



**Хусаинов Н.Ш., Балабаева И.Ю., Кривша В.В.**

Институт компьютерных технологий и информационной безопасности  
ФГАОУ ВО "Южный федеральный университет" (ЮФУ)

## **Опыт применения соревновательного и проектного подходов в обучении основам программирования и алгоритмизации студентов первого курса**

16-17 мая 2024 года, г. Тверь



- Повышение уровня компетентности студентов в области разработки программ на языках высокого уровня
  - более эффективное формирование профессиональных навыков составления алгоритмов и их реализации
  - знакомство с возможностями языка программирования как инструмента для профессиональной разработки ПО
  - развитие около-профессиональных навыков (регулярность работы, соблюдение дедлайнов, работа с технической документацией)

# Исходные данные

- Место дисциплины в учебном плане
- "Портрет" студента
- Анализ имевшихся проблем



# Исходные данные. Дисциплина



- Дисциплина "Программирование и основы теории алгоритмов" (09.03.04 Программная инженерия, 1-й курс, осенний семестр)
- Нагрузка на студента (в неделю): 2 ч лек / 1 ч практики / 2 ч лаб
- Контингент - 90..100 чел. (4 группы)
- Основной язык программирования в рамках дисциплины - С (или С++ без классов)
- Дополнительно:
  - дисциплина-спутник (факультатив), 3 ч в неделю
  - еженедельные консультации преподавателя (для желающих)



## *hard-skills*

- ЕГЭ (210+): Информатика / Физика ( $\geq 50$  баллов)
- знакомство с Python на базовом уровне (75% студентов);
- не более 10..15 % студентов, имеют опыт участия в олимпиадах по информатике (выход на муниципальный этап)

## *soft-skills*

- быстрая адаптация к предлагаемому формату работы
- эффективное использование средств коммуникации (соцсети, гаджеты, LMS)
- трудности “долгосрочного” планирования работ
- неумение рефлексировать, осознавать проблемы, обращаться за помощью
- неумение работать с технической документацией, запоминать материал
- ориентир не на знания, а на баллы ( $\Rightarrow$  плагиат)



# Исходные данные. Имевшиеся проблемы



- Мало программирования
  - 1 лаба = 1 программа, итого всего 6-8 программ за семестр;
- Большие группы (25 чел)
  - трудности с код-ревью;
- Выполнение задания дома с защитой в классе
  - вместо оценивания навыка “умеет составлять алгоритм, разрабатывать и отлаживать программу” проверяется навык “умеет читать и понимать программу”;
- Оценка преподавателем “на глаз”
  - высокая степень субъективности в оценке результатов;
- Снижение мотивации студентов
  - язык программирования - компьютерно-ориентированный способ представления алгоритма (без инструментов, используемых на практике);
  - свободное посещение занятий, нарушения графика выполнения и сдачи работ.

До 75 % студентов, "закрывших" дисциплину, признавались что их уровень подготовки недостаточен для самостоятельного выполнения заданий, связанных с разработкой программ в последующих дисциплинах (вплоть до влияния на выбор темы ВКР)

# Алгоритм действий

- Актуализация содержания курса
- Методика реализации дисциплины и инструменты
- Дополнительная мотивация



# Алгоритм действий. Содержание курса



## Информатика для программирования

- Представление информации в ЭВМ, Алгоритмы и их описание, Понятие о языке программирования и компиляции программы

## Язык С

- Типы данных, Операции, Базовые конструкции, Массивы, Указатели, Строки, Подпрограммы, Файлы, Структуры, Препроцессор, Классы памяти

## Элементы профессиональной разработки ПО

- Разработка библиотек, Интернационализация приложений, Стиль кодирования, Автодокументирование кода, Переносимость кода, Система сборки





# Алгоритм действий. Структура курса. Микромодули



Тематические микромодули

Итоговая к/р

Проектное задание

Структура микромодуля (2 недели, 10 баллов)	Балл	Ауд.	С/р	Факультатив
Лекция # 1 (в конце - мини-тест)	0,5	*		
Практика # 1 - разбор 3-5 типовых заданий	-	*		
Д/з # 1 на 5-6 дней (4-6 программ)	2,0		*	* (разбор и дорешка д/з)
Лекция # 2 (в конце - мини-тест)	0,5	*		
Практика # 2 - тест, разбор 2-3 типовых заданий	1,0	*		
Д/з # 2 на 2-3 дня (2-3 программы)	1,0		*	* (разбор и дорешка д/з)
Лабораторная (контрольная) работа (2-5 программ)	5,0	*		* (разбор и дорешка к/р)



# Алгоритм действий. Инструменты



## Онлайн-банки задач и системы проведения соревнований

- практики, д/з ([informatics.msk.ru](http://informatics.msk.ru), [acmp.ru](http://acmp.ru), [stepic.org](http://stepic.org), Codeforces)
- лабы (Codeforces, Яндекс.Контест)

## LMS

- тестирование (Moodle)

