**Semestrální práce**

Autoři : Rakhimova Aiya

**Funkční požadavky**

**F1. Entity**

1. FoodChain – reprezentuje celkový system a zabezpečuje komunikaci s uživatelem
2. Farmer – zdroj potravin
3. Processor – zpracovává potraviny od farmáře
4. Storehouse – skladuje potraviny
5. Dealer – doručuje potraviny ze skaldu do obchodu
6. Seller –prodává potraviny zákaznikovi
7. Customer – uživatel, který objednává potraviny
8. Product – samotná potravina

**F2. F3. Transakce**

Transakce je popsána ve třídě TransactionBlock. Třídá má nasledující proměnné: produkt, text a party, která provědla transakce. Transakční řetězec se provádí ve třídách TransactionChain a TransactionIteartor. TransactionChain třída je zodpovědná za vytvoření/přidání nového bloku do řetězce a TransactionIteartor je iterator.

**F4. Informace o potravině**

Po dokončení všech operací můžete vygenerovat dokument, ve kterém jsou uloženy všechny zprávy o tom, přes jaké parties produkt prošel a co s ním bylo provedeno.

**F5. F9. Kanály**

Projekt obsahuje 2 kanály: kanál pro přenos produktů (ProductChannel) a platební kanál (MoneyChannel). Prostřednictvím potravinového kanálu se provádí přenos dotazů mezi parties. Platba za zboží probíhá prostřednictvím platebního kanálu. Každá party se může připojit nebo opustit kanál

**F6. F7. Double spending**

To, bohužel, v projektu není

**F8. Stavový automat**

Fáze(state) produktu se mění při přechodu z jedné party do druhé

**F10. Reporty**

Projekt umožňuje generovat Party report, Product Report a Transaction report(reporty celého systému). Security report v projektu není realizován.

**Nefunkční požadavky**

* Není požadována autentizace ani autorizace **- splněno**
* Aplikace nemá GUI. Aplikace komunikuje s uživatelem pomocí command line, případně výpisů do souboru **- splněno**
* Aplikace má běžet pouze v jedné JVM **- splněno**
* Dobře skryté metody a proměnné, které nemají být dostupné ostatním třídám **- splněno**
* Reporty jsou generovány do textového souboru **- splněno**
* Konfigurace ekosystému může být vytvořena přímo kódem z třídy nebo externího souboru (např json) **- splněno**

**Patterny**

1. Factory – PartyFactory/createParty()
2. Singleton – FoodChain(system)/getInstance()
3. Builder – třída FoodChain
4. Strategy – Product/setProductParams()
5. State – State/nextTick(), nextState()
6. Iterator – TransactionChainIterator/hasNext(), next()

**Požadované výstupy**

* 2 konfigurace **– částečně splněno(funguje jen s 6 parties)**
* Use case diagram, uml diagram, stručný popis **– splněno**
* JavaDoc **– splněno**