Лабораторна робота №3

**UserStory**

package lab\_03;

import java.util.ArrayList;

import java.util.List;

class UserStory extends Ticket {

    private List<UserStory> dependencies;

    public UserStory(int id, String name, int estimate, List<UserStory> dependencies) {

        super(id, name, estimate);

        this.dependencies = dependencies;

    }

    public List<UserStory> getDependencies() {

        return new ArrayList<>(dependencies);

    }

    @Override

    public void complete() {

        boolean allDependenciesCompleted = dependencies.stream().allMatch(UserStory::isCompleted);

        if (allDependenciesCompleted) {

            super.complete();

        }

    }

    @Override

    public String toString() {

        return "[US " + getId() + "] " + getName();

    }

}

**Bug**

package lab\_03;

class Bug extends Ticket {

    private UserStory userStory;

    private Bug(int id, String name, int estimate, UserStory userStory) {

        super(id, name, estimate);

        this.userStory = userStory;

    }

    public static Bug createBug(int id, String name, int estimate, UserStory userStory) {

        if (userStory != null && !userStory.isCompleted()) {

            return new Bug(id, name, estimate, userStory);

        }

        return null;

    }

    @Override

    public String toString() {

        return "[Bug " + getId() + "] " + userStory.getName() + ": " + getName();

    }

}

**Ticket**

package lab\_03;

public class Ticket {

    private int id;

    private String name;

    private int estimate;

    private boolean completed;

    public Ticket(int id, String name, int estimate) {

        this.id = id;

        this.name = name;

        this.estimate = estimate;

        this.completed = false;

    }

    public int getId() {

        return id;

    }

    public String getName() {

        return name;

    }

    public int getEstimate() {

        return estimate;

    }

    public boolean isCompleted() {

        return completed;

    }

    public void complete() {

        completed = true;

    }

}

**Sprint**

package lab\_03;

import java.util.ArrayList;

import java.util.List;

public class Sprint {

    private int capacity;

    private int ticketLimit;

    private List<Ticket> tickets;

    public Sprint(int capacity, int ticketLimit) {

        this.capacity = capacity;

        this.ticketLimit = ticketLimit;

        this.tickets = new ArrayList<>();

    }

    public boolean addUserStory(UserStory userStory) {

        if (userStory != null && !userStory.isCompleted() && canAddTicket(userStory)) {

            tickets.add(userStory);

            return true;

        }

        return false;

    }

    public boolean addBug(Bug bug) {

        if (bug != null && !bug.isCompleted() && canAddTicket(bug)) {

            tickets.add(bug);

            return true;

        }

        return false;

    }

    public List<Ticket> getTickets() {

        return new ArrayList<>(tickets);

    }

    public int getTotalEstimate() {

        int totalEstimate = 0;

        for (Ticket ticket : tickets) {

            totalEstimate += ticket.getEstimate();

        }

        return totalEstimate;

    }

    private boolean canAddTicket(Ticket ticket) {

        return !ticket.isCompleted() && getTotalEstimate() + ticket.getEstimate() <= capacity && tickets.size() < ticketLimit;

    }

}

Результат:

