CSCI1530 Computer Principles and Java Programming

Tutorial 9
Declaring new classes

Content

- Declare new classes
  - Student class
  - Octopus Card

Declare new classes

- For java program, usually one file contains only one class

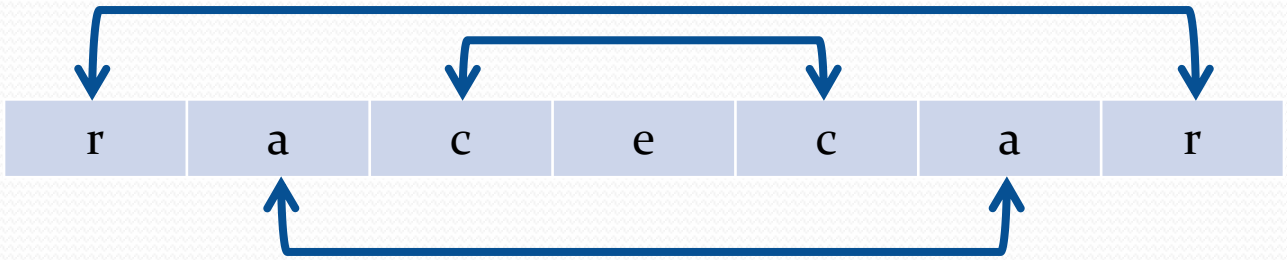
- Java Rule
  - A public class must be implemented in a file with the same name as the class
  - Advanced: you could define more than one private class in that file

Task on a new class (Example)

- Declare a class called "Student" and a method to check if the student name is a palindrome

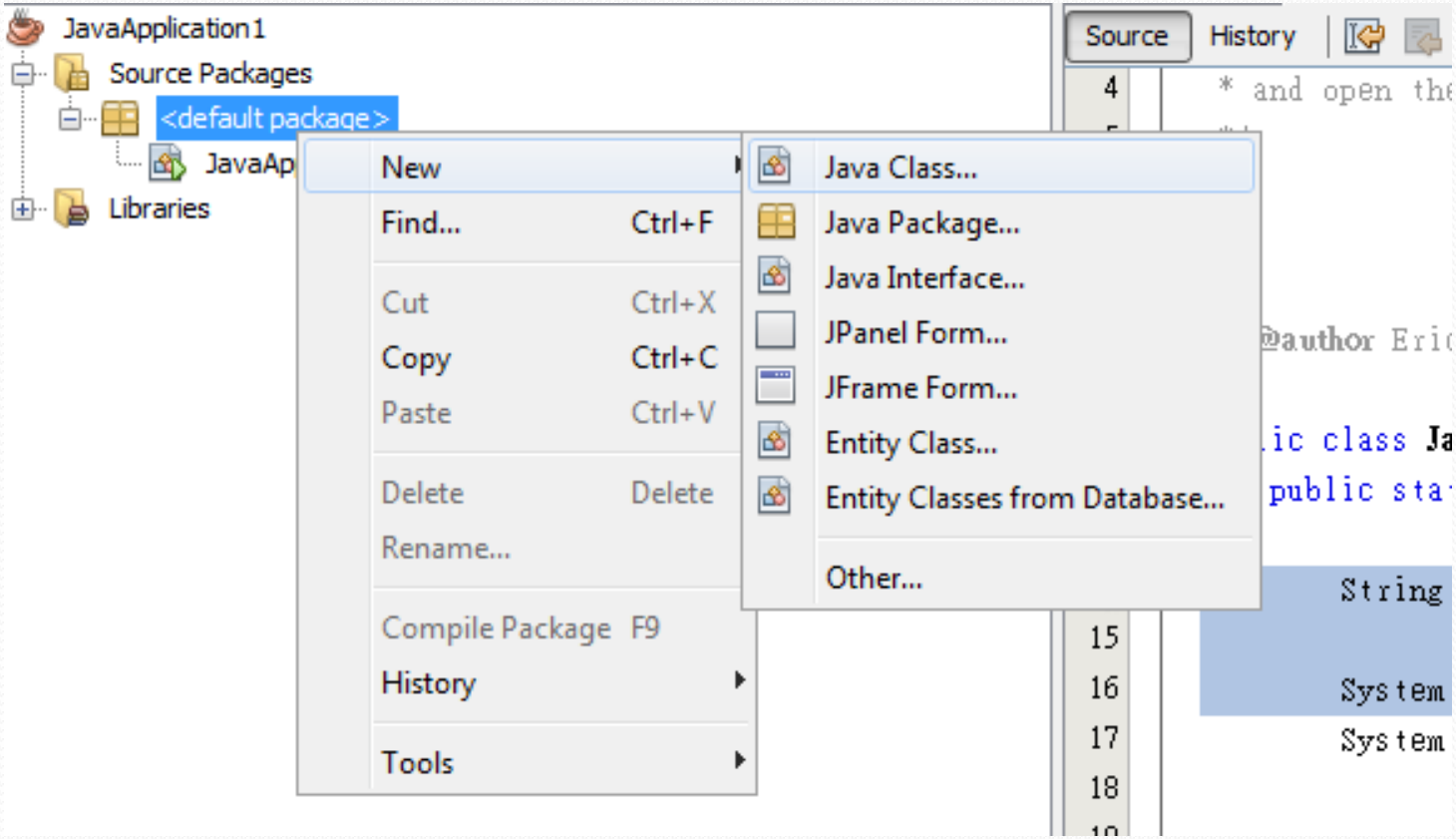
- Palindrome
  - The string remains unchanged after reverse
  - Example

