Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №1

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Простые структуры данных»

Выполнил:

студент группы 20ВВ2

Беляков А.М.

Проверили:

Митрохин М.А.

Юрова О.В.

Пенза 2021

Задание 1: написать программу, вычисляющую разницу между максимальным и минимальным элементами массива.

Листинг:

from random import randint  
  
  
def create\_massive(minimum, maximum, lenght):  
 return list(randint(minimum, maximum) for \_ in range(lenght))  
  
  
massive = create\_massive(int(input("Введите минимальное число ")), int(input("Введите максимальное число ")), int(input("Введите длинну массива ")))  
massive.sort()  
print(massive[-1]-massive[0])

Задание 2: написать программу, реализующую инициализацию массива

случайными числами.

Листинг:

from random import randint  
  
  
def create\_massive(minimum, maximum, lenght):  
 return list(randint(minimum, maximum) for \_ in range(lenght))  
  
  
massive = create\_massive(int(input("Введите минимальное число ")), int(input("Введите максимальное число ")), int(input("Введите длинну массива ")))  
print(massive)

Задание 3: написать программу, реализующую создание массива произвольного размера, вводимого с клавиатуры.

Листинг:

#include <iostream>

#include <random>

#include <time.h>

int main()

{

srand(time(NULL));

int lenght;

scanf\_s("%d", &lenght);

int\* massive = (int\*)malloc(lenght\*sizeof(int));

for (int i = 0;i < lenght;i++) {

massive[i] = rand() % 200 - 100;

}

for (int i = 0;i < lenght;i++) {

printf("%4d ", massive[i]);

}

}

Задание 4: написать программу, вычисляющую сумму значений в каждом столбце (или строке) двумерного массива.

Листинг:

from random import randint  
  
  
class Matrix():  
 def \_\_init\_\_(self):  
 self.\_minimum = int(input("Введите минимальное число "))  
 self.\_maximum = int(input("Введите максимальное число "))  
 self.\_strings = int(input("Введите количество строк "))  
 self.\_rows = int(input("Введите количество столбцов "))  
 self.\_data = [[randint(self.\_minimum, self.\_maximum) for \_ in range(self.\_rows)] for \_ in range(self.\_strings)]  
  
 @property  
 def minimum(self):  
 return self.\_minimum  
  
 @property  
 def maximum(self):  
 return self.\_maximum  
  
 @property  
 def strings(self):  
 return self.\_strings  
  
 @property  
 def rows(self):  
 return self.\_rows  
  
 @property  
 def data(self):  
 return self.\_data  
  
 def print\_matrix(self):  
 lenght = len(str(self.\_maximum)) + 1  
 print(("+" + "-" \* lenght) \* (self.\_rows + 1) + "+")  
 for i in range(self.\_strings):  
 for j in range(self.\_rows):  
 print(("|{:^" + str(lenght) + "}").format(self.\_data[i][j]), end="")  
 print(("|{:^" + str(lenght) + "}|").format(str(sum(self.\_data[i]))))  
 print(("+" + "-" \* lenght) \* (self.\_rows + 1) + "+")  
 for i in range(self.\_rows):  
 print(("|{:^" + str(lenght) + "}").format(sum([self.\_data[j][i] for j in range(self.\_strings)])), end="")  
  
 print("|" + " " \* lenght + "|\n" + ("+" + "-" \* lenght) \* (self.\_rows + 1) + "+")  
  
  
matrix = Matrix()  
matrix.print\_matrix()

Задание 5: написать программу, осуществляющую поиск среди структур student структуру с заданными параметрами (фамилией, именем и т.д.).

Листинг:

from random import choice, randint  
  
  
class student():  
 def \_\_init\_\_(self, name, surname, faculty, number\_checkbook):  
 self.\_name = name  
 self.\_surname = surname  
 self.\_faculty = faculty  
 self.\_number\_checkbook = number\_checkbook  
  
 @property  
 def name(self):  
 return self.\_name  
  
 @property  
 def surname(self):  
 return self.\_surname  
  
 @property  
 def faculty(self):  
 return self.\_faculty  
  
 @property  
 def number\_checkbook(self):  
 return self.\_number\_checkbook  
  
 def print\_student(self):  
 print(  
 "Cтудент", self.name(), self.surname(), "обучается на факультете",  
 self.faculty() + ", номер зачётной книжки", str(self.number\_checkbook()), sep=" ")  
  
 def \_\_eq\_\_(self, other):  
 if (self.\_name == other.name or other.name == "") and (  
 self.\_surname == other.surname or other.surname == "") and (  
 self.\_faculty == other.faculty or other.faculty == "") and (  
 self.\_number\_checkbook == other.number\_checkbook or other.number\_checkbook == ""):  
 return True  
 else:  
 return False  
  
  
def request():  
 print("Введите ... человека которого хотите найти (если хотите пропустить параметр нажмите enter)")  
 print("Для поиска по нескольким одноимённым параметрам вводите их через пробел ")  
 names = input("Имя ").split(sep=" ")  
 surnames = input("Фамилия ").split(sep=" ")  
 faculties = input("Факультет ").split(sep=" ")  
 number\_checkbooks = input("Номер записной книжки ").split(sep=" ")  
  
 students = []  
 for name in names:  
 for surname in surnames:  
 for faculty in faculties:  
 for number\_checkbook in number\_checkbooks:  
 students.append(student(name, surname, faculty, number\_checkbook))  
  
 return students  
  
  
def search(massive):  
 searching\_students = request()  
 for i in range(len(massive)):  
 for searching\_student in searching\_students:  
 if massive[i] == searching\_student:  
 massive[i].print\_student()  
  
  
massive = []  
names\_gen = ["Александр", "Михаил", "Дмитрий", "Иван", "Роман", "Даниил", "Кирилл", "Максим", "Егор", "Матвей"]  
surnames\_gen = ["Смирнов", "Иванов", "Кузнецов", "Соколов", "Попов", "Лебедев", "Козлов", "Новиков", "Морозов", "Соловьев"]  
faculties\_gen = ["ФВТ", "ФИТЭ", "ФПТЭТ"]  
max\_number\_checkbook = 1000  
for \_ in range(100):  
 massive.append(student(choice(names\_gen), choice(surnames\_gen), choice(faculties\_gen), str(randint(0, max\_number\_checkbook))))  
search(massive)  
print("\n" \* 5)  
for i in range(len(massive)):  
 massive[i].print\_student()

Вывод: Во время выполнения данной лабораторной работы мы вспомнили как работать с списками и структурами.