Методические рекомендации для учащихся предпрофессиональных классов

Москва
2023

Содержание

Рекомендации к выполнению конкурсных вариантов

Материалы практического этапа Московского конкурса межпредметных навыков и знаний «Интеллектуальный мегаполис. Потенциал» (далее – Конкурс) предназначены для оценки уровня практической подготовки участников Конкурса.

Для выполнения задания уровня сложности *“базовый”* участнику необходимо обладать навыками работы с системой контроля версий, элементарными структурами данных, а также уметь работать со строками и файлами. Все требования к которым перечислены в спецификации. Участник должен уметь оформлять код в соответствии с требованиями, уметь работать со списками, стеками и др. похожими типами данных, понимать и уметь выполнять арифметические операции с числами, уметь работать со строками (сравнивать, искать подстроки, разделять строки). Участник должен знать как работать с различными файлами(открывать, изменять, записывать).

Для выполнения задания уровня сложности “*повышенный*” участнику необходимо уметь применять различные алгоритмы поиска и сортировок на практике, исходя из ситуаций описанных в задачах. Также участник должен уметь работать с хеш-таблицами с закрытой и открытой адресацией, понимать свойства хеш-таблицы и применять свойства с современном программировании.

Спецификация конкурсных материалов для проведения практического этапа Московского конкурса межпредметных навыков и знаний «Интеллектуальный мегаполис. Потенциал» в номинации « ИТ-класс» по направлению ИТ.

1. Назначение конкурсных материалов

Материалы практического этапа Московского конкурса межпредметных навыков и знаний «Интеллектуальный мегаполис. Потенциал» (далее – Конкурс) предназначены для оценки уровня практической подготовки участников Конкурса.

2. Условия проведения

Практический этап Конкурса проводится в очной форме на базе вуза. При выполнении работы обеспечивается строгое соблюдение порядка организации и проведения Конкурса.
Используемое оборудование: компьютеры, с установленным ПО (PyCharm, Visual Studio, Visual Studio Code, Git) и доступом в интернет.

3. Продолжительность выполнения

На выполнение заданий практического этапа Конкурса отводится 120 минут.

4. Содержание и структура

Индивидуальный вариант участника включает 6 заданий, базирующихся на содержании элективного курса: «Программирование»

5. Система оценивания

Задание считается выполненным, если ответ участника совпал с эталоном.
Максимальный балл за выполнение всех заданий – 60 баллов. Для получения максимального балла за практический этап Конкурса необходимо дать верные ответы на все задания.

6. Приложения

1. План конкурсных материалов для проведения практического этапа Конкурса.
2. Демонстрационный вариант конкурсных заданий практического этапа Конкурса.

Приложение 1

План конкурсных материалов для проведения *практического* этапа Конкурса

№ задания	**Уровень сложности**	***Уникальные кодификаторы Конкурса***	**Контролируемые требования к проверяемым умениям**	**Балл**
1.	*базовый*	Работа со строками, файлами и графикой 3.1 - Символьные строки 3.2 - Операции со строками 3.3 - Поиск в строках 3.4 - Примеры обработки строк 3.5 - Преобразование число-строка 3.6 - Строки в процедурах и функциях 3.7 - Рекурсивный перебор 3.8 - Работа с файлами 3.9 - Работа с текстовым файлом: чтение, запись, дозапись	Умение работать с файлами(чтение, запись, дозапись). Умение применять операции со строками. Умение решать задачи на рекурсивный перебор	6
2.	*повышенный*	Оценка сложности алгоритмов на примере алгоритмов сортировки: 1.9 - Алгоритмы сортировки 1.10 - Алгоритмы сортировки, основанные на сравнении: сортировка слиянием, быстрая сортировка	Умение оценивать сложность алгоритмов. Умение писать алгоритмы сортировок в заданных условиях	13
3.	*повышенный*	Алгоритмы поиска: 4.1 - Последовательный поиск 4.2 - Двоичный поиск в отсортированном массиве	Умение оценивать сложность алгоритмов. Умение писать алгоритмы поиска в заданных условиях	12
4.	*базовый*	Элементарные структуры данных: 2.1 - Стек 2.2 - Использование списка 2.3 - Вычисление арифметических выражений с помощью стека	Понимание принципов работы списков и стека, умение использовать и	*6*

