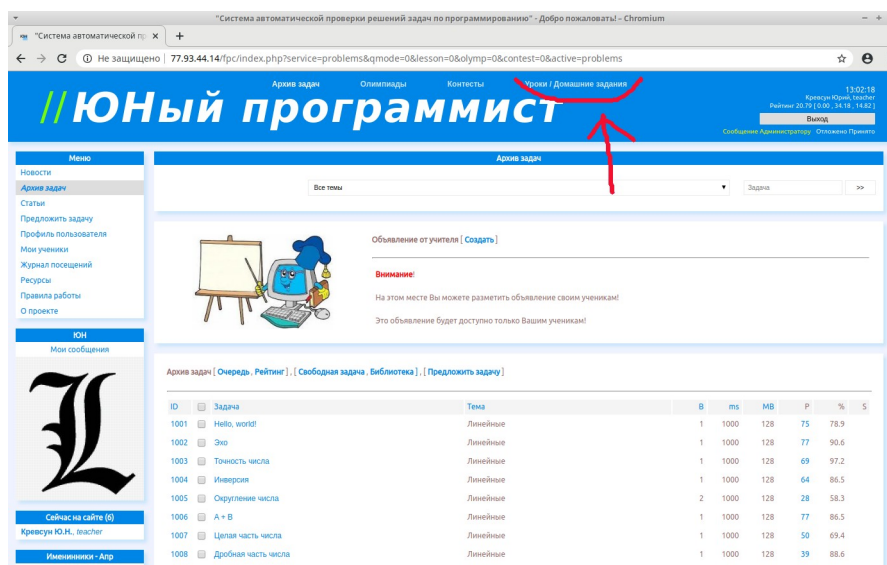
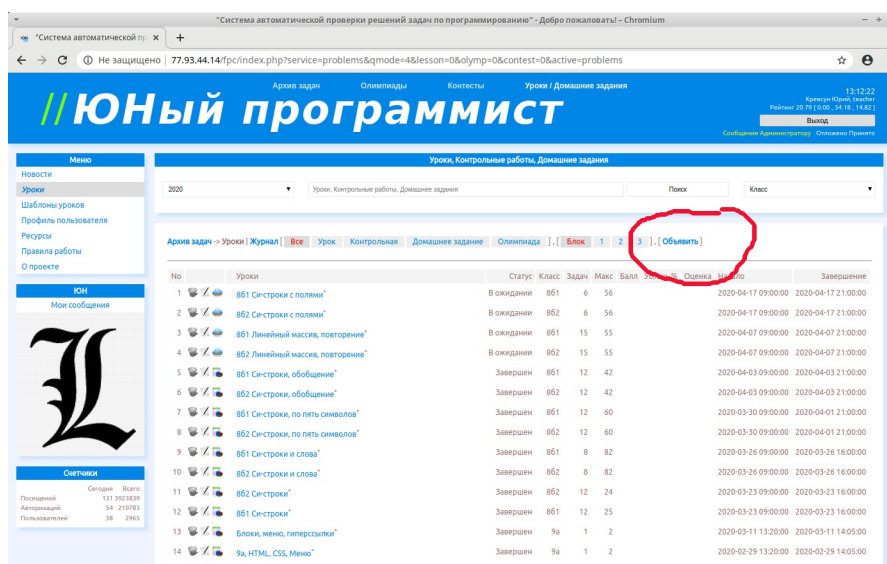


Система автоматической проверки решений задач по программированию

В связи с карантином Система автоматической проверки решений задач по программированию, в дальнейшем Система, претерпела некоторые изменения, которые позволят более эффективно организовать дистанционные уроки по курсу «Программирование».



Для создания дистанционного урока учитель должен перейти в раздел «Уроки/Домашнее задания» и найти ссылку «Объявить». По клику на эту ссылку Система переходит в режим редактирования параметров урока. Так как урок дистанционный, то в описании урока включен пункт «Пояснительная записка». В этой «Пояснительной записке» учитель в произвольной форме дает теоретические сведения по уроку, краткую характеристику предлагаемых на уроке задач и т. д.

