2de herkansing OOP2 Make IT Work  
Cohort 6

DATUM INVULLEN

|  |  |
| --- | --- |
| Naam |  |
| Studentnr. |  |
| Inlevertijd (door surveillant in te vullen |  |

# Opmerkingen

1. Je moet voor deze opgave code toevoegen aan een bestaand startproject. Download de zip van het startproject van Moodle.
2. Upload na afloop je hele Eclipse project naar Moodle.
3. Deze toets bestaat uit 3 pagina’s en is enkelzijdig afgedrukt. Vergeet de achterkant niet!
4. Het gebruik van het internet voor andere doeleinden dan het raadplegen van het e-boek of het uploaden van de oplossing naar Moodle is *niet toegestaan*.
5. Uitsluitend toegestaan is het gebruik van het boek, je eigen aantekeningen en al het materiaal op je eigen laptop (inclusief uitwerkingen van practicumopgaven).
6. Na afloop van het tentamen dient dit opgaveblad bij de surveillant te worden ingeleverd. Uitwerkingen in Moodle waarbij geen bijbehorend opgaveblad is ingeleverd kunnen niet worden nagekeken!
7. Als je het tentamen inlevert checkt de docent of de upload in Moodle is gelukt. Verlaat het lokaal niet voordat die check is uitgevoerd en je toestemming hebt gekregen!

# Stap 0 (0 punten)

Download het project en importeer het in Eclipse (Zie het document “Importeren en exporteren van Eclipse projecten” op Moodle). Run het een keer. In het begin wordt de string “Dit is de toets van Naam 50071234.” afgedrukt. Wijzig die boodschap zo dat je eigen naam en studentnummer worden afgedrukt. Wijzig ook de naam van het project zodat je eigen naam en studentnummer er in voorkomt.

# Stap 1 (25 punten)

## Inleiding

Deze opdracht gaat over abstracte klassen en interfaces. Maak deze opgave in het package stap1 van het startproject. Dit package is nu nog leeg omdat je dit helemaal zelf moet programmeren.

1. Maak een interface met de naam Repeatable die maar 1 methode heeft:void repeat(int times);
2. Maak een abstracte klasse met de naam Work die de interface Repeatable implementeert.
   1. Maak in Work een methode abstract void doIt();
   2. Maak in Work een implementatie van de methode repeat die de methode doIt zo vaak aan roept als door de parameter times wordt aangegeven.
3. Maak een klasse Exam die een subklasse is van Work.
   1. Geef Exam een private attribuut van type String genaamd course.
   2. Geef Exam een constructor met een parameter course, en sla de waarde op in het attribuut course.
   3. Implementeer de methode doIt en geef deze een body die de volgende tekst in beeld print waarbij je voor X de course invult: “Taking the X exam…”.
4. Maak een klasse Main met een main methode. Test in deze klasse of de implementatie werkt met de volgende stappen.
   1. Declareer in Main een Repeatable genaamd exam.
   2. Construeer een nieuw Exam object met als argument “OOP2”, en ken dit toe aan exam.
   3. Roep op het exam object de methode repeat aan met als argument 3.

# Stap 2. (30 punten)

## Inleiding

In het startproject is in package een klasse StringSplitter gedefinieerd die twee statische methoden definieert: first en last. Beide methoden hebben een String parameter en geven een String als resultaat terug. Deze methoden knippen een string in twee delen van (ongeveer) gelijke lengte waarvoor geldt dat   
first(str) + last(str) == str.

Het project bevat ook een klasse Recursion met een main methode en een paar hulpfuncties. Daarnaast zijn er in deze klasse twee methoden gedefinieerd die je zelf moet afmaken, nl. invert en isPalindroom.

In de main methode worden deze methoden aangeroepen om je werk te testen. Aan main hoef je niets te veranderen.

## Opgave 2.1 (20 punten)

De methode private String invert(String str) is een recursieve methode om een string om te keren. Dus invert(“abcde”) is “edcba”.

Maak de methode invert af waarbij je gebruik maakt van recursie en de methoden StringSplitter.first en StringSplitter.last.

**Opgave 2.2 (10 punten)**

Een *palindroom* is een string die van links naar rechts gelezen hetzelfde is als van rechts naar links. Dus “lepel” en “meetsysteem” zijn palindromen maar “tentamen” niet.

De klasse Recursion bevat ook een methode private boolean isPalindroom(String str) die true oplevert als str een palindroom is, anders false. Maak deze methode af, waarbij je gebruik maakt van de methode invert uit de vorige opgave.

# Stap 3 (20 punten)

Maak unit tests voor de methoden die je in de vorige stap hebt gemaakt. Test daarin zowel de methode invert als de methode isPalindroom. Maak voor ieder methode minimaal drie tests, waarvan er één de lege string als testgeval gebruikt. Test ook andere bijzondere gevallen (wel/geen palindroom, strings van lengte 1 en 2…) verzin zelf ook andere testgevallen. Maak de tests met behulp van de unit test hulpmiddelen van Eclipse en gebruik een aparte source folder voor de testcode.

# Stap 4 (20 punten)

In het startproject zit een file met de naam artis.dat. Die file bevat in binair formaat instanties van de klasse Dier die in het package stap4 is gedefinieerd. Maak een klasse Dierentuin in package stap4 die deze file leest en daarna:

1. Een regel afdrukt met de tekst: “De file bevat <n> dieren”, waarbij <n> het aantal instanties is van Dier dat je in de file hebt gevonden.
2. Druk daarna iedere instantie van Dier die je hebt gelezen af.
3. Maak ook een main methode in Dier zodat die klasse getest kan worden.

## Beoordeling

|  |  |
| --- | --- |
| Stap 0: | 0 |
| Stap 1: | 25 |
| Stap 2: | 30 |
| Stap 3: | 20 |
| Stap 4: | 20 |
| Code conventions: | 5 |
| **Totaal** | **100** |