## IDE

- проприетарная (MS Visual Studio)
- свободная (Code::Blocks)
- онлайн ([onlinegdb.com](http://onlinegdb.com))

## Онлайн-документация

- презентации лекций
- справочник по стандарту C/C++ ([cppreference.com](http://cppreference.com))

## Утилиты

- автодокументирование (doxygen)
- сборка (cmake)



# Алгоритм действий. Примеры



Что выведет на экран программа:

```
#include <stdio.h>

int main(void)
{
    int i = 1, j = -2, k;

    k = (i >= 0) || (j >= 00) && (i <= 0) || (j <= 0);

    printf("%d",k);

    return 0;
}
```

- 2
- 1
- 3
- 0

- Тестирование на основе сертификационных экзаменов ([www.cppinstitute.org](http://www.cppinstitute.org)):
  - CLE (C Certified Entry-Level Programmer Certification)
  - CLA (C Programming Language Certified Associate)

ЗАДАЧИ ОТОСЛАТЬ МОИ ПОСЫЛКИ СТАТУС ПОЛОЖЕНИЕ АДМ. РЕД. ЗАПУСК

Задачи

№	Название	input.txt / output.txt		
<a href="#">А</a>	<a href="#">Передатчик Радика</a>	input.txt / output.txt 1 с, 256 МБ		
<a href="#">В</a>	<a href="#">Поломанный принтер</a>	input.txt / стандартный вывод 1 с, 256 МБ		
<a href="#">С</a>	<a href="#">Запись матрицы</a>	input.txt / output.bin 1 с, 256 МБ		
<a href="#">+</a>	<a href="#">Добавить новую задачу</a>   <a href="#">Добавить новые задачи из контекста</a>		<a href="#">Показать удаленные задачи (2)</a>	

[Все задачи](#)

- Могут быть заданы ограничения и требования к коду
- Проводится в виде соревнования
- Засчитываются частичные решения, набравшие >25% баллов
- В набор задач включается одна из задач домашнего задания
- Ограничен доступ в интернет (только система проведения соревнования, онлайн-справочник по языку и лекции)
- Выборочный контроль сданных решений после занятия



# Алгоритм действий. Проектное задание



**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ - 2022**

**В. Введение**  
На кафедре МОИ ЯМ Уем несколько лет работает образовательный проект "Школа:Код", который предполагает системные развитие образовательных программ в области информатики и программирования.

Последние образовательные программы реализуются в Школе:Код на платной основе, но наиболее востребованы бесплатные программы. Такие программы были созданы, а они успешно управляли администратор Школы, используя электронные таблицы. Но строить и управлять работой Школы традиционными методами уже недостаточно, программа, работая, несколько достигла пределов, она уже не могла сама слушать, и в этой ситуации ее обойти без автоматизации учета всей фискальной деятельности школы.

Для проработки в Школе:Код принято сокращение наименования образовательных программ, которые преподаватель одной своей школе (различные центры, лагеря, программы "Preparing for ЕГЭ" будет называться просто "ЕГЭ", а "Programming in Java, Basic level" – "Java 1".

Основными документами, на основе которых строится текущая работа, являются Каталог образовательных программ (далее – КАТОГ) и Фестр реализуемых программ (далее – ФЕСТР). Каталог содержит информацию об образовательных программах, которые в принципе могут быть реализованы на базе Школы:Код. В Каталоге для каждой программы фиксируются:

- уникальное полное наименование программы;
- уникальное краткое наименование программы;
- общий объем часов по программе;
- общее число аудиторных занятий по программе.

Все образовательные программы Школы:Код реализуются строго на основании приказа, который формируется в конце учебного года по той или иной программе. Все выходные приказы выносятся в Фестр, который делит каждую программу (список):

- уникальное краткое наименование программы (соответс. с названием в Каталоге);
- период ее реализации (месяц и год начала, месяц и год окончания);
- наименование преподавателя на нее (слушатель) (Идем считать, что в процессе реализации количества слушателей программы не меняется);
- стоимость одного часа обучения на программе для одного слушателя.

В зависимости от количества слушателей программы они могут обучаться в одной группе или быть разбиты на несколько групп администратором.

В Школе:Код принят порядок, что иногда не выполняются реализации одной программы до окончания реализации другой программы преподаватель набора (т.е. в любой момент времени реализуется не более одной программы определенного наименования).

**1. Исходные данные**

Наиболее трудными в Школе:Код являются задачи:

- поиск наименования программы (оставшая часть строки, не содержащая цифры и начальной пробела);
- Общий размерный пр

где  $T$  – стоимость обучения  $N$  – количество обучаемых

```

Получено:
50 15 10
Распределение средств,
выполнено по следующему пр:
Python(1) 108 72 Programming in Python, Basic Level
Java 14 72 Programming for ЕГЭ in Information Systems
C++(1+2) 144 108 Programming in C/C++ - Advanced Level

```

Также в Фестр экспортируется акт Реестр реализуемых образовательных программ **Вестр ЯМ**, который имеет следующий формат:

В первой строке записано число  $S$  – количество программ в Реестре (1 ≤  $S$  ≤ 1000).