"racecar"



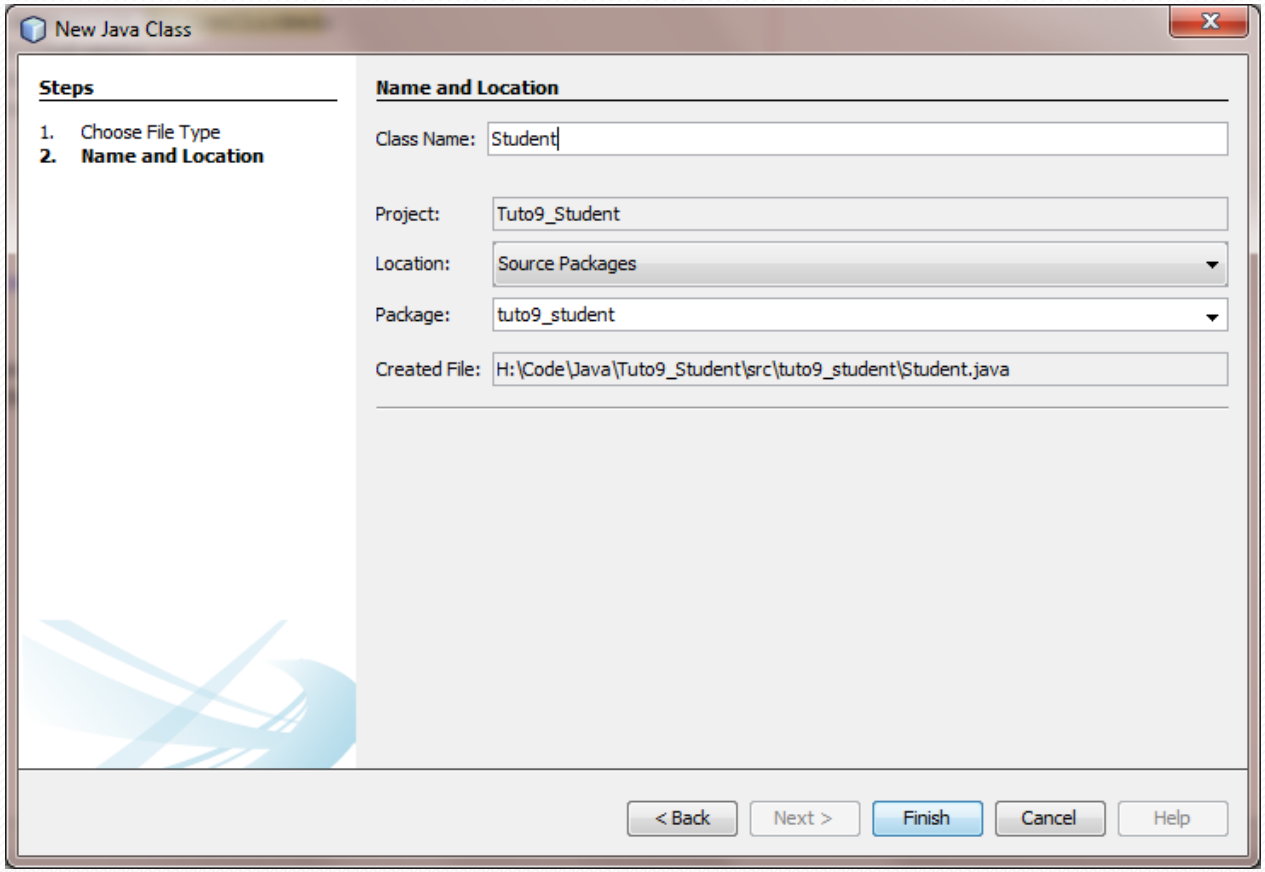
Declare Student class

- Add a new class in your project



Declare Student class

- Give the class name -> Finish



Student example

- Create constructor and method

```
public class Student {
    private String name;
    public Student(String s){
        this.name = s;
    }
    @Override
    public String toString(){
        return "Student name: "+this.name;
    }

    public boolean isPalindrome(){
        for (int i = 0; i < this.name.length()/2; i++) {
            int lPos = i;
            int rPos = (this.name.length()-1) - i;
            if(this.name.charAt(lPos) != this.name.charAt(rPos))
                return false;
        }
        return true;
    }
}
```

Declaring the field

Reference to current object

String of your object

File: *Student.java*

Test your class

```
public static void main(String[] args) {
    Student stu = new Student("alice");
    System.out.println(stu);
    System.out.println(stu.isPalindrome());

    stu = new Student("otto");
    System.out.println(stu);
    System.out.println(stu.isPalindrome());
}
```

Executed `stu.toString()`

File:Tutorial09.*java*

```
Student name: alice
false
Student name: otto
true
```

Class exercise

- Based on student's `name`, create 2 methods to implement `startsWith()` and `endsWith()`
  - Student.nameStartsWith()
  - Student.nameEndsWith()

  - Use `String.indexOf()` & `String.length()`

Octopus Class

- A Java program to emulate Octopus card operations
- which can
  - Store card id
  - Add & use card balance
  - Copy the card for backup purpose

Class skeleton

```
public class Octopus {
    private String id; // Card id
    private double balance;