		2.4 - Проверка скобочных выражений	реализовывать работу со списками и стеком на выбранном языке программирования	
5.	*повышенный*	Хеширование: 6.1 - Хеш-таблицы с закрытой и открытой адресацией 6.2 - Свойства хештаблицы 6.3 - Хеширование в современных языках программирования	Умение реализовывать алгоритмы хеширования текстовых и числовых данных, понимание принципов работы хэш-таблиц, умение работать с хэш-таблицами	20
6.	*базовый*	*Системы контроля версий. Совместная работа над проектом: 8.2 - Оформление программного кода в соответствии с установленными требованиями 8.3 - Руководство по стилю 8.5 - Ветки в Git 8.7 - Проект на Github*	Умение работать с системой контроля версий Git, знание основных компонентов и команд, знание правил ведения репозитория, умение совместно работать над проектом с использованием веток Умение документировать код выбранного языка программирования	*3*
			Сумма баллов:	**60**

Приложение 2

Демонстрационный вариант конкурсных заданий *практического* этапа Конкурса

Добрый день! Сегодня Вам предстоит побывать в роли наставника ребят, которые делали различные проекты. В рамках заданий Вы сможете сделать личный кабинет школьника, получить статистику по проектам, а также создать для каждого из подопечных свой собственный логин и пароль. Думаю, что Вы готовы приступать к задачам, но сначала необходимо **создать репозиторий** для проекта и задач, которые Вы будете выполнять. Обязательно сделайте его **PUBLIC**, а то мы не сможем проверить решение вашей последней задачи. Каждую задачу вам необходимо будет правильно оформить и залить в репозиторий. *Код, который Вы напишете необходимо задокументировать, чтобы другие программисты могли понять, что делает код и за что отвечает. Примеры документирования приведены ниже.*

Пример документирования кода на языке C++

```
/**
  * Это описание функции foo
  *
  * @param str это описание аргумента str
  * @param pattern это описание аргумента pattern
  * @return это описание того, что вернет функция
  */
int foo(std::string str, std::string& pattern)
{
    ...
};
```

Пример документирования на языке Python

```
def complex(real=0.0, imag=0.0):
    """Описание функции complex.

    Описание аргументов:
    real - описание аргумента
    imag - описание аргумента

    """
    if imag == 0.0 and real == 0.0: return complex_zero
    ...
```

Теперь можете приступать к решению задач!

К задачам прикреплен файл `students.csv`, который хранит в себе информацию о учениках и их проектах.

Столбцы: *id, Name(в формате ФИО), titleProject_id(номер проекта, целое число), class(класс, в формате цифра+буква), score(оценки, в формате целого числа или None).*

Разделитель *«,»*.

Задача 1.

Все ребята сдали свои проекты и получили оценки на защите, но Хадаров Владимир все прослушал и просит помочь ему узнать какую оценку за проект он получил. Пожалуйста, подскажите Владимиру какую оценку он получил. Формат вывода: Ты получил: <ОЦЕНКА>, за проект - <id>

Пока помогали Владимиру увидели, что многие ученики потеряли свои оценки при выкачке с сайта. Из-за этого нет возможности посмотреть общую статистику. Чтобы избежать путаницы поставьте вместо ошибки среднее значение по классу и округлите до трех знаков после запятой. Сохраните данные в новую таблицу с названием `student_new.csv`.

Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка (для языка Python – PEP 257,). После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория.

Задача 2

Данные из таблицы `student.csv` необходимо отсортировать по столбцу `оценки(score)` с помощь сортировки вставками (В задаче нельзя использовать встроенные функции сортировок!). Из полученного списка выделите первых 3х победителей из 10 класса. Данные о победителях необходимо вывести в формате:

<X> класс:

1 место: <И. Фамилия>

2 место: <И. Фамилия>

3 место: <И. Фамилия>

…

Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка. После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория

Задача 3

Ввод: стандартный ввод

Вывод: стандартный вывод

Напишите небольшую программу, которая на вход будет получать id проекта (гарантируется, что вводимые числа всегда целые), а на выходе будет предоставлять информацию о ученике, который делал этот проект и его оценку за этот проект в формате: Проект № <N> делал: <И. Фамилия> он(а) получил(а) оценку - <ОЦЕНКА>. Если по заданному запросу ничего не найдено вывести: Ничего не найдено.

Поиск ученика необходимо осуществить с помощью линейного поиска в файле `students.csv`.

Ваша программа должна всегда работать и отключиться только в случае, когда пользователь введет СТОП.

Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка. После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория

Задача 4

Вам необходимо создать личные кабинеты для каждого пользователя, чтобы каждый из них видел свои достижения и мог лично взаимодействовать с вами. Для этого необходимо создать логины и пароли для каждого из школьников. Реализуйте методы/функции, которые будут генерировать логины и пароли для пользователей. Логин должен состоять из фамилии и инициалов, например, если школьника зовут Соколов Иван Иванович, его логин должен выглядеть как Соколов_ИИ. Также для каждого пользователя необходимо сгенерировать пароль, пароль должен состоять из 8 символов, включать в себя заглавные, строчные буквы английского алфавита и цифры.

“0,Сербин Геннадий Михаилович,7,8в,2” → “0,Сербин Геннадий Михаилович,7,8в,2,Сербин_ГМ,fhGi45Bq”

На вход подается CSV файл, который необходимо записать в список, для каждого элемента сгенерировать логин и пароль, после чего дополнить список сгенерированными элементами. Последним этапом полученный список записать в новый students_password.csv файл.

Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка. После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория.

Задача 5

В следующем году планируется дополнительный набор школьников на обучение, в связи с этим поиск по ФИО пользователя будет работать неэффективно. Необходимо составить хэш-таблицу, в которой будет выстроено соответствие ФИО и значения хэша ФИО. На основании этого необходимо составить хэш-таблицу и заменить id ученика на полученный хэш и результат записать в csv файл.

Для хэширования необходимо использовать следующий алгоритм.

$$hash(s) = s[0] + s[1] * p + s[2] * p^2 + \ldots + s[n-1] * p^{n-1} mod\, m = \sum_{i=0}^{n-1} s[i] * p^i mod\, m,$$

где p и m - некоторые выбранные положительные числа.

Рекомендации по выбору чисел p и m.

Целесообразно сделать p простым числом, примерно равным количеству символов во входном алфавите. Например, если входные данные состоят только из строчных букв английского алфавита, можно взять p = 31. Если же входные данные могут содержать как прописные, так и строчные буквы, то возможен выбор p = 53. Если используются прописные и строчные буквы русского алфавита, а также символ пробел, то возможен выбор p = 67.

m должно быть большим числом, так как вероятность столкновения двух случайных строк составляет примерно ≈ 1/m. Иногда выбирают $m = 2^{64}$, поскольку тогда целочисленные переполнения 64-битных целых чисел работают точно так же, как операция модуля. Однако существует метод, который генерирует строки с коллизиями (которые работают независимо от выбора p). Поэтому на практике, $m = 2^{64}$ не рекомендуется. Хорошим выбором для m является какое-либо большое простое число. (можно использовать $m = 10^9+9$, это большое число, но все же достаточно малое, чтобы можно было выполнять умножение двух значений, используя 64-битные целые числа).

Для вычисления хэша строки s, которая содержит только строчные буквы необходимо преобразовать каждый символ строки s в целое число. Можно использовать преобразование a →1, b →2, … z → 26 . Преобразование a → 0 не является хорошей идеей, поскольку тогда хэши строк a, aa, aaa, … все оцениваются как 0.

На вход подается CSV файл students.csv результаты необходимо записать в новый students_with_hash.csv файл.

Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка. После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория.

Задача 6

Ваш код будет использоваться программистом, которого возьмут на работу, поэтому он должен быть правильно оформлен и выложен на GitHub. Весь написанный код должен быть задокументирован согласно стандартам документирования кода выбранного языка.

Также необходимо оформить README.md для Вашего репозитория. Пункты, которые должны быть описаны:

1. Название проекта

2. Описание проекта

3. Оглавление (необязательно)

4. Как установить и запустить проект

5. Как использовать проект

Пример решения демонстрационного варианта

Задание 1

Пример решения на python

```
import csv

with open('students.csv', encoding="utf8") as csvfile:
    reader = csv.reader(csvfile, delimiter=',', quotechar='"')
    answer = list(reader)[1:]
    for id, name, titleProject_id, level, score in answer:
        if 'Хадаров Владимир' in name:
            print(f"Ты получил: {score}, за проект - {titleProject_id}")
            break
    count_class = {}
    sum_class = {}
    for el in answer:
        count_class[el[-2]] = count_class.get(el[-2], 0) + 1
        sum_class[el[-2]] = count_class.get(el[-2], 0) + (int(el[-1]) if
el[-1] != 'None' else 0)
    for el in answer:
        if el[-1] == 'None':
            el[-1] = round(sum_class[el[-2]] / count_class[el[-2]], 3)



with open('students_new.csv', 'w', newline='', encoding='utf-8') as file:
    w = csv.writer(file)
    w.writerow(['id', 'Name', 'titleProject_id', 'class', 'score'])
    w.writerows(answer)
```

