МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

Одесский национальный политехнический университет

Институт компьютерных систем

Кафедра информационных систем

Протокол

лабораторной работы № 2

# «Создание классов и объектов»

по дисциплине

«Объектно-ориентированное программирование»

студент группы АИ-172

Княжев Андрей Алексеевич

Преподаватели:

Годовиченко Н.А.

Одесса 2018

**Цель работы:**

Рассмотреть процесс описания классов и создания объектов, выполнить задания.

**Ход выполнения лабораторной работы:**

Были успешно выполнены задания с созданием и писанием классов и объектов.

**Выполнение заданий:**

**Задание 1.1**

**Поля**

**private int hours**;  
**private int minutes**;

**Метод TimeSpan(hours, minutes)**

Конструктор. Входные аргументы – количество часов и минут

Ниже приведен исходный код запрограммированного метода:

**public** TimeSpan(**int** hours, **int** minutes) {  
 **if** (hours >= 0 && hours < 24) {  
 **this**.**hours** = hours;  
 } **else** {  
 System.***out***.println(**"Wrong time"**);  
 }  
  
 **if** (minutes >= 0 && minutes < 60) {  
 **this**.**minutes** = minutes;  
 } **else** {  
 System.***out***.println(**"Wrong time"**);  
 }  
}

**Метод getHours()**

Возвращает целое количество часов во временном интервале

Ниже приведен исходный код запрограммированного метода:

**public int** getHours() {  
 **return this**.**hours**;  
}

**Метод getMinutes()**

Возвращает целое количество минут во временном интервале

Ниже приведен исходный код запрограммированного метода:

**public int** getMinutes() {  
 **return this**.**minutes** + **this**.**hours** \* 60;  
}

**Метод** add(hours, minutes)

Добавляет указанное количество часов и минут к промежутку. Новое значение промeжутка пересчитывается корректно.

Ниже приведен исходный код запрограммированного метода:

**public void** add(**int** hours, **int** minutes){  
 **if** (hours < 0 || hours >= 24 || minutes < 0 || minutes >= 60) {  
 System.***out***.println(**"Wrong time"**);  
 **return**;  
 }  
  
 **this**.**hours** += hours;  
 **this**.**minutes** += minutes;  
  
 **if** (**this**.**minutes** > 60) {  
 **this**.**minutes** -= 60;  
 **this**.**hours**++;  
 }  
}

**Метод addTimeSpan(timespan)**

Добавляет входной промежуток времени к указанному промежутку

Ниже приведен исходный код запрограммированного метода:

**public void** addTimeSpan(TimeSpan span) {  
 **this**.**hours** += span.getHours();  
 **this**.**minutes** += span.getMinutes() - span.getHours() \* 60;  
  
 **if** (**this**.**minutes** > 60) {  
 **this**.**minutes** -= 60;  
 **this**.**hours**++;  
 }  
}

**Метод getTotalHours()**

Возвращает количество часов в промежутке в виде дробного числа.

Ниже приведен исходный код запрограммированного метода:

**public double** getTotalHours() {  
  
 **return this**.**hours** + **this**.**minutes** / 60;  
}

**Задание 1.2**

**Метод subtract(TimeSpan span)**

Метод вычитает из текущего временного интервала входной временной интервал.

Ниже приведен исходный код запрограммированного метода:

**public void** subtract(TimeSpan span) {  
 **if** (span.**hours** > **this**.**hours**) {  
 System.***out***.println(**"Time span is too big"**);  
 **return**;  
 }  
  
 System.***out***.print(**this**.**hours** + **" h "** + **this**.**minutes** + **" m - "** + span.**hours** + **" h "** + span.**minutes** + **" m = "**);  
 span.**hours** = **this**.**hours** - span.getHours();  
 span.**minutes** = **this**.**minutes** - span.**minutes**;  
  
 **if** (**this**.**minutes** < 0) {  
 **this**.**hours**--;  
 **this**.**minutes** += +60;  
 }  
  
 System.***out***.println(span.getHours() + **" h "** + span.**minutes** + **" m"**);  
}

**Задание 1.3**

**Метод** scale()

Метод принимает массив целых чисел типа int, если 0-ой элемент равен 6 или последний элемент равен 6, возвращает true, иначе возвращает false.

Ниже приведен исходный код запрограммированного метода:

**public static boolean** firstLast6(**int**[] array) {  
 **if** (array[0] == 6 || array[array.**length** -1] == 6) {  
 **return true**;  
 } **else** {  
 **return false**;  
 }  
}

**Задание 2**

**Метод transfer()**

Метод transfer() осуществляет перевод денег из одного счета (текущего) на другой счет (объект receiver). Первый параметр - счет получателя денег, второй параметр - количество денег, которые переводятся со счета на счет.

Ниже приведен исходный код запрограммированного метода:

**public void** transfer(BankAccount receiver, **double** amount) {  
  
 **if** (amount < 0) {  
 System.***out***.println(**"No money"**);  
 **return**;  
 } **else if** (**this**.**balance** < amount + **this**.**transactionFee**) {  
  
 receiver.**balance** += amount - **this**.**balance**;  
 **this**.**balance** = 0;  
 } **else if** (**this**.**balance** < **transactionFee**) {  
 **return**;  
 } **else** {  
 receiver.**balance** += amount;  
 **this**.**balance** -= amount + **transactionFee**;  
 }  
}

**Выводы по лабораторной работе**:

В рамках данной лабораторной работы были решены поставленные задания и был написан код для реализации методов.