Wert über die Laufzeit ab. Damit weiß man wie viel Geld man heute in die sichere Kapitalanlage investieren muss um zum Laufzeitende genau den vorher festgelegten Mindestwert zu erhalten.

Die CPPI Strategie berechnet jetzt wie viel Geld zu jedem Zeitpunkt in die sichere Kapitalanlage investiert sein muss um am Ende mindestens den Sicherungswert zu erhalten. Mit dem Kapital was über den Wert der sicheren Kapitaleinlage hinausgeht kann in risikoreiche Geschäfte investiert werden, die aber auch höhere Zinsen versprechen. Die Strategie berechnet jetzt dynamisch zu jedem Zeitpunkt wie viel Geld in die sichere und wie viel in die unsichere Kapitalanlage investiert werden sollte. Zusätzlich gibt es in der CPPI Strategie noch einen Parameter mit dem ich das Risiko etwas erhöhen kann. Und einen Parameter mit dem ich den Maximalen Anteil an risikoreichen Geschäften begrenzen kann.

**Beispiel:** Wenn ich heute 500€ habe und ich möchte diese Investieren, dabei aber sicher stellen dass ich nach einem Jahr wenigstens noch meine 500€ habe. Und ich bekomme bei der Bank 5% Zinsen, dann muss ich heute 476,19€ fest anlegen um bei 5% Zinsen nach einem Jahr genau wieder meine 500€ zu haben.

Das bedeutet ich habe von den 500€ noch 500€-476,19€ = 23,81€ übrig. Falls m = 1 kann ich diese 23€ jetzt risikoreicher investieren um dafür zu sorgen, dass ich nach einem Jahr vielleicht sogar etwas mehr als die 500€ habe. Die Aktienkurse schwanken natürlich und die Anlage ist nicht so sicher wie mein Geld bei der Bank. Durch die CPPI Strategie wird aber trotzdem garantiert, dass selbst wenn sich die Aktienkurse ganz schlecht entwickeln ich am Ende des Jahres trotzdem mindestens meine 500€ wieder habe. Bei guter Aktienkurs Entwicklung ist natürlich auch mehr möglich.

# Zuordnung eines PDCA-Management-Infrastruktur

## Beschreibung der Aufgabenstellung

Ordnen Sie die identifizierten Aktivitäten einer PDCA-Management-Infrastruktur (Aktivitäten und Informationsobjekte) zu. Begründete Entscheidung für eine konkrete Management-Infrastruktur, welche zur Implementierung der CPPI-Strategie verwendet wird.

## MGT-Aktivitätsdiagramm inkl. Begründung



## Aktivitäten und Informationen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Stereotyp** | **Bezeichnung** | **Beschreibung** |
| 1 | [Plan]  Aktivität | Wenigstens Mindestportofoliowert erreichen | Es gilt am Ende der Periode eine möglichst hohe Redite zu erzielen ohne aber unter den Mindestportfoliowert zu fallen |
| 1a | [Objective]  Information | Portfolio maximieren: | Floorbarwert Ft = SOLL |
| 1b | [Do Rule]  Vorschrift | Investieren | Investitionsvorschrift |
| 1c | [Control Rule]  Vorschrift | Cushion = IST – SOLL | Eine hohe Cushion ermöglicht einen hohen Risikokapitalanteil. Ohne den Sicherungswert zu unterschreiten. |
| 1d | [Control Rule]  Vorschrift | Regelungsvorschrift:  Risikoinvestment:  Min(m\*C; b\*W(t))  Risikolos:  W(t) - Risikoinvestment | Wir investieren den Cushionwert mal einem freiwählbaren Multiplikator in Risikoinvestments. Maximal aber nur b% von unserem aktuellen Portfoliowert. Den Rest in risikolose Kapitalanlagen. |
| 1f | [Measure Rule]  Vorschrift | Portfoliowert aus Aktienrendite berechnen | Kursänderung an der Börse auswerten. |
| 2 | [Measure]  Aktivität | Portfoliowert | Aktienkurs wird gemessen. |
| 2a | [Performance]  Aktivität | Messergebnis: Portfoliowert | Portfoliowert: Wt = IST |
| 3 | [Check]  Aktivität | Vergleich= IST – SOLL | Sobald nur noch in die sichere Anlage investiert wird, weil keine Cushion mehr für die Risikoinvestitonen zu Verfügung steht, ist die Investmentstrategie beendet. |
| 3a | [Deviation]  Information | Cushion | Abweichung vom Soll- Ist Vergleich |
| 4 | [Act]  Aktivität | Regelung: Portfoliooptimieren | Nach den Vorschriften wird hier berechnet wie viel Kapital in welche Anlage investiert wird. |
| 4a | [Control Input] | Anpassung der Investmentanteile | Aufteilen der Anteile. |
|  |  |  |  |

# Dokumentation der Implementierung

## Beschreibung der Aufgabenstellung

Implementieren Sie die CPPI-Strategie. Verwenden Sie dafür die vorgesehenen abstrakten Klassen. Dokumentieren Sie welche Klassen und Methoden konkret welchen CPPI-Aktivitäten entsprechen. Verwenden Sie dafür Code-Snippets für eine detaillierte Beschreibung. (Programmierer-Perspektive).