    public Octopus(String id){...}

    public String toString(){...}
    public Octopus copyCard(){...}
    public void addValue(double value){...}
    public void useValue(double value){...}
}
```

Fields

Constructor

Methods

Octopus.java

Constructor method

```
public Octopus(String id){
    this.id = id;
    this.balance = 50.0; // Default with some balance
}
```

- Store card id
- Set initial balance as $50

Show card id && balance

```
public String toString(){
    String text = "Card id: "+this.id+"\n";
    text += String.format("Current balance: $%.2f\n", this.balance);
    return text;
}
```

- Add `id`, `balance` first
- Then add the `transaction` in single line

```
Card id: Tom
Current balance: $413.60
```

Sample output

Copy the card

```
public Octopus copyCard(){
    // Copy card id
    Octopus newCard = new Octopus(this.id);

    // Copy card balance
    newCard.balance = this.balance;

    return newCard;
}
```

- "new" a Octopus object inside a method

Add & use card balance

- Update the balance

```
public void addValue(double value){
    this.balance += value;
}
```

```
public void useValue(double value){
    if(value > this.balance){
        System.out.println("Transaction aborted for negative balance.");
        return;
    }

    this.balance -= value;
}
```

Test an Octopus class

- Unzip and open tuto9_Octopus.zip in NetBeans
- Run the project/OctopusTest.java

Q & A

- Tutorial
- Assignment 4
  - import package

In "Animation.java"

```
import java.awt.Color;
import java.awt.GridLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
```

In "ButtonTracker.java"

```
import static animation.Animation.buttonClicked;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
```