Минобрнауки России

ФГБОУ ВПО «НИУ МЭИ» АВТИ

Кафедра математического и компьютерного моделирования

**Лабораторная работа №7**

**Технологии программирования**

**"Разработка консольных приложений на языке Java"**

**Работу выполнил:**

Солонин Егор А-14-19

Вариант 14

**Работу принял:**

Князев А. В.

Москва 2021

**Задание:**

1. Составить на языке Java описания классов для указанных объектов.
2. Разработать консольную программу, иллюстрирующую использование объектов заданных классов.

|  |  |  |
| --- | --- | --- |
| 14 | Солонин Е.В. | Рабочий календарь на основе массива (включение, исключение с начала, перенос сроков). |

**Описание работы программы:**

Запустив программу, пользователю предлагается выбрать один из пунктов меню:

1. Инициализация календаря
2. Вывод календаря на экран
3. Включить событие в календарь
4. Удалить ближайшее событие
5. Перенести ближайшее событие
6. Выход

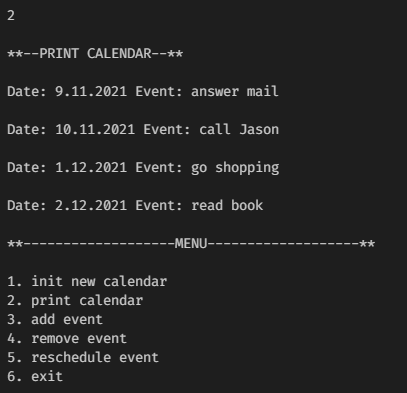
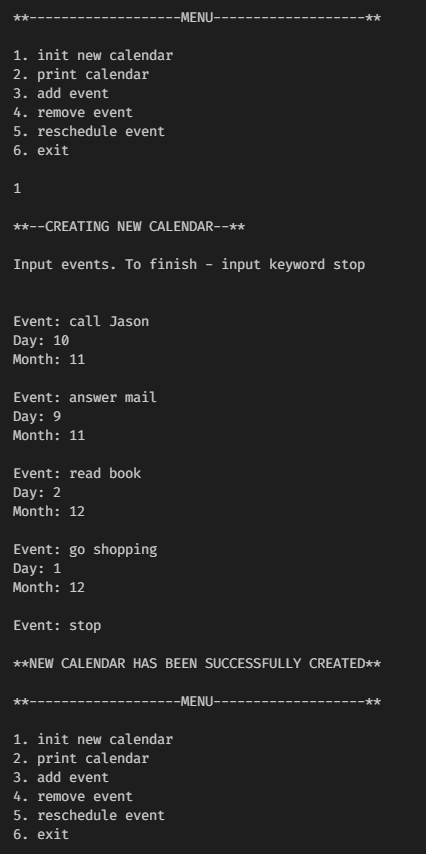
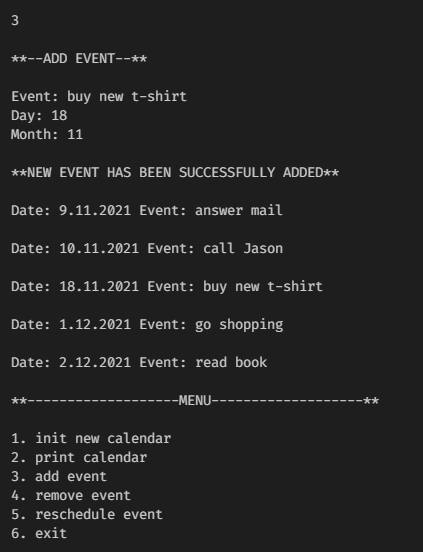
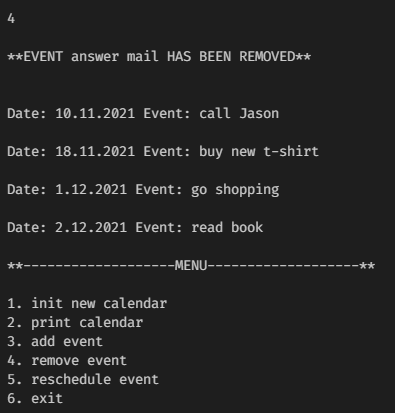
**Алгоритмы операций на псевдокоде:**

