УНИВЕРСИТЕТ ИТМО
КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Отчёт по лабораторной работе №6
Вариант 401

Выполнил Ощепков Артём,
группа p3102
Преподаватель Письмак А. Е.

Санкт-Петербург 2017

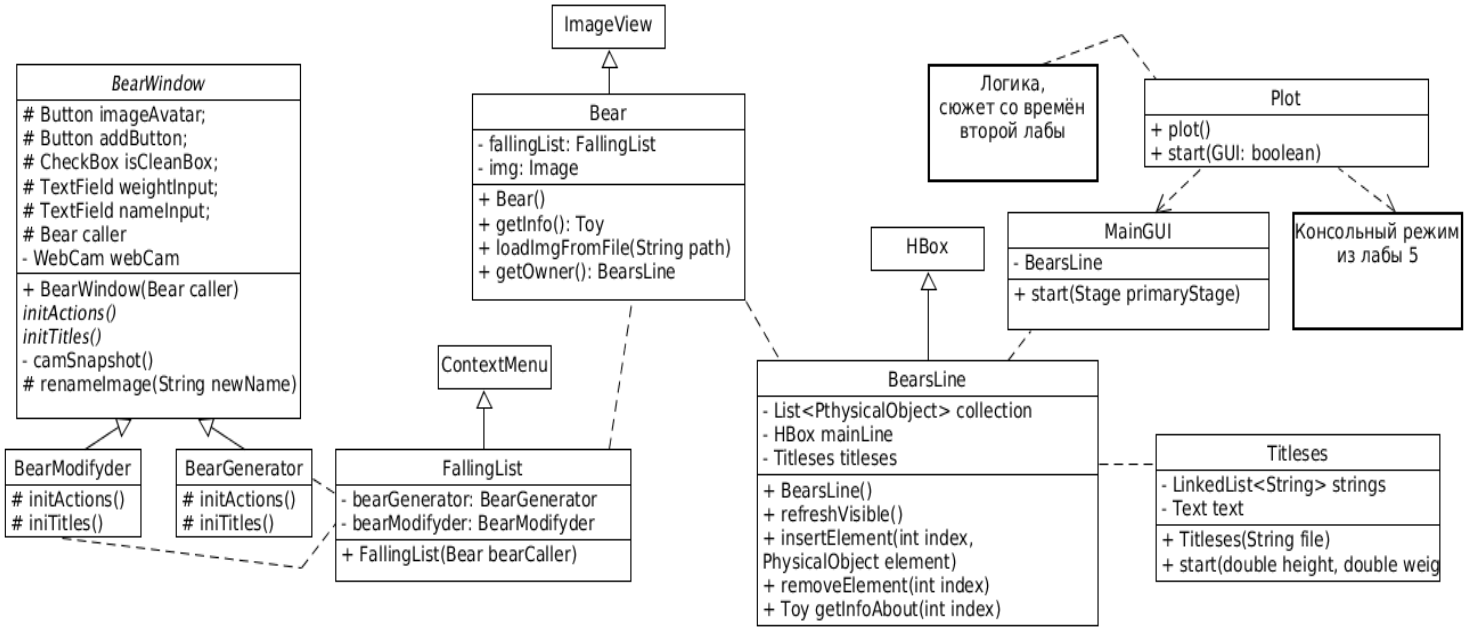
Текст задания

Доработать программу из лабораторной работы №5, реализовав на её основе многопоточное приложение с графическим пользовательским интерфейсом (GUI), , которое реализует управление коллекцией объектов. GUI должен быть построен с использованием библиотеки **JavaFX**.

