Sarajevo School of Science and Technology
CS/IS 160 Programming and Problem Solving II
May 2015

Test 3 Practice Problems

(5% of the final grade, 90 minutes)

Problem 1: Short answers (20 points).

Answer the following questions (each part 2 points):

(a) What is a default constructor?

(b) What is an interface?

(c) What is the difference between the `throws` clause and the `throw` statement?

(d) What is the difference between a checked exception and an unchecked exception?

State whether each of the statements that follows are *true* or *false*. If *false*, explain why.

(e) A method declared *final* in a superclass can be overridden in a subclass.

(f) An *abstract* class normally contains of more than one *abstract* method.

(g) The superclass constructor always executes before the subclass constructor.

Write the following simple declarations and code (each part 3 points):

(h) Declare and initialize an array list variable `intList` that can dynamically hold integer numbers.

(i) Write a "bad" code segment that generates an `ArithmeticException`.

Problem 2: Diagonal entries of a square matrix (20 points).

Implement a static method called `diagonal`, which returns the diagonal elements of an input square integer matrix. For example, if the input matrix is

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 3 \end{bmatrix},$$

then the diagonal elements are returned as the array (1, 1, 3).

Use the following method prototype, where `matrix` is the input square matrix with integer entries:

```
public static int[] diagonal( int[][] matrix )
```

Problem 3: Count duplicate elements in an array (20 points).

Implement a static method `countDuplicates` that returns the number of duplicate elements in an integer array. For example, if the input array is (1, 2, 2, 3, 5, 1), then this method returns 2, since there are two duplicate elements (1, 2) in the array.

Use the following method prototype, where `array` is an input integer array:

```
public static int countDuplicates( int[] array )
```

Problem 4: Abstract and concrete classes (20 points, each part 10 points).

(a) Define an abstract class `Animal` with a private instance variable `age` (integer) and an abstract method `sound` that has no arguments and returns a string.

Implement a getter and setter for the age variable and make sure that it must be set to a positive integer value. If that is not the case, throw a `RuntimeException` with an appropriate message.

Therefore, the `Animal` class should have the following method signatures:

```
public int getAge()
public void setAge(int age) throws RuntimeException
abstract String sound()
```

(b) Next define two concrete classes that extend the `Animal` class: `Dog` and `Duck`. Implement their `sound` method to return an appropriate sound specific to that species.

Problem 5: Inheritance and interfaces (20 points).

Implement the class hierarchy shown below, where the `Actor` is an interface and the `Comedian` and `ActionHero` classes implement the interface. The `act` method for the `Comedian` class should return the string "I'm Austin Powers," while the `ActionHero` class should return "My name is Bond, James Bond."