Im Source-Code finden Sie im Ordner „src“ folgende Pakete:

* at.ac.tuwien.imw.pdca

Hier befinden sich alle abstrakte Klassen und Interfaces die Sie für eine Implementierung benötigen. Je nachdem welche Management-Infrastruktur gewählt wird, sollen die jeweiligen Klasse ableiten und implementiert werden. Hier sollen Sie KEINE Änderung vornehmen!

* at.ac.tuwien.imw.pdca.cppi

In diesen Packet sollen die abgeleiteten Klassen angelegt werden und implementiert werden.

Zum Beispiel:



* at.ac.tuwien.imw.pdca.cppi.service

Hier befindet sich unter anderen die Klasse „CPPISimulation.java“ die die Main-Methode enthält. Hier sollen die verschiedenen Prozesse gestartet werden.

## Lösung

**Folgende Variablenbelegung haben wir uns überlegt (Im Kommentar steht der Variablenname aus den Folien welche unsere Meinung nach den Variablen entsprechen)**

private BigDecimal portfolio; // F(T)

private BigDecimal tsr; // TSR

private BigDecimal floor; // F(t) Sicherer Teil um am Ende W(t) zu haben.

private BigDecimal cushion; // Cushion

private BigDecimal exposure; // W(t)

private BigDecimal reserveasset; //ausgelassen

private BigDecimal partRiskyAsset; // X(r,t)

private BigDecimal partRisklessAsset; // X(f,t)

private BigDecimal previousStockPrice; // S(t-1)

private BigDecimal actualStockPrice; // S(t)

private BigDecimal riskAssetPercent = new BigDecimal(0.3); // b

private BigDecimal risklessAssetPercent = new BigDecimal(0.7); // 1-b

private Integer risklessAssetLastDays = 365; // d

private BigDecimal risklessAssetInterest = new BigDecimal(0.05); // R0

private BigDecimal laverage = new BigDecimal(2.0); // m

private BigDecimal portfolio = new BigDecimal(100); // F(T)

**Klassenaufteilung**

<<Plan>> → CPPIPlanProcess.plan()

<<Plan Rules>> → CPPIPlanRules.applyPlanningRules()

<<Objective>> → CPPIObjectiveSetting

<<Performance>> → CPPIMeasuredPerformanceValue

<<Check>> → CPPICheckProcess.getCheckResult(ObjectiveSetting,

MeasuredPerformanceValue)

<<Control Rules>> → CPPICheckRules.applyCheckingRules()

<<Act>> → CPPIActProcess.act(Deviation)

<<Control>> → CPPIActRules.applyActRules()

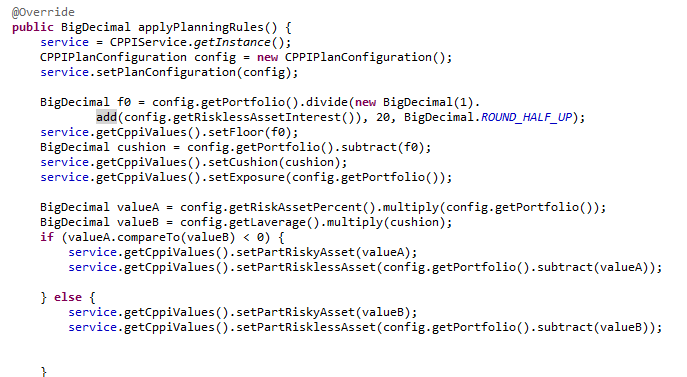
<<Do>> → CPPIDoProcess.operate()

<<Do Rules>> → CPPIDoRules.applyDoRules()

**Threads**

Die Threads laufen nicht synchronisiert. Jeder Prozesschritt hat 100 ms Zeit für seine Berechnung. Die Simulationsgeschwindigkeit haben wir von 10 Sekunden auf 1 Sekunde reduziert. Wenn die Geschwindigkeit der Simulation geändert werden, muss neben der Geschwindigkeit des StockpriceGenerators auch die Geschwindigkeit der Do, Check und Act Prozesse angepasst werden. (Alle Prozesse brauchen den selben sleep() Wert).