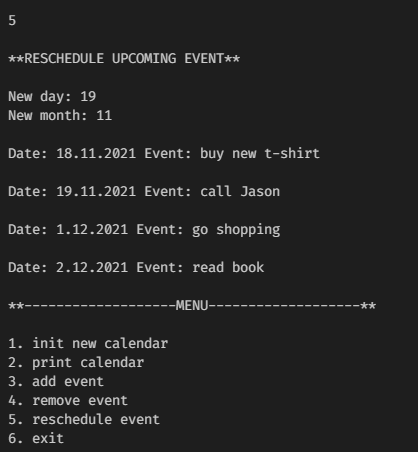
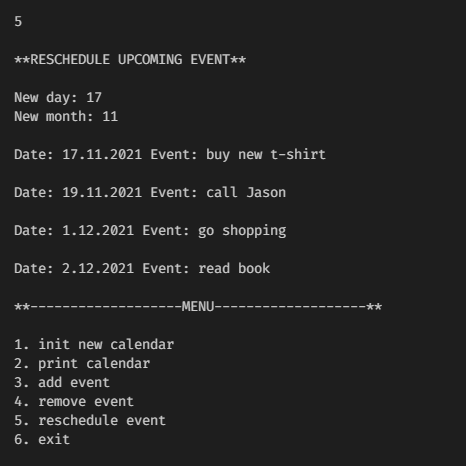
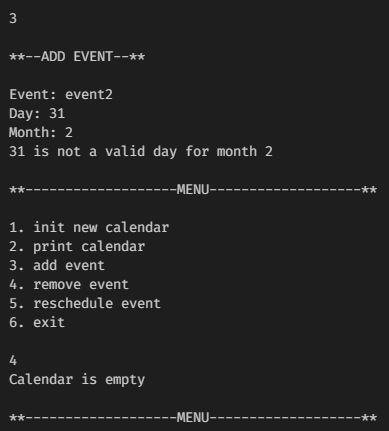
класс Event:

1. int day - день
2. int month - месяц
3. String even – событие

класс Calendar:

1. Внутренний класс *InnerCalendar* с двумя полями
   1. *Event[] events* – массив событий
   2. *int Count* – число событий
2. Объект *calendar* класса *InnerCalendar*
3. *void Add(int day, int month, String event*) – добавление события:
   1. Создаем новое событие *newEvent*
   2. Находим индекс в массиве, после которого добавляем *newEvent*
   3. Сдвигаем все элементы массива после этого индекса вправо на 1
   4. Записываем *newEvent* на освободившееся место
   5. Увеличить на 1 число событий
4. *Event Remove()* – удаление события из начала
   1. Если календарь пуст – сообщить об ошибке, выйти
   2. Запомнить удаляемое событие *removingEvent*
   3. Сдвигаем все элементы массива влево на 1
   4. Уменьшить на 1 число событий
   5. Вернуть *removingEvent*
5. *void Reschedule(int day, int month*) – перенос ближайшего события
   1. *Remove()*
   2. *Add(day, month)*

******Тесты работы программы:**

**Пример ошибок:**

**Листинг программы:**

public class App {

public static void main(String[] args) throws Exception {

Calendar c = new Calendar();

Menu menu = new Menu(c);

int option;

do {

menu.ShowMenu();

option = Integer.parseInt(System.console().readLine());

} while (menu.Launch(option));

}

}

public class Event {

int day;

int month;

String event;

public Event(int month, String event) {

this.event = event;

this.month = month;

}

public Event withDay(int day) {

switch (this.month) {

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

this.day = (day <= 31) ? day : 0;

break;

case 4:

case 6:

case 9:

case 11:

this.day = (day <= 30) ? day : 0;

break;

case 2:

this.day = (day <= 28) ? day : 0;

break;

default:

if (this.month < 1 || this.month > 12)

throw new IllegalArgumentException(

this.month + " is not a valid month. Month must be between 1 and 12");

break;

}

if (this.day <= 0)

throw new IllegalArgumentException(day + " is not a valid day for month " + this.month);

return this;