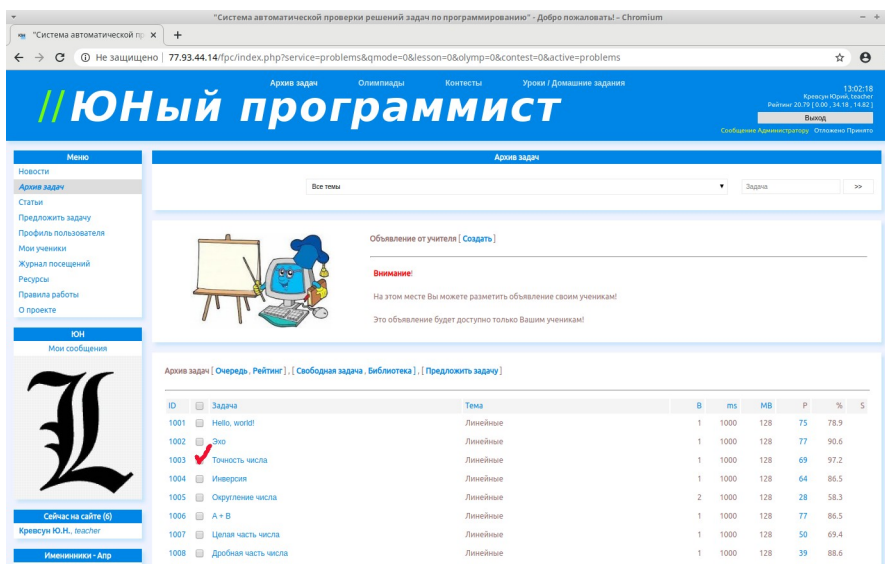


При формировании пояснительной записки допускается использование форматирования текста средствами языка разметки HTML.

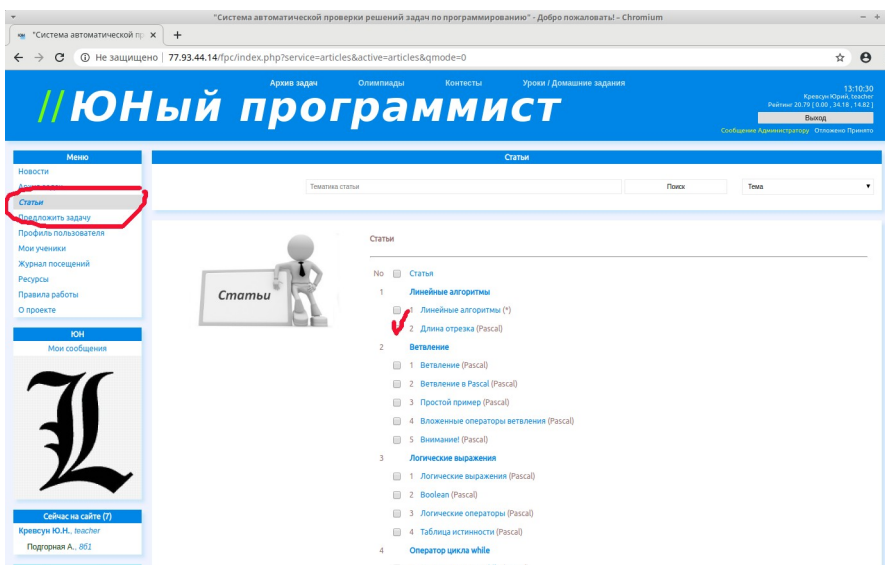
При необходимости подключить к уроку внешний ресурс в пункте «Внешний ресурс» укажите полный URL требуемого ресурса. Это может быть как простая Веб-ссылка, так и скрипт для встраивания внешнего ресурса в страницу урока.

Перед объявлением урока учитель в разделе «**Архив задач**» выбирает задачи, которые планирует предложить ученикам на уроке. Для этого он выставляет «флажок» возле номера нужной задачи. При создании урока выбранные номера задач автоматически будут вписаны в поле «**Номера задач в пределах темы**». Эти задачи и будут предложены ученикам во время проведения урока.

Система также поддерживает автоматический подбор задач на урок. Для этого поле «**Номера задач в пределах темы**» следует оставить пустым и выставить флажки напротив тех разделов тем, по которым будут выбраны случайным образом указанное количество задач. В случае, если не будет выбрана ни одна тема, **Система** случайно выберет задачи из всех тем.

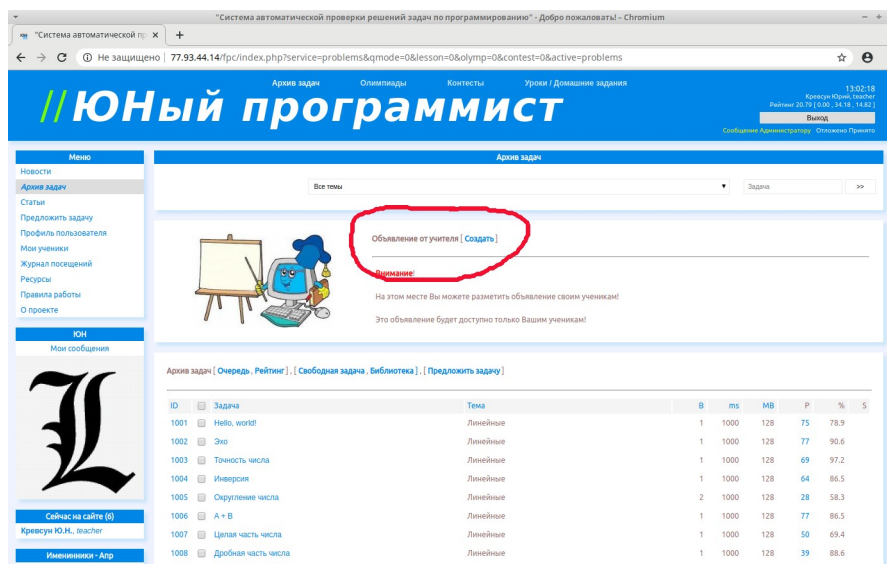


На сайте создан раздел «**Статьи**». В этом разделе собраны различные статьи по различным темам программирования с изложением теории и практическими примерами. При необходимости статью можно прикрепить к уроку, выставив «флажок» возле названия статьи. При создании урока выбранные статьи будут автоматически закреплены за уроком.



