

**ордена Трудового Красного Знамени федеральное государственное  
бюджетное образовательное учреждение высшего образования  
«Московский технический университет связи и информатики»**



**Московский конкурс  
межпредметных навыков и знаний  
«Интеллектуальный мегаполис.  
Потенциал»**

**Методические рекомендации  
для подготовки к практическому этапу  
в номинации «ИТ – класс в московской школе»  
По направлению «ИТ»  
2023-2024 учебный год**

**Москва 2023**

## Содержание

Введение .....	3
Спецификация конкурсных материалов .....	5
Обобщённый план конкурсных материалов для проведения практического этапа Конкурса .....	6
Демонстрационный вариант .....	9
Решение задач демонстрационного варианта.....	16
Список учебной и методической литературы и другие источники .....	20

## Введение

Данные методические рекомендации предназначены для подготовки школьников ИТ-классов по информатике в практическом этапе Московского конкурса межпредметных навыков и знаний «Интеллектуальный мегаполис. Потенциал» (далее – Конкурс) в номинации «ИТ-класс» по направлению «ИТ».

Целями конкурса является формирование у обучающегося алгоритмического мышления и, соответственно, необходимых знаний и умений, необходимых для успешного развития в направлении дальнейшей деятельности в области программирования. Для достижения поставленных целей образование в области разработки программных средств призвано обеспечить решение следующих задач:

- формирование в процессе решения практических задач у учащихся навыков алгоритмического мышления и понимания средств формального описания алгоритмов;
- овладение приёмами написания программ на языках программирования с использованием основных конструкций;
- осознание практической применимости выполняемых учебных задач в современном обществе для возможного выбора этой области в качестве будущей профессии.

Обучение по направлению «Программирование» должно быть направлено на достижение обучающимися следующих личностных результатов:

- формирование алгоритмического мышления;
- формирование интеллектуальных умений: анализировать информацию, анализировать основные изученные понятия, строить рассуждения, анализировать и сопоставлять теоретические знания с их практической применимостью;
- готовность и способность к самостоятельной, творческой и ответственной деятельности;
- навыки сотрудничества со сверстниками, детьми младшего возраста, взрослыми в образовательной, общественно полезной, учебно-исследовательской, проектной и других видах деятельности;
- готовность и способность к образованию, в том числе самообразованию;

- сознательное отношение к непрерывному образованию как условию успешной профессиональной и общественной деятельности;
- осознанный выбор будущей профессии и возможностей реализации собственных жизненных планов;
- отношение к профессиональной деятельности как возможности участия в решении личных, общественных, государственных, общенациональных проблем;
- понимание роли информационных процессов в современном мире;
- владение первичными навыками анализа и критичной оценки получаемой информации; Метапредметными результатами освоения программы по направлению «Программирование» являются:
  - знание общепредметных понятий: информация, данные, алгоритм, исполнитель, программа, программирование, языки программирования, система, функция, объект,
  - владение универсальными умениями: постановка задачи, формулирование проблемы; поиск, выделение и структурирование необходимой информации; выбор наиболее эффективных методов решения задачи в зависимости от конкретных условий; самостоятельное создание алгоритмов решения задач;
  - умение применить изученные понятия для реализации учебных задач;
  - умение анализировать имеющийся инструментарий и применять его к поставленной задаче;
  - умение анализировать результат своей предыдущей деятельности и приводить его к виду, требуемому на следующем этапе;
  - умение работать с различными источниками информации, применять на практике полученные знания, анализировать модели.

# **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.



**Московский конкурс  
межпредметных навыков и знаний  
«Интеллектуальный мегаполис.  
Потенциал»**

**Обобщённый план  
конкурсных материалов  
для проведения практического этапа Конкурса**



№ задания	Уровень сложности	Уникальные кодификаторы Конкурса	Контролируемые требования к проверяемым умениям	Балл
1.	базовый	<p><i>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</i></p>	<p>Умение работать с системой контроля версий Git, знание основных компонентов и команд, знание правил ведения репозитория, умение совместно работать над проектом с использованием веток</p> <p>Умение документировать код выбранного языка программирования</p>	3
2.	базовый	<p>Элементарные структуры данных: 2.1 - Стек 2.2 - Использование списка 2.3 - Вычисление арифметических выражений с помощью стека 2.4 - Проверка скобочных выражений</p>	<p>Понимание принципов работы списков и стека, умение использовать и реализовывать работу со списками и стеком на выбранном языке программирования</p>	6
3.	базовый	<p>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</p>		6
4.	повышенный	<p>Алгоритмы поиска: 4.1 - Последовательный поиск 4.2 - Двоичный поиск в отсортированном массиве</p>		12

5.	<i>повышенный</i>	Оценка сложности алгоритмов на примере алгоритмов сортировки: 1.9 - Алгоритмы сортировки 1.10 - Алгоритмы сортировки, основанные на сравнении: сортировка слиянием, быстрая сортировка		13
6.	<i>повышенный</i>	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
<b>Сумма баллов:</b>				<b>60</b>