}

}

public class Calendar {

public class InnerCalendar {

public Event[] events;

public int Count;

public InnerCalendar() {

events = new Event[100];

this.Count = 0;

}

}

InnerCalendar calendar;

public Calendar() {

calendar = new InnerCalendar();

}

public void Add(int day, int month, String event) {

Event newEvent = new Event(month, event).withDay(day);

int i;

for (i = 0; i < calendar.Count; i++) {

if (month > calendar.events[i].month)

continue;

if (month == calendar.events[i].month)

if (day > calendar.events[i].day)

continue;

else

break;

else

break;

}

for (int j = calendar.Count - 1; j >= i; j--)

calendar.events[j + 1] = calendar.events[j];

calendar.events[i] = newEvent;

calendar.Count++;

}

public Event Remove() {

if (calendar.Count == 0)

throw new IllegalArgumentException("Calendar is empty");

Event removingEvent = calendar.events[0];

for (int j = 0; j < calendar.Count - 1; j++)

calendar.events[j] = calendar.events[j + 1];

calendar.Count--;

return removingEvent;

}

public void Reschedule(int day, int month) {

Event removingEvent = this.Remove();

this.Add(day, month, removingEvent.event);

}

public void Print() {

if (calendar.Count == 0) {

System.out.println("\n\*---EMPTY---\*");

return;

}

for (int i = 0; i < calendar.Count; i++) {

String outputDate = String.format("\nDate: %d.%d.2021", calendar.events[i].day, calendar.events[i].month);

System.out.print(outputDate);

System.out.println(" Event: " + calendar.events[i].event);

}

}

}

public class Menu {

Calendar c;

public Menu(Calendar c) {

this.c = c;

}

public void ShowMenu() {

System.out.println("\n\*\*-------------------MENU-------------------\*\*\n");

System.out.println("1. init new calendar");

System.out.println("2. print calendar");

System.out.println("3. add event");

System.out.println("4. remove event");

System.out.println("5. reschedule event");

System.out.println("6. exit\n");

}

public boolean Launch(int option) {

try {

switch (option) {

case 1:

this.Init();

break;

case 2:

this.Print();

break;

case 3:

this.Add();

break;

case 4:

this.Remove();

break;

case 5:

this.Reschedule();

break;

default:

return false;

}

} catch (Exception e) {

        System.out.println(e.getMessage());

    }

return true;

}

public void Init() {

System.out.println("\n\*\*--CREATING NEW CALENDAR--\*\*\n");

System.out.println("Input events. To finish - input keyword stop \n");

while (true) {

System.out.print("\nEvent: ");

String event = System.console().readLine();

if (event.equals("stop"))

break;

System.out.print("Day: ");

int day = Integer.parseInt(System.console().readLine());

System.out.print("Month: ");

int month = Integer.parseInt(System.console().readLine());

c.Add(day, month, event);

}

System.out.print("\n\*\*NEW CALENDAR HAS BEEN SUCCESSFULLY CREATED\*\*\n");

}

public void Print() {

System.out.println("\n\*\*--PRINT CALENDAR--\*\*");

c.Print();

}

public void Add() {

System.out.println("\n\*\*--ADD EVENT--\*\*\n");

System.out.print("Event: ");

String event = System.console().readLine();

System.out.print("Day: ");

int day = Integer.parseInt(System.console().readLine());

System.out.print("Month: ");

int month = Integer.parseInt(System.console().readLine());

c.Add(day, month, event);

System.out.print("\n\*\*NEW EVENT HAS BEEN SUCCESSFULLY ADDED\*\*\n");

c.Print();

}

public void Remove() {

Event deletedItem = c.Remove();

System.out.println("\n\*\*EVENT " + deletedItem.event + " HAS BEEN REMOVED\*\*\n");

c.Print();

}

public void Reschedule() {

System.out.println("\n\*\*RESCHEDULE UPCOMING EVENT\*\*\n");

System.out.print("New day: ");

int day = Integer.parseInt(System.console().readLine());

System.out.print("New month: ");

int month = Integer.parseInt(System.console().readLine());

c.Reschedule(day, month);

c.Print();

}

}