

Eksamensopgave

Objektorienteret Programmering og Design (OOPD)

2013–2014, blok 2

Jon Sparring

1 Individuel skriftlig eksamensopgave

Denne opgave omhandler simulering af rovdyr og byttedyr. Opgaven skal løses individuelt og afleveres skriftligt i Absalon. Kun studerende, som er indstillet til eksamen i OOPD, vil blive bedømt. En indstilling til eksamen gælder for de 3 efterfølgende (r)eksaminer.

2 Opgave

Du skal skrive et program, som simulerer et lukket miljø, hvori der er 2 slags dyr: mus og ugle, som er hhv. bytte- og rovdyr. Uglerne spiser musene, musene kan formere sig, og der skal være sten, hvor musene kan gemme sig for uglerne.

Løsningen skal som minimum bestå af følgende klasser:

1. **Simulering**, som indeholder `main` metoden, der afvikler selve simuleringen. For **Simulering** skal gælde:
 - (a) Simuleringen skal foregå i skridt, således at alle ændringer sker samtidigt.
 - (b) Dyrene sover aldrig.
2. **Miljø**, som repræsenterer det lukkede miljø, hvori simuleringen foregår. For **Miljø** skal gælde:
 - (a) Miljøet skal være firkantet og inddelt i 20x20 felter.
 - (b) Simuleringens elementer kan kun befinde sig i felternes centrum.
 - (c) For et givent felt vil de tilstødende felter være de 8 nærmeste nabofelter, på nær langs kanten, hvor der vil være færre nabofelter.
3. **Sten**, som repræsenterer en sten. For **Sten** skal gælde:
 - (a) Der skal være 10 sten tilfældigt placeret i miljøet.
 - (b) Stenene ændrer sig ikke med tid.
4. **Mus**, som repræsenterer en mus. For **Mus** skal gælde:
 - (a) Simuleringen skal starte med 150 mus, tilfældigt placeret i miljøet.

- (b) Mus skal kunne bevæge sig tilfældigt fra felt til felt.
- (c) Mus bevæger sig med en hastighed på maksimalt 1 felt per skridt.
- (d) Mus har en synsradius på 1 felt (nabofelterne).
- (e) Med mindre en mus er blevet spist, dør den efter 20 skridt.
- (f) Der kan maksimalt være 2 mus på et felt.
- (g) Hvis der er 2 mus på et felt, og et af nabofelterne har plads til en mus, så er der 10% chance for at der opstår en ny mus på et ledigt nabofelt.
- (h) Hvis en mus ser en ugle, og musen ikke allerede har gemt sig, skal den flygte eller gemme sig under en ledig sten.
- (i) En sten kan kun beskytte 1 mus mod ugler.

5. **Ugle**, som repræsenterer en ugle. For **Ugle** skal gælde:

- (a) Simuleringen skal starte med 2 ugler.
- (b) Ugler skal afsøge miljøet for mus.
- (c) Ugler bevæger sig med en konstant hastighed på 1 felt per skridt.
- (d) Ugler har en synsradius på 2 felter (nabofelterne og nabofelternes nabofelter).
- (e) Ugler kan ikke dø og formerer sig ikke.
- (f) Der må højst være 1 ugle per felt.
- (g) Hvis en ugle ser en mus, skal den bevæge sig i retningen af musen.
- (h) Hvis en ugle lander på et felt med 1 mus uden sten eller på et felt med 2 mus, bliver 1 mus spist.

Programmet skal kunne vise miljøet for hvert skridt. Efter hvert skridt kan indholdet af et felt være:

- 0. tomt,
- 1. en sten,
- 2. en ugle,
- 3. en mus,
- 4. en sten og en ugle,
- 5. en sten og en mus,
- 6. en ugle og en mus,
- 7. to mus,
- 8. en sten, en ugle og en mus.
- 9. en sten og to mus.

Det anbefales, at miljøet illustreres som tekstfelter uden brug af grafisk brugergrænseflade, og man kan evt. benytte tallene 0–9 til at illustrere indholdet af hvert felt efter hvert skridt. Desuden skal der til slut vises en tabel over antallet af sten, ugler og mus per skridt i simuleringen.

3 Krav til afleveringen

Afleveringen skal bestå af:

1. Et antal Java-filer, som kan oversættes og køres uden yderligere pakker, og som løser opgaven beskrevet ovenfor. Java-filerne skal være dokumenterede ved hjælp af Javadoc kommentarer, og ud over simuleringen skal der implementeres unittests, der illustrerer kodens korrekthed. Unittests kan udføres vha. JUnit, men det er ikke et krav.
2. En pdf fil, som beskriver:
 - (a) Designet af programmet og de overvejelser, du har gjort dig om din løsning (maks. 1 side),
 - (b) en udskrift (skærmdump) af miljøet efter 10 skridt for en simulering,
 - (c) en udskrift af tabellen efter 50 skridt af samme simulering,
 - (d) en udskrift af kørslen af unittests,
 - (e) et afsnit, der kommenterer resultatet af samme simulering og unittests (maks 10 linjer).

Ved evaluering af din aflevering vil vi lægge særlig vægt på om

1. programmet løser opgaven,
2. der er valgt nogle hensigtsmæssige programmeringsmetoder og strukturer,
3. programteksten er opsat på en overskuelig måde,
4. du har valgt hensigtsmæssige navne til variabler, metoder og klasser,
5. der er fyldestgørende kommentarer uden at programmet drukner i kommentarer,
6. dine overvejelser er fornuftige i forhold til opgaven,
7. strukturen på dit ind- og uddata er hensigtsmæssig,
8. du har implementeret fornuftige unittests.

Husk at det er vigtigt at prioritere din tid. Denne opgave er berammet til 20 timer, og langt fra alle vil have tid til at opfylde alle opgavens krav. Det er derfor vigtigt at:

1. En delvis aflevering er bedre end ingen aflevering
2. En aflevering bestående af en kørende simulering uden alle delelementer er bedre end en, hvor programmet slet ikke fungerer.
3. Denne opgave afprøver dine programmeringsevner, ikke dine evner inden for visuelt design.
4. Det er vigtigt at aflevere BÅDE java-filer og pdf-fil, da det ellers bliver svært for lærere og censorer at give en samlet vurdering af opgaven.