НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет Автоматики и Вычислительной Техники

Кафедра Вычислительной Техники

**ЛАБОРАТОРНАЯ РАБОТА №3**

по дисциплине «Технология программирования»

*Основы программирования на Java. Обработка событий. Механизм делегирования событий.*

**Группа:** АВТ-710

**Студент:** Шелудько В.

Бурдинский И.

**Вариант:** 8

**Преподаватель:** Васюткина И.А.

# Цели работы:

1. Изучить особенности реализации классов-коллекций в Java.
2. Доработать программу, созданную в лабораторной работе № 2:
3. добавить генерируемым объектам понятия «время рождения» и «время жизни». Время рождения устанавливается в момент генерации объекта и по значению соответствует времени, прошедшему от начала симуляции. Время жизни – время, через которое объект должен исчезнуть, считая от времени рождения;
4. вынести установку параметров времени жизни объектов в пользовательский интерфейс. Для каждого типа объекта должно задаваться собственное время. Рекомендуется использовать текстовые поля, но следуют помнить о проверке на ввод некорректных данных;
5. добавить генерируемым объектам уникальные целочисленные идентификаторы (случайные числа), которые назначаются при генерации объекта. Для хранения сгенерированных идентификаторов используйте коллекцию удобную для поиска по варианту;
6. использовать коллекции по варианту. При генерации объекта происходит добавление его в коллекцию (в класс добавить поле идентификатора), а во вторую коллекцию: идентификаторы существующих объектов, в третью идентификатор + время рождения. При возникновении события по таймеру обойдите коллекцию и удалите все объекты, время жизни которых истекло, а также все данные во вспомогательных коллекциях;
7. добавьте в панель управления кнопку «Текущие объекты». По нажатию на эту кнопку появляется модальное диалоговое окно, содержащее список всех «живых» объектов на момент нажатия со временем их рождения (тип объекта, время рождения, идентификатор). В класс диалогового окна должна передаваться коллекция с хранением объектов по времени рождения. Типы коллекций задаются вариантом.

***Вариант 6***

1. Коллекция для хранения объектов: LinkedList
2. Коллекция для хранения и поиска уникальных идентификаторов: TreeSet
3. Коллекция для хранения времени рождения объектов: HashMap

# Описание структуры классов:

Изменить способ хранения информации об объектах в классе Habitat согласно варианту. Добавить в HabitatView вывод статистики созданных объектов на кнопку и в конце симуляции. Объекты должны получить дополнительно поле ID.

# Описание функционала:

* + - 1. Добавление объекта и информации о нем в коллекции

public void addToCollections(int i) {  
 int empAmount = Employee.*Amount*;  
 int lowBound = HabitatView.*Margin*;  
 int upBoundWidth = HabitatView.*Width* - lowBound;  
 int upBoundHeight = HabitatView.*Height* - lowBound;  
 if(i == 0) ObjCollection.add(empAmount, new Worker((int) (lowBound + Math.*random*() \* upBoundWidth), (int) (lowBound + Math.*random*() \* upBoundHeight)));  
 else ObjCollection.add(empAmount, new Manager((int) (lowBound + Math.*random*() \* upBoundWidth), (int) (lowBound + Math.*random*() \* upBoundHeight)));  
 UUID tmpID;  
 do{  
 tmpID = UUID.*randomUUID*();  
 ObjCollection.get(empAmount).setID(tmpID);  
 } while (IdCollection.contains(tmpID));  
 IdCollection.add(tmpID);  
 BornTimeCollection.put(tmpID, ObjCollection.get(empAmount).getBornTime());  
}

При создании объект помещается в первую коллекцию и дополнительная информация о нем, в дополнительные две.

* + - 1. Демонстрация списка живых объектов

public void showCurrent(){  
 pauseSim();  
 JDialog container = new JDialog(\_frameParent, "Current object", true);  
 JTextArea textArea = new JTextArea();  
 JScrollPane scrollPane = new JScrollPane(textArea);  
 container.addWindowListener(new WindowAdapter() {  
 public void windowClosing(WindowEvent e) {  
 continueSim();  
 container.dispose(); }  
 });  
 StringBuilder str = new StringBuilder();  
 for(Employee tmp : \_habitat.ObjCollection)  
 if(tmp instanceof Worker && (*ElapsedTime* - tmp.getBornTime() < Worker.*LifeTime*))  
 str.append(String.*format*("Worker: %s, born time: %6.3f\n", tmp.getID(), tmp.getBornTime()));  
 else if(tmp instanceof Manager && (*ElapsedTime* - tmp.getBornTime() < Manager.*LifeTime*))  
 str.append(String.*format*("Manager: %s, born time: %6.3f\n", tmp.getID(), tmp.getBornTime()));  
 textArea.setText(str.toString());  
 textArea.setDisabledTextColor(Color.*BLACK*);  
 container.setPreferredSize(new Dimension(600, 200));  
 container.add(scrollPane);  
 textArea.setEnabled(false);  
 container.pack();  
 container.setVisible(true);  
}

Демонстрация списка живых объектов. Выводится их ID, время рождения и тип.

# Пользовательский интерфейс

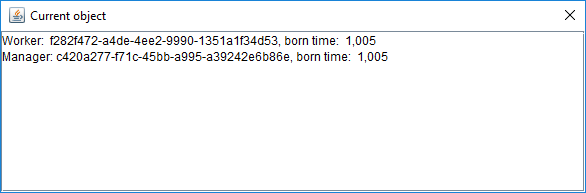
Рис 1. Окно приложения.

Рис 2. Список “живых” объектов

# Выводы:

В ходе лабораторной работы изучена библиотека коллекций java.util. Ознакомились с основными принципами работы с различными коллекциями и их способами хранения данных