**CPPIPlanProcess**

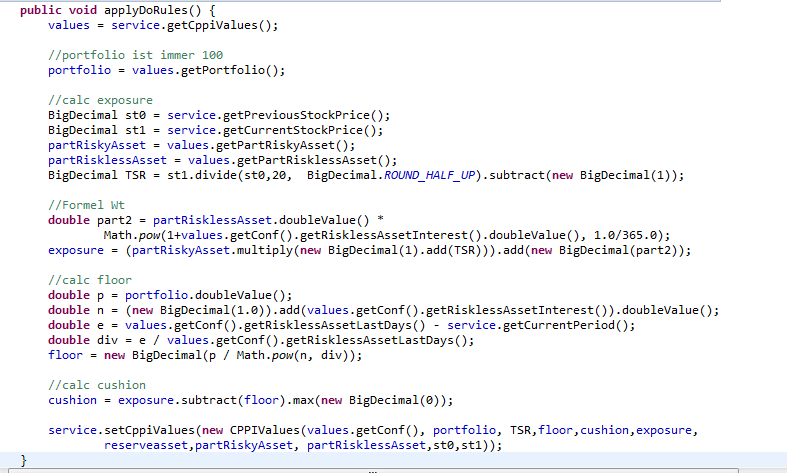


Im Plan gibt es 2 Klassen, CPPIPlanProcess und CPPIPlanning Rules. CPPIPlanProcess ist der Thread, der die PlanningRules aufruft. Die aufgerufene Methode applyPlanningRules ist auf dem Screenshot ersichtlich.

Hier wird unser Service initialisiert, so dass es überall verwendbar ist, da es ein Singleton ist. In weiterer Folge werden in der nullten Periode alle notwendigen Parameter initialisiert, welche zuvor in CPPIPlanConfiguration spezifiziert wurden. Dann werden noch für die nullte Periode die Anteile von risikolosem und risikoreichem Anteil des Portfolios berechnet, was später im Act passieren wird. Diese Werte werden in unserer CPPIValues Instanz gespeichert, auf welche wir in den Rules immer wieder zugreifen.

Der PlanProcess wird bei uns nur einmalig ausgeführt, da wir eine Single-Closed-Loop gewählt haben.

**CPPIDoProcess**

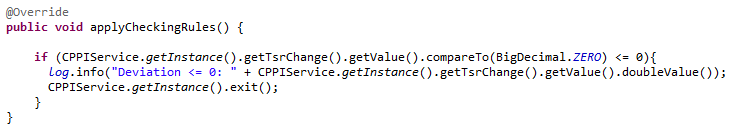


Vor dem DO-Prozess wird vom CPPIStockPriceGenerator der neue Aktienwert, der Wert der risikoreichen Aktien. Nach dieser Periode wird unser CPPIDoProcess aufgerufen, welche die im Screenshot zu sehende Methode anstößt.

In dieser Methode wird der aktuelle Aktienwert berechnet. Mit Hilfe dieses Wertes kann ein neuer Portfoliowert zum Zeitpunkt t berechnet werden. Dieser ist in unserem Beispiel der IST-Wert. Dazu wird der Floorvalue, also der Wert der zur aktuellen Periode mindestens risikolos angelegt werden muss um am Ende zu dem Zielwert zu gelangen. Dieser wird einfach durch Abzinsen des Zielwertes, über die bis zur Zielzeit verbleibenden Zeit, berechnet. Er dient dazu einen SOLL-Wert zum Zeitpunkt t zu berechnen. Als dritten Schritt wird auch die Cushion berechnet. Diese ist die Differenz aus IST und SOLL, mindestens jedoch 0. Laut Folien sollten diese Berechnungen im Check-Prozess stattfinden, da wir allerdings der Ansicht sind, dass Berechnungen nicht im Check-Prozess sondern in einem vorhergehenden passieren sollten haben wir diese im DO-Prozess realisiert.

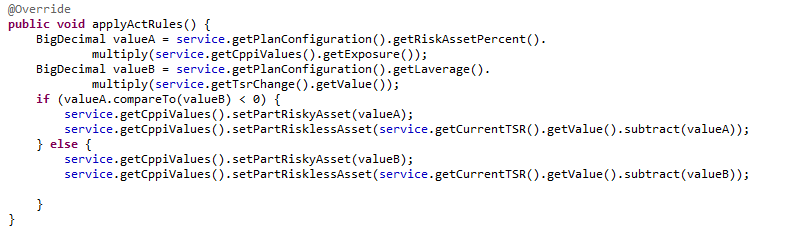
Aus all den Werten wird durch einen überladenen Konstruktor eine neue Instanz von CPPIValues erstellt, welche die aktuellen Werte beinhält.

**CPPICheckProcess**



Der CHECK-Prozess wird über die Klassen CPPICheckProcess sowie CPPICHeckingRules realisiert. Dabei wird einfach überprüft ob die vorher berechnete Deviation kleiner oder gleich null ist. Sollte dies der Fall sein, wird der gesamte CPPI-Prozess abgebrochen und zukünftig 100% in das risikoarme Anlagen investiert. Sollte noch immer eine Deviation, bzw. Cushion vorhanden sein, wird mit dem ACT-Process fortgefahren.

**CPPIActProcess**



Auch hier ruft unsere CPPIActProcess die Methode applyActRules der Klasse CPPIActRules auf. In dieser Methode wird der Anteil des riskanten Anteils berechnet. Dies geschieht mit einem Vergleich der beiden Parameter der Formel um das Minimum herauszufinden. Das ausgerechnete Minimum wird in das risikoreiche Investment investiert und um auf den risikolosen Anteil zu kommen wird der ausgerechnete Anteil vom Portfoliowert abgezogen.