При проведении урока прикрепленные статьи отображаются в виде ссылок, при переходе по которым открывается выбранная статья.

При необходимости сделать общее объявление для всех своих учеников, учитель может создать такое объявление на сайте. В разделе «Архив задач» (основной раздел Системы) в верхней части страницы размещена панель «Объявление от учителя». Там же учитель найдет ссылку «Создать», перейдя по которой откроется редактор текста. Набрав необходимый текст и сохранив его объявление будет доступно всем ученикам этого учителя. Для удаления объявления кликните по ссылке «Удалить».



Для возможности многократно объявлять урок по одному плану, например на следующий учебный год или есть несколько параллельных групп учеников, Система поддерживает возможность создания «Шаблона урока». Объявляя шаблон урока учитель получает простой способ создания аналогичного урока в будущем, что значительно ускорит процесс создания урока. Остается только указать дату, время проведения урока и выбрать класс.

В Системе шаблоны уроков, объявляемые администратором для своих учеников сделаны общедоступными для всех учителей. Шаблоны уроков, создаваемые другими учителями доступны только своему создателю.

PS: Учителю, при формированию плана урока предоставляются потенциально небезопасные средства форматирования текста «Пояснительной записки», неверное использование которых может привести к нарушению нормального отображения страницы. Используя теги HTML следите за тем, чтобы все теги, требующие закрытия были закрыты. В противном случае это может привести к неправильному отображению страницы, вплоть до полного разрушения ее структуры, так как ваш HTML код, содержащийся в «Пояснительной записке», будет встроен в код основной страницы Системы.

Мы сознательно пошли на такое нарушение безопасности, чтобы предоставить учителю более гибкий инструмент для формирования страницы урока.

Пример оформления «Пояснительной записки»

Здравствуйтесь,.. садитесь!:)

Сегодня мы повторим тему "Си строки". Напоминаю, что для работы со строками следует подключить библиотеку `string.h`. Для чтения строк будем использовать функцию `gets()`, а саму строку объявлять как линейный массив элементов типа `char`. Например, программа "Эхо" со строками будет выглядеть так:

```
<pre>
#include <stdio.h >
#include <string.h >

int main() {
    char s[265];
    gets(s);
    printf("%s", s);
}
</pre>
```

Напоминаю, что любая строка должна заканчиваться специальным символом `'\0'` - символ, код которого равен ноль. Функции для работы со строками, объявленные в библиотеке `string.h` автоматически контролируют данный символ.

Перечислим некоторые функции библиотеки `string.h`:

```
<pre>
char s1[256], s2[256] - Объявляем две строки s1 и s2

int l = strlen(s1) - Длина строки s1
int r = strcmp(s1, s2) - Сравнить строки s1 и s2:
                        если s1 равно s2, то результат функции равен нулю
                        если s1 больше s2, то результат функции больше нуля
                        если s1 меньше s2, то результат функции меньше нуля

strcpy(s1, s2) - Копирует все символы строки s2 в строку s1
strstr(s1, s2) - Поиск подстроки s2 в строке s1,
                возвращает NULL, если подстрока не найдена
                или адрес ее начала в памяти
strchr(s1, c1) - Поиск символа c1 в строке s1
                возвращает NULL, если символ не найден
                или адрес его в памяти
</pre>
```

На странице это будет выглядеть так:

Здравствуйтесь,.. садитесь!)

Сегодня мы повторим тему "Си строки". Напоминаю, что для работы со строками следует подключить библиотеку `string.h`. Для чтения строк будем использовать функцию `gets()`, а саму строку объявлять как линейный массив элементов типа `char`. Например, программа "Эхо" со строками будет выглядеть так:

```
#include < stdio.h >
#include < string.h >

int main() {
    char s[265];
    gets(s);
    printf("%s", s);
}
```

Напоминаю, что любая строка должна заканчиваться специальным символом `'\0'` - символ, код которого равен ноль. Функции для работы со строками, объявленные в библиотеке `string.h` автоматически контролируют данный символ.

Перечислим некоторые функции библиотеки `string.h`:

```
char s1[256], s2[256]; - Объявляем две строки s1 и s2

int l = strlen(s1)    - Длина строки s1
int r = strcmp(s1, s2) - Сравнить строки s1 и s2:
                        если s1 равно s2, то результат функции равен нулю
                        если s1 больше s2, то результат функции больше нуля
                        если s1 меньше s2, то результат функции меньше нуля

strcpy(s1, s2)        - Копирует все символы строки s2 в строку s1
strchr(s1, s2)        - Поиск подстроки s2 в строке s1,
                        возвращает NULL, если подстрока не найдена
                        или адрес ее начала в памяти
strchr(s1, c1)        - Поиск символа c1 в строке s1
                        возвращает NULL, если символ не найден
                        или адрес его в памяти
```