**Московский конкурс  
межпредметных навыков и знаний  
«Интеллектуальный мегаполис.  
Потенциал»**

**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 помощью 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 Михайлович,7,8в,2” → “0,Сербин Геннадий Михайлович,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.

$$\begin{aligned} \text{hash}(s) &= s[0] + s[1] * p + s[2] * p^2 + \dots + s[n - 1] * p^{n-1} \text{ mod } m \\ &= \sum_{i=0}^{n-1} s[i] * p^i \text{ mod } m, \end{aligned}$$

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  $\approx 1/m$ . 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  $a \rightarrow 1, b \rightarrow 2, \dots, z \rightarrow 26$ . Преобразование  $a \rightarrow 0$  не является хорошей идеей, поскольку тогда хэши строк  $a, aa, aaa, \dots$  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*



**Московский конкурс  
межпредметных навыков и знаний  
«Интеллектуальный мегаполис.  
Потенциал»**

**Решение задач  
демонстрационного варианта  
конкурсных заданий  
практического этапа Конкурса**





### ***Задача 1***

Ты получил: 5, за проект – 278

### ***Задача 2***

10 класс:

1 место: Д. Дориков

2 место: В. Королупов

3 место: И. Моторыгин

### ***Задача 3***

Ввод: 56

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

Ввод: 789

Вывод: Ничего не найдено

### ***Задача 4***

*Логины должны строго соответствовать, пароли совпадать по количеству символов и используемым символам.*

#### ***Входные данные – файл students.csv***

```
id,Name,titleProject_id,class,score
0,Сербин Геннадий Михайлович,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 Михайлович,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
```

## Задача 5

### **Входные данные – файл *students.csv***

```
id,Name,titleProject_id,class,score
0,Сербин Геннадий Михайлович,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 Михайлович,7,8в,2
780100198,Папандина Клавдия Яковлевна,304,9в,5
645499270,Белагина Галина Андреевна,396,9а,4
.....
27729397,Житин Эдуард Адамович,481,9а,5,Житин_ЭА,
kqSiD5B1
```

## Задача 6

### 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  
...
```



**Московский конкурс  
межпредметных навыков и знаний  
«Интеллектуальный мегаполис.  
Потенциал»**

**Список учебной и методической литературы  
и другие источники**



1. Информатика. 10 класс. Базовый и углубленный уровни : учебник : в 2 ч. Ч. 1. ФГОС / К. Ю. Поляков, Е. А. Еремин. — М. : БИНОМ. Лаборатория знаний, 2018. — 352 с. : ил.
2. Информатика. 10 класс. Базовый и углубленный уровни : учебник : в 2 ч. Ч. 2. ФГОС / К. Ю. Поляков, Е. А. Еремин. — М. : БИНОМ. Лаборатория знаний, 2018. — 352 с. : ил.
3. Информатика. 11 класс. Базовый и углубленный уровни : учебник: в 2 ч. Ч. 1. ФГОС / К. Ю. Поляков, Е. А. Еремин. — М. : БИНОМ. Лаборатория знаний, 2019. — 232 с. : ил.
4. Информатика. 11 класс. Базовый и углубленный уровни : учебник: в 2 ч. Ч. 2. ФГОС / К. Ю. Поляков, Е. А. Еремин. — М. : БИНОМ. Лаборатория знаний, 2019. — 302 с. : ил.
5. Поляков К. Ю. Программирование. Python. C++. Часть 1. Учебное пособие — М. : БИНОМ. Лаборатория знаний, 2019. — 144 с.
6. Поляков К. Ю. Программирование. Python. C++. Часть 2. Учебное пособие — М. : БИНОМ. Лаборатория знаний, 2019. — 176 с.
7. Поляков К. Ю. Программирование. Python. C++. Часть 3. Учебное пособие — М. : БИНОМ. Лаборатория знаний, 2019. — 208 с.
8. Поляков К. Ю. Программирование. Python. C++. Часть 4. Учебное пособие — М. : БИНОМ. Лаборатория знаний, 2019. — 192 с.
9. Грэхэм Р., Кнут Д., Паташник О. Конкретная математика. Основание информатики: Пер. с англ. – М. Мир, 1998. – 703 с., с илл.
10. Дейт К. Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом "Вильямс", 2005. — 1328 с.: ил.
11. Вирт Н. Алгоритмы и структуры данных. – М.: ДМК Пресс, 2010. – 272 с.
12. Кормен Т., Лейзерсон Ч., Ривест Р, Штайн К. Алгоритмы. Построение и анализ — М., Вильямс, 2011 г. — 1296 стр.
13. Алгоритмы. Справочник с примерами на C, C++, Java и Python, 2-е изд.: Пер. с англ. — СПб.: ООО “Альфа-книга”, 2017. — 432 с. : ил.

14. Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2018. – 284 с.

15. Мэтиз Э. Изучаем Python. Программирование игр, визуализация данных, веб- приложения. — СПб.: Питер, 2017. — 496 с.: ил. — (Серия «Библиотека программиста»)