1. Hvad starter programmet med at gøre?

Programmet starter med at finde main metoden, og herefter checker om der er noget data der kan loades eller om der skal oprettes nyt.

2. Hvordan afgøres om brugeren skal have mulighed for at starte et nyt spil eller sættes direkte til flowet hvor man registrerer spillere?

Brugern har mulighed for selv at vælge hvad om de vil loade gammelt eller registrerer nye spiller

3. Hvorfor kører der et while loop inde i runGameLoop() ? Hvad sker der hvis man taster N?

For at checke om spillern stadig vil spille videre

4. Hvad er formålet med betingelsen inde i kroppen af while-loopet, hvor det tjekkes om playerCount == antallet af spillere?

Spillet sørger for at der stadig er spillere tilbage i spillet før den laver en ny runden

5. Se på metoden doTurn(Player player) Forklar hvordan currentPosition får sin værdi. Hvad returnerer board.getField() (prøv at være mere specifik end blot at nævne datatypen)

Spilleren får en værdi fra en metode der hedder throwDice der returnerer et nummer som spilleren skal rykke frem. Det nummer sender vi videre til en anden metode der opdatere spillernes position på mappet, og checker om spillern har passeret start. Hvis de passere start får de et beløb tilsvarende at have passeret start. Hvis de lander på start får de mere. Borad.getField retunere en action tilsvarende til det felt den modtager.

6. Under hvilke omstændigheder kaldes metoden processesResponse(Action a, Field f)

Denne metode kaldes når en spiller lander på en grund der kan købes eller en grund en anden spiller har købt. Spillern får muligheden for at købe en grund eller de har mulighed for at betale en anden spiller for at lande på grunden. Spilleren kan inputte hvad de gerne vil.

7. Hvorfor står der et ord i en parentes foran variablen f i linie 85 – det samme i linie 100 ?

Fordi vi kaster en type Field ind som en type Property. Vi ændrer simpelthen den type data som Field er, som bliver sendt videre fra der hvor metoden er kaldt.

8. Giv et eksempel på en konkret værdi der kan returneres ved metodekaldet i linje 86 a.getResponseMsg()

Spilleren kan skrive ”buy” for at købe feltet de er landet på.

9. Forklar helt overordnet hvilket ansvar der hviler på metoden processesResponse()

Denne metode skal håndtere hvad der sker når en spiller lander på et felt og skal finde ud af hvad spilleren gerne vil

10. List alle klasser i programmet som ikke nedarver fra Field

Alt for mange..

11. Forklar formålet med at køre switch-casen i Board klassens setFields()

For at finde ud af hvilken type felt skal oprettes

12. Forklar hvilke tre ting der tjekkes for i Property klassens metode getAction()

Metoden finder ud af om spilleren har lyst til at købe det givne felt, eller ejer feltet, eller om en anden spiller ejer feltet

13. I Property klassens setMessages() metode initialiseres tre String variable. Hvad er forklaringen på at den sidste initialisering er pakket ind i en try/catch?

Den er til for hvis et felt ikke er ejet af nogen

14. Hvilken effekt har kaldet super() i Property konstruktoren?

Det betyder at den henter data fra Field som property extends fra