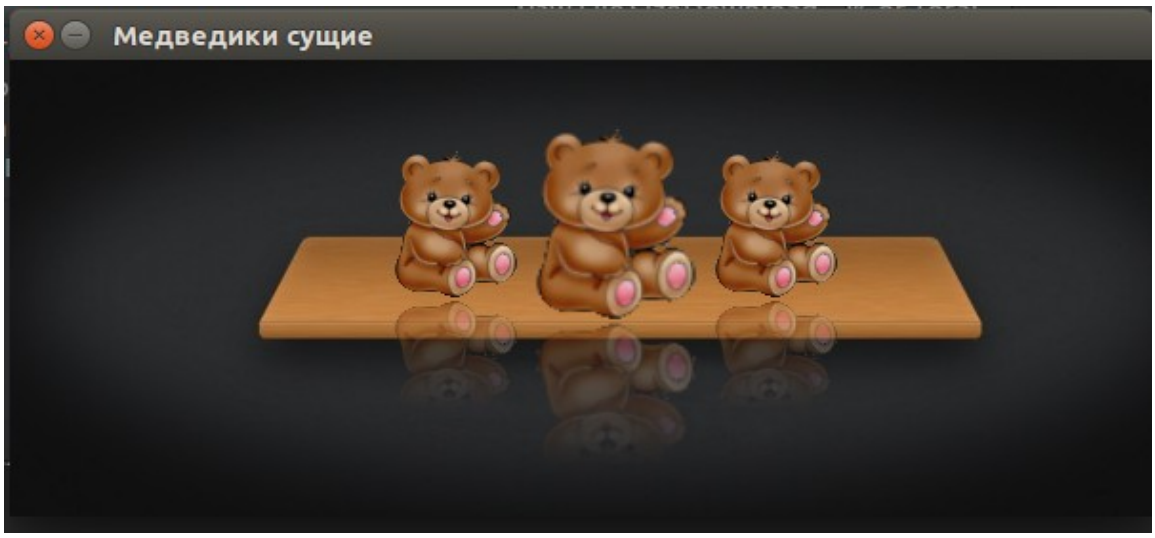
Получившаяся в итоге программа должна удовлетворять следующим требованиям:

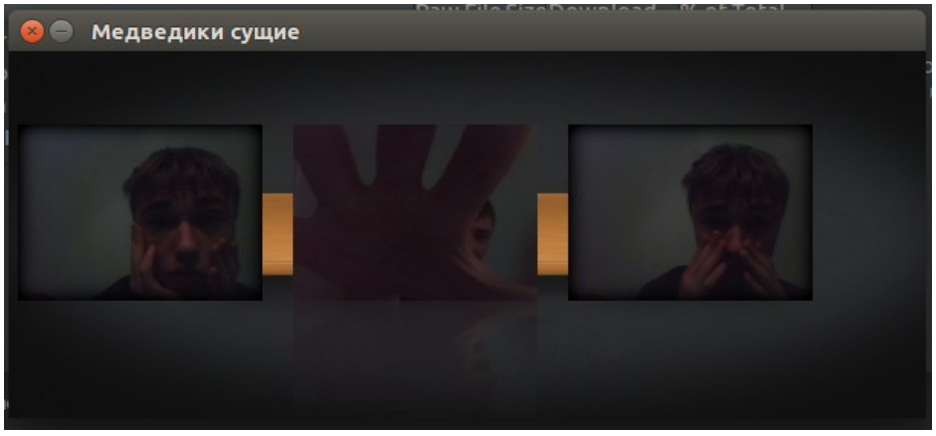
1. Приложение должно содержать CRUD-интерфейс, позволяющий пользователю взаимодействовать со списком объектов в интерактивном режиме.
2. Для вывода списка объектов должен использоваться компонент (виджет) `TreeTableView`.
3. Необходимо реализовать возможность сортировки и фильтрации значений в таблице по каждому из столбцов.
4. Интерфейс приложения должен использовать layout `HBox`, и содержать, как минимум, следующие компоненты (виджеты):
    - `Button`
    - `HTMLEditor`
    - `RadioButton`
    - `ProgressBar`
5. Все действия над коллекцией, реализованные в предыдущей лабораторной работе, в новой версии программы должны быть реализованы в виде отдельных кнопок, полей ввода или пунктов меню в интерфейсе (по согласованию с преподавателем).
6. При запуске приложения коллекция должна автоматически заполняться значениями из файла. Все операции файлового ввода / вывода должны выполняться в отдельных потоках.