Вывод решения программы: Ты получил: 5, за проект – 278

Задание 2

Пример решения на python

```
import csv


with open('students.csv', encoding="utf8") as csvfile:

    reader = list(csv.DictReader(csvfile, delimiter=',', quotechar='"'))

    for i in range(len(reader)):
```

```
        j = i - 1
        key = reader[i]
           while float(reader[j]['score'] if reader[j]['score'] != 'None'
else 0) < float(key['score'] if key['score'] != 'None' else 0) and j >= 0:
            reader[j + 1] = reader[j]
            j -= 1
        reader[j + 1] = key



print('10 класс:')
count = 1
for el in reader:
    if '10' in el['class']:
        surname, name, patronymic =  el["Name"].split()
        print(f'{count} место: {name[0]}. {surname}')
        count += 1
    if count == 4:
        break
```

Вывод программы:
10 класс:
1 место: Д. Дориков
2 место: В. Королупов
3 место: И. Моторыгин

Задание 3

Пример решения на python

```
import csv
with open('students_new.csv', encoding="utf8") as csvfile:
    reader = csv.DictReader(csvfile, delimiter=',', quotechar='"')
    data = sorted(reader, key=lambda x: x['titleProject_id'])


id_project = input()
```

```
while (id_project != 'СТОП'):
    id_project = int(id_project)
    for el in data:
        if int(el['titleProject_id']) == id_project:
            surname, name, patronymic =  el["Name"].split()
            print(f"Проект №{id_project} делал: {name[0]}. {surname} он(а) получил(а) оценку - {el['score']}.")
            break
    else:
        print('Ничего не найдено')
    id_project = input()
```

Пример вывод программы:

Ввод: 56

Вывод: Проект №56 делал: О. Россомахов он(а) получил(а) оценку - 2.

Ввод: 789

Вывод: Ничего не найдено

Задание 4

```
import csv
import string
import random

def create_initials (s):
  names=s.split()
  return f'{names[0]}_{names[1][0]}{names[2][0]}'

def create_password():
  characters = string.ascii_letters + string.digits
  password = ''.join(random.choice(characters) for _ in range(8))
  return password

students_passwords=[]
with open('students.csv', encoding="utf8") as csvfile:
    reader = list(csv.DictReader(csvfile, delimiter=',', quotechar='"'))
    for row in reader:
      row['login']=create_initials(row['Name'])
      row['password']=create_password()
      students_passwords.append(row)
```

```
with open('students_new.csv', 'w', newline='', encoding='utf-8') as file:
      w = csv.DictWriter(file, fieldnames=['id', 'Name', 'titleProject_id',
'class', 'score', 'login', 'password'])
    w.writeheader()
    w.writerows(students_passwords)
```

Входные данные – файл students.csv

```
id,Name,titleProject_id,class,score
0,Сербин Геннадий Михаилович,7,8в,2
1,Папандина Клавдия Яковлевна,304,9в,5
2,Белагина Галина Андреевна,396,9а,4
……
499,Житин Эдуард Адамович,481,9а,5
```

Выходные данные – файл students_password.csv

```
id,Name,titleProject_id,class,score
0,Сербин Геннадий Михаилович,7,8в,2,Сербин_ГМ, fhGi45Bq
1,Папандина Клавдия Яковлевна,304,9в,5,Папандина_КЯ,lhAi45B2
2,Белагина Галина Андреевна,396,9а,4,Белагина_ГА,khSiE5qy
……
499,Житин Эдуард Адамович,481,9а,5,Житин_ЭА, kqSiD5B1
```

Задание 5

```
import csv

def generate_hash(s):

alphabet='абвгдеёжзийклмнопрстуфхцчшщъыьэюяАБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭ
ЮЯ '
  d = {l: i for i, l in enumerate(str1,1)}
  p = 67;
  m = 1e9 + 9;
  hash_value = 0;
  p_pow = 1;
  for c in s:
    hash_value = (hash_value + d[c] * p_pow) % m;
    p_pow = (p_pow * p) % m;
  return int(hash_value)

students_with_hash=[]
with open('students.csv', encoding="utf8") as csvfile:
    reader = list(csv.DictReader(csvfile, delimiter=',', quotechar='"'))
    for row in reader:
      row['id']=generate_hash(row['Name'])
      print(row)
      students_with_hash.append(row)
```

```
with open('students_with_hash.csv', 'w', newline='', encoding='utf-8') as file:
     w = csv.DictWriter(file, fieldnames=['id', 'Name', 'titleProject_id', 'class', 'score'])
    w.writeheader()
    w.writerows(students_with_hash)
```

