

<https://bitbucket.org/markushilding/objectorienteddesignsem3/src>

Lätt att förstå koden, metoderna har tydliga och representative namn. Kod konventioner ser ut följas till stor grad, upptäckte ingen duplicerad kod. Koden har hög grad av inkapsling då primitiv data "göms" på ett bra sätt. Upplever att koden har låg koppling och hög nivå av sammanhållning. Kunde ej hitta några kommentarer inuti metoderna. Paket strukturen känns rimligt uppbyggd. Diskussionen känns relevant för problemet. Finns relevanta tester

I View.pay finns raden: `if(change == -1)`, kunde varit något tydligare.

<https://github.com/bagtab/cashierSystem>

Innehållsförteckning och introduction saknas. Koden har rimliga kommentaren och kan ej hitta några kommentarer inuti metoderna. Enhetstester ser ut att täcka de flesta scenariona I CostTest kan det som sker i metoden fill(), flyttas till before().

Överlag är det en lätt läst kommentar. I stor utsträckning används objekt istället för primitiv data och statiska objekt. Parameter listorna känns rimliga och ingen som sticker ut allt för mycket. Paket strukturer är lätt att följa och känns god.

Det lilla som stor i diskussionen känns relevant men diskussionen är ganska tunn och "programming best practices" förklaras inte.

<https://github.com/lunjoa/PointOfSaleAssignment/>

Koden var relativt lättöverskådlig och metoderna har själv förklarande namn och kodkonventionen ses ut att följas. Objekt används för att dölja primitiv data. Paket strukturen känns rimligt och har inga kommentarer på den. Programmet verka ha någorlunda låg koppling, hög sammanhållning och bra inkapsling, ingen som sticker ut på ett negativt sätt. Programmet ser ut att ha en flexibel kod. Enhetstestningen var gedigen, och ser åtminstone ut att täcka allt och lite till. Kommenteringen ser ut att följa Javadocs och kunde inte hitta några kommentarer i någon metod. Innehållet i diskussionen känns både rimligt och håller en bra längd.