Диаграмма классов программы



Скриншоты графического интефрейса





Исходный код:

```
public class Bear extends javafx.scene.image.ImageView {
  private FallingList fallingList;
  private Image img;
Bear(String imgPath){
    super();
    img = new Image(getClass().getResourceAsStream("defaultBear.png"));
    this.setImage(img);
    initAnim();
  }
  void initAnim(){
    ScaleTransition scaling = new ScaleTransition();
    scaling.setDuration(Duration.valueOf("0.25s"));
    scaling.setNode(this);
    setOnMouseEntered(e->scaling.play());
    this.setEffect(new Reflection());
    setOnContextMenuRequested(e->{
      if(fallingList == null)
        fallingList = new FallingList(this);
      fallingList.show(this,e.getScreenX(),e.getScreenY());
    });
  }
  Toy getInfo(){
    if(this.getId() == null)
      return null;
    else // Модель медведя принадлежит главному боксу BearsLine, который сам является
боксом
      return
((BearsLine)this.getParent().getParent()).getInfoAbout(Integer.valueOf(this.getId()));
  }
  public BearsLine getOwner(){
    if( this.getParent() != null && (this.getParent().getParent() instanceof  BearsLine))
      return ((BearsLine) (this.getParent().getParent()));
    else
      return null;
  }
}
public abstract class BearWindow {
  // Нужно знать, какой медведь нас вызвал
  protected Bear caller;
  BearWindow(Bear caller){
    this.caller = caller;
    defaultImage = new Image(getClass().getResourceAsStream("defaultBear.png"));
    imageAvatar.setOnAction(e-> {
          if(!camSnapshot.isAlive())
            camSnapshot = new Thread(new Runnable() {
              @Override
              public void run() {
                // Фотография с камеры устанавливается на кнопку
                camSnapshot();
              }
```

```
            });
        camSnapshot.start();
      }
    );

    addButton = new Button("Сотворить медведя!");
    addButton.setOnAction(e->{
      stage.close();
    });

    initTitles();
    initActions();
  }
  abstract protected void initActions();
  abstract protected void initTitles();
  // Класс, поставляемый библиотекой
  private Webcam webcam;
  // Довольно долго записать в файл и получить изображение, чтобы окно не висло,
сделаем отдельный поток.
  private Thread camSnapshot;
  // Вызывается потоком camSnapshot
  private void camSnapshot(){
    try {
      if(webcam == null)
        webcam = Webcam.getDefault();
      webcam.open();
      ImageIO.write(webcam.getImage(), "PNG", new File(
              System.getProperty("user.dir") + File.separator + "things" + File.separator +
"temp"
          )
      );
      webcam.close();
      imageView.setImage(new Image("file:" +
          System.getProperty("user.dir") + File.separator + "things" + File.separator + "temp" )
      );
    }
    catch (java.io.IOException e){
      System.out.println("Беда с файлами.");
    }
  }
  /**
    Камера сохраняет в темп-файл.
    При закрытии окна нужно определиться,
    к какому медведю относится данное изображение
   */
  protected void renameImage(String newName){
    File file = new File(System.getProperty("user.dir") + File.separator + "things" +
File.separator + "temp");
    if(file.exists())
      file.renameTo(new File(System.getProperty("user.dir") + File.separator + "things" +
File.separator + newName));
  }
```

```
}
public class BearsLine extends HBox{
   private List<PhysicalObject> collection;  // Зависим от абстракции
   private HBox mainLine;  // Полоска медведей
   private Titleses titleses; // Стихотворение
   private static String collectionXMLFile = System.getProperty("user.dir") +
                        + File.separator + "BabykAndMotherThings.xml";
   public BearsLine(){ // По-умолчанию - загружаем коллекцию из файла
      Home loader = new Home(); // Возможно, следовало бы добавить нечто вроде util.
     loader.loadThingsFromFile(collectionXMLFile);
     collection = loader.getThings();
     refreshVisible();
     // Яков рассказывал, как его друга на экзамене попросили доказать, что замыкание
замкнуто
     class MouseDraggedWrapper{ //Вложенный, однако
       public double value;
     }
      MouseDraggedWrapper mouseDragged = new MouseDraggedWrapper();
     this.setOnMousePressed(start->      // Ловим жест
       mouseDragged.value = start.getSceneX(););
     this.setOnMouseReleased(end->{
       translateTransition.setFromX(mainLine.getTranslateX());
       translateTransition.setToX(
            mainLine.getTranslateX() + end.getSceneX() - mouseDragged.value);
       translateTransition.play();
     });
   }
   // Отображение коллекции в видимых медведей
   public void refreshVisible() {
     mainLine.getChildren().clear();
     for (int i = 0; i < collection.size(); i++) {
       Bear bear = new Bear(); // Спасибо огромное составителям JavaFX за возможность
присудить ID!
       bear.setId(Integer.toString(i));
       bear.loadImgFromFile(System.getProperty("user.dir") +
            File.separator + "things" + File.separator +
            Integer.toString(collection.get(i).hashCode()));
       mainLine.getChildren().add(bear);
       if(!collection.get(i).isClean())
         bear.setEffect(new InnerShadow());
     }
   }
   public Toy getInfoAbout(int index){
     return index < collection.size() ? (Toy)collection.get(index) : null;
   }
}
```

Вывод:

ЫІ