Входные данные – файл students.csv

```
id,Name,titleProject_id,class,score
0,Сербин Геннадий Михаилович,7,8в,2
1,Папандина Клавдия Яковлевна,304,9в,5
2,Белагина Галина Андреевна,396,9а,4
……
499,Житин Эдуард Адамович,481,9а,5
```

Выходные данные – файл students_with_hash.csv

```
id,Name,titleProject_id,class,score
789355148,Сербин Геннадий Михаилович,7,8в,2
780100198,Папандина Клавдия Яковлевна,304,9в,5
645499270,Белагина Галина Андреевна,396,9а,4
……
27729397,Житин Эдуард Адамович,481,9а,5
```

Задание 6

https://github.com/mmoseva/predprof

Критерии оценивания

Максимальный балл за выполнение всех заданий - 60 баллов. Проверка правильности заданий с написанием кода осуществляется за счет проверки тестов, а также проверки корректного использования алгоритмов. Если участник использовал встроенные библиотеки или использовал не тот алгоритм, который описан в задании - 0 баллов. В задачах с работой с файлами можно получить 50% баллов, те 3 балла, если участник решит первую часть задания, но не сможет создать второй файл.
Задача номер 6 проверяется наличием прикрепленной ссылки на гитхаб и выполненных требований по оформлению репозитория и файла README.md.

Для получения максимального балла за практический этап Конкурса необходимо предоставить все необходимые файлы, ссылку на открытые репозиторий в GitHub с закоммичинными задачами, а также прикрепленные файлы необходимых расширений того языка программирования на котором вы писали.

Список источников

1. Кнут Д.Э. Искусство программирования: в 3-х томах. — 2-е издание. – М.: Мир, 1976 – 1978 .(3-е изд.: Вильямс, 2010)
2. Ахо Альфред В. Структуры данных и алгоритмы: Вильямс / пер. с английского и ред. Минько А. А., Ахо Альфред В., Хопкрофт Джон Э., Ульман Джеффри Д. — М. и др.: Вильямс, 2001. – 382 с.
3. Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.
4. Седжвик Р. Фундаментальные алгоритмы на C++. Части 1 — 5. Анализ. Структуры данных. Сортировка. Поиск. Алгоритмы на графах: Пер. с англ. – К.: Издательство “ДиаСофт”, 2001.
5. Свейгарт, Эл. Автоматизация рутиных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямс, 2016. – 592 с.
6. Сузи, Р. А. Язык программирования Python : учебное пособие / Р. А. Сузи. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 350 c. — ISBN 978-5-4497-0705-5.
7. Вирт, Никлаус Алгоритмы и структуры данных / Никлаус Вирт ; перевод Ф. В. Ткачева. — 2-е изд. — Саратов : Профобразование, 2019. — 272 c.
8. Осипов Н.А. Технологии программирования: Учебное пособие. - Санкт-Петербург: Университет ИТМО, 2016. - 61 с.
9. Мэтиз Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — СПб.: Питер, 2017. — 496 с.: ил. — (Серия «Библиотека программиста»)
10. Алгоритмы. Справочник с примерами на C, C++, Java и Python, 2-е изд.: Пер. с англ.— СпБ.: ООО “Альфа-книга”, 2017. — 432 с .: ил.
11. Поляков К. Ю. Программирование. Python. C++. Часть 1. Учебное пособие — М. : БИНОМ. Лаборатория знаний, 2019. — 144 с.
12. Поляков К. Ю. Программирование. Python. C++. Часть 2. Учебное пособие — М. : БИНОМ. Лаборатория знаний, 2019. — 176 с.
13. Поляков К. Ю. Программирование. Python. C++. Часть 3. Учебное пособие — М. :БИНОМ. Лаборатория знаний, 2019. — 208 с.
14. Поляков К. Ю. Программирование. Python. C++. Часть 4. Учебное пособие — М. :БИНОМ. Лаборатория знаний, 2019. — 192 с.