Content Objective: Students will begin to work with abstract classes and methods while extending interfaces in Java.

|  |  |
| --- | --- |
| **On the Tech Horizon (10pts.)**  **link to a tech/coding related article or journal no more than one month old (no blogs or reddit clones see below)** | |
| URL: | http://www.javaworld.com/article/3011961/java-language/java-9-delayed-by-slow-progress-on-modularization.html |
| Reaction/Commentary: | There was a planned release for Java 9 in September 2016, but it is being postponed by six months until March 2017 so that the maintainers of the essential build tools and IDEs have enough to time to implement a good support for the modular development. |

|  |  |
| --- | --- |
| **Tech Terms and History (20pts.)**  **vocabulary from BJ p.463-512 and The Information Chapter 8 (definition/commentary/significance in your words)** | |
| Syntax to create interface | BJ p.465  publicinterface *InterfaceName*  {  *method headers*  } |
| Syntax to implement interface | BJ p.466-467  public class *ClassName* implements *InterfaceName, InterfaceName, . . .*  {  *instance variables*  *methods*  } |
| Why can an interface NOT be instantiated? | BJ p.470  You cannot construct an object of an interface type. Interfaces aren’t classes. |
| Syntax for constants in Java | BJ p.470  All variables in an interface automatically have the *public static final*  int NORTH = 1;  int SOUTH = 2; |
| Syntax to convert class to interface (2 ways) | BJ p.471  You can convert from a class type to an interface type, provided the class implements the interface.  EX 1:  Country uruguay = new Country("Uruguay", 176220);  Measurable meas = uruguay;  EX 2:  double averageBalance = Data.average(accounts); |
| Syntax to cast interface to class | BJ p.472  public static Measurable larger(Measurable obj1, Measurable obj2) {  if(obj1.getMeasure() > obj2.getMeasure()) {  return obj1;  } else {  return obj2;  }  } |
| Syntax for clone Method | BJ p.475  BankAccount clonedAccount = (BankAccount) account.clone(); |
| Syntax for cloneable Interface | BJ p.475  public class Object {  protected Object clone() throws CloneNotSupportedException {  if (this instanceof Cloneable) {  // Copy the instance variables  } else {  throw new CloneNotSupportedException();  }  }  } |
| What is a Callback | BJ p.477-478  Leads to a more flexible average method and how a callback can be implemented using interface types. |
| What is an “inner class” and when/how would you use it? | BJ p.481-482  A class within another class. You would use this when a specific method needs an implementation or etc. |
| What is an “anonymous class” and when/how would you use it? | BJ p.482  EX.  Replace:  Country belgium = new Country(“Belgium”, 30510);  countries.add(belgium);  with:  countries.add(new Country(“Belgium”, 30510);  This can only be done is Country has not used elsewhere in the same method. |
| What is a “mock object” and when/how would you use it? | BJ p.483  Where you create an object without creating a fully functional object class.  EX.  this:  public void addScroe(int studenId, double score)  public double getAverageScore(int studentId)  public void save(String filename)  turns into this:  public interface IGradeBook {  void addScroe(int studenId, double score);  double getAverageScore(int studentId);  void save(String filename);  . . .  } |
| What is an event listener? | BJ p.485  Belongs to a class that is provided by the application programmer. Its methods describe the actions to be taken when an event occurs. |
| Syntax for creating an inner class listener | BJ p.487  JButton button = new JButton(“. . .”);  // This inner class is declared in the same method as the button variable  class MyListener implements ActionListener {  . . .  };  Action Listener listener = new MyListener();  button.addActionListener(listener); |
| Syntax for mousePress() | BJ p.498  class MousePressListener implements MouseListener {  public void mousePressed(MouseEvent event) {  int x - event.getX();  int y = event.getY();  component.moveRectangleTo(x, y);  }    // Do-nothing methods  public void mouseReleased(MouseEvent event) {}  public void mouseClicked(MouseEvent event) {}  public void mouseEntered(MouseEvent event) {}  public void mouseExited(MouseEvent event) {}  } |
| Summarize Turing’s downfall in 50-100 words. | TI p.254-256  Turing designed an amazing computer that could imitate certain things. This was amazing for his time in the 1950s. Or at least that is what he wanted us to believe, but the computer was made up of such a complicated language that it was impossible to prove or disprove. However, he was convicted of homosexuality and was basically stripped of his manhood and security. He later committed suicide. |
| Summarize Shannon’s advancements through the 50’s and 60’s in 50-100 words. | TI p.259-268 |

|  |  |
| --- | --- |
| **Code Snippets (30pts.)**  **only submit snippets or classes no full programs required (test and run in IDE, then copy/paste applicable code frag)** | |
| E10.3-10.4 | E10.3  public class Persons implements Measurable {  private String name;  private double height;    public Persons(String name, double height) {  this.name = name;  this.height = height;  }    public double getHeight() {  return height;  }    public String getName() {  return name;  }    public double getMeasure() {  return height;  }  }  E10.4  public static Measurable max(Measurable[] objects) {  if (objects.length == 0) {  return null;  }  Measurable max = objects[0];  for (Measurable obj : objects) {  if (obj.getMeasure() > max.getMeasure()) {  max = obj;  }  }  return max;  } |
| E10.9 |  |
| E10.10-E10.11 |  |
| E10.13-E10.17 | E10.13  public class ClickListener implements ActionListener  {  int n = 0;    public void actionPerformed(ActionEvent event)  {  if (true) {  n++;  }    if (n==1) {  System.out.println("I was clicked " + n + " time");  } else {  System.out.println("I was clicked " + n + " times");  }  }  }  E10.14  public class ButtonViewer  {  private static final int FRAME\_WIDTH = 300;  private static final int FRAME\_HEIGHT = 100;  public static void main(String[] args)  {  JFrame frame = new JFrame();  JButton button1 = new JButton("Button 1");  button1.setSize(100,100);  JButton button2 = new JButton("Button 2");  button2.setSize(100,100);  frame.add(button1);  frame.add(button2);    ActionListener listener1 = new ClickListener();  ActionListener listener2 = new ClickListener();  button1.addActionListener(listener1);  button2.addActionListener(listener2);  frame.setSize(FRAME\_WIDTH, FRAME\_HEIGHT);  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  frame.setVisible(true);  }  }  E10.15 |
| E10.21-E10.22 |  |

|  |  |
| --- | --- |
| **Code Challenge (30pts.)**  **full functioning application sent to** [**GitHub**](https://github.com/SkylineHigh/CSAdvanced/tree/master/06%20Abstract%20Data%20Types) | |
| You may choose any one of the following code challenges: P10.8-P10.21 and  post code to GitHub under “Abstract Data Types” | |
| Notes: | [Link to CarTester.java file](CSAdvanced/KRISTOPHER-THE-AWESOME-FILE/CHAPTER-10/CAR/CarTester.java) |

|  |  |
| --- | --- |
| **Badge Progress (10pts.)**  **building your coding profile: Java coding training site to earn badges (recommended site** [**http://coderbyte.com**](http://coderbyte.com) **)** | |
| Screenshot/URL: |  |
| Notes/Issues: |  |

|  |  |
| --- | --- |
| **Notes**  **your notes** | |
| Notes: |  |