

CS180: Problem Solving and Object Oriented Programming

Recitation Week 2

Department of Computer Science, Purdue University

January 17, 2011

Outline

- 1 **Basic Java program constructs**
- 2 **Common Errors**

Deer Population in Tippecanoe county



Deer Population Calculator

Need to consider the following factors:

- Current-Year Population
- Percentage of Does
- Number of Fawns per Doe

Question:

- What is the Next-Year Deer Population?
- After a given number of years, what is the Deer Population?

Deer Population Calculator (cont.)

Given:

- Current-Year Population, denoted as x
- Percentage of Does, denoted as r
- Number of Fawns per Doe, denoted as p

Answer:

- Next-Year Deer Population :
$$x + x * r * p = x(1 + rp)$$
- 2nd-Year Deer Popoluation :
$$x + x * r * p + (x + x * r * p) * r * p = x(1 + rp)^2$$
- yth-Year Deer Popoluation : $x(1 + rp)^y$

Deer Population Calculator - Revisited

Need to consider more factors:

- Current-Year Population
- Percentage of Does
- Number of Fawns per Doe
- **Death Rate**
- **Maximum population supported by the ecosystem**
- **Assume that if population crosses maximum, it will start falling by 20% annually**

Question:

- What is the Next-Year Deer Population?
- After a given number of years, what is the Deer Population?

Deer Population Calculator - Revisited(cont.)

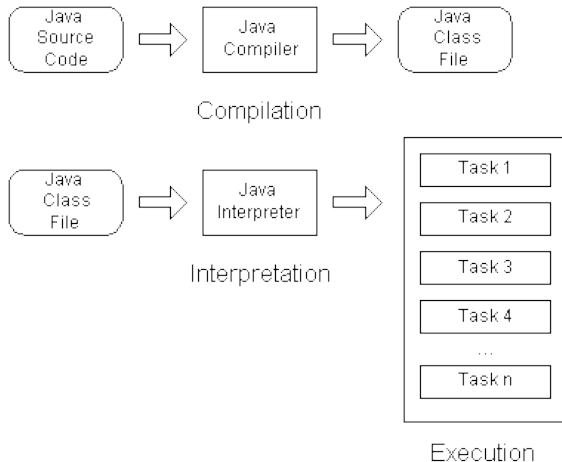
Given:

- Current-Year Population, denoted as x
- Percentage of Does, denoted as r
- Number of Fawns per Doe, denoted as p
- Death Rate, denoted as d
- Maximum population supported by the ecosystem, denoted as M

Answer:

- Next-Year Deer Population : Is simply
$$x + x * r * p - x * d = x(1 + rp - d)?$$
- 2nd-Year Deer Population : ?
- yth-Year Deer Population : ?

Java Compilation and Interpretation



Syntax Error

- Occurs during Compilation step.
- Can be discovered by *javac* command
- Example:
 - Compile *SyntaxError.java* with:
javac week1/SyntaxError.java
 - Result:

```
week1/SyntaxError.java:16: incompatible types
found   int
required: boolean
    if (x = 2) {
1 error
```

Runtime Error

- Occurs during Interpretation step.
- Can only be discovered when running Java program with *java* command
- There are many different types of Runtime Error, to name just a few: Divided by Zero, Infinite Loop, etc.

Runtime Error - Divided by Zero

- Compile *RuntimeError.java* with:
javac week1/RuntimeError.java
- Run: **java week1.RuntimeError**
- Result:

```
This program computes the division of two integer number a/b
Please input a (press Enter when finish):10
Please input b (press Enter when finish):0
Exception in thread "main" java.lang.ArithmeticException: / by zero
    at week1.RuntimeError.main(RuntimeError.java:25)
```

Runtime Error - Infinite Loop

- Compile *RuntimeError.java* with:
javac week1/LogicalError.java
- Run: **java week1.LogicalError**
- Result:

```
Balance after 56586 years: Infinity
Balance after 56587 years: Infinity
Balance after 56588 years: Infinity
Balance after 56589 years: Infinity
Balance after 56590 years: Infinity
Balance after 56591 years: Infinity
Balance after 56592 years: Infinity
Balance after 56593 years: Infinity
Balance after 56594 years: Infinity
Balance after 56595 years: Infinity
Balance after 56596 years: Infinity
Balance after 56597 years: Infinity
Balance after 56598 years: Infinity
```