В каждой из следующих  $S$  строк записаны через пробел следующие поля, соответствующие  $i$ -й программе (1 ≤  $i$  ≤  $S$ ):

- краткое наименование программы (одно слово до 30 символов);
- период (1, 12) – год начала реализации программы (первый, 10.2022, 3.2023);
- месяц (1, 12) – год окончания реализации программы (первый, 10.2023, 12.2023);
- стоимость обучения на программе в рублях  $T$  (1 ≤  $T$  ≤ 100000);
- количество обучающихся на программе  $N$  (1 ≤  $N$  ≤ 1000);
- количество групп на программе  $M$  (1 ≤  $M$  ≤ 10).

```

Получено:
3
Python(1) 10 2022 1 2023 15000 10 4
Java 10 2022 4 2023 20000 12 1
C++(1) 10 2022 4 2024 20000 12 3

```

Бюджетный администратор готовит таблицу на оплату, который содержит в файле **tab\_MM\_YYYY.txt**. Где  $MM$  и  $YYYY$  – месяц и год, за который сформирована таблица.

Первая строка содержит число  $L$  – общее количество заявок о работе преподавателей за месяц (1 ≤  $L$  ≤ 100).

В каждой из следующих  $L$  строк записаны через пробел следующие поля, соответствующие работе преподавателя на программе:

- фамилия и инициалы преподавателя (одним словом **длинной до 30 символов**);
- пробела, запятой, например, **ПетровИИ**;
- краткое наименование программы (одно слово до 30 символов);
- количество аудиторных часов, отработанных на программе за месяц (1 ≤  $S$  ≤ 200).

```

Получено:
файл tab_12_2022.txt
8
ПетровИИ Python(1) 15
Васюкова Java 10
Труфанов Java 10
Викторянц C++(1+2) 18
Викторянц C++(1+2) 18
Викторянц C++(1+2) 18
Викторянц C++(1+2) 18

```

**2. Запрос на выполнение расчета заработной платы**  
Выходными являются запрос на формирование расчетного листа/расчетные листы по отдельному преподавателю или по всем преподавателям за месяц. Запрос на расчет записан в файле **FamilyID MM.YYYY**

где: "MM" – номер месяца, означающего запрос на формирование; **FamilyID** – фамилия и инициалы преподавателя (одно слово), для которого необходимо выполнить расчет; для выполнения расчета по всем преподавателям в поле указывается символ "+".

```

MM.YYYY – расчетный период в формате "MM.YYYY" в интервале 01.2001-12.2029.
Sum_Teacher(12_2022)
Список:
Sum = 12_2022

```

**3. Результаты работы программы**  
Программа должна сформировать текстовый файл **result.txt**, содержащий расчетный лист, представляющий в указанной форме. Результат работы программы должен иметь следующий вид. В первой строке через пробел записаны следующие поля:

- краткое наименование программы **FamilyID** MM.YYYY
- фамилия и инициалы преподавателя (одно слово), для которого выполнен расчет;
- MM.YYYY – расчетный период в формате "MM.YYYY" (MM означает два обязательных цифра, YYYY – четыре обязательных цифра);
- Символы "+" – количество строк дубликации, которое будет выведено сформир для данного преподавателя (не считая итоговой строки);
- Файл с названием подпрограммы строки выводится дублированной информацией по каждой программе, на которой преподаватель выполнял работу в расчетном месяце, который имеет следующий формат:

```

ProgramName Name (From_SumMM.YYYY) IN SumInProg

```

В каждой из строк наименование преподавателя; **Name** – количество часов, отработанных преподавателем по программе в расчетный период (длина строки); **SumInProg** – плановая ставка оплаты по программе в расчетный период в формате "руб/час"; **From\_SumMM.YYYY** – стоимость аудиторных часов по образовательной программе в расчетный период (по всем преподавателям, выполняющим в реализации этой программы в отчетный период); **SumInProg** – сумма оплаты преподавателю по программе в расчетный период в формате "руб/час/месяц".

После решения строго одной проблемы, дополнительные проблемы отсутствуют, в конце каждой строки – символ конца строки. Строки упорядочиваются по краткому наименованию программы в обратном порядке.

Выводится расчетный листок строки:

```

TOTAL FamilyID TotalValue

```

- ТЗ примерно на 3-4 стр. с описанием "кейса", приближенного к реальной жизни
- Выдается на 1 месяц
- Включает задания в т.ч. по темам, которые не вошли в микромодулях, но есть в лекциях
- Многокритериальная проверка преподавателем (вручную):
  - правильность работы программы;
  - архитектура приложения (грамотность декомпозиции на модули и подпрограммы);
  - особенные требования (поддержка кириллицы, реализация в виде библиотеки);
  - стиль кодирования;
  - разработка документации (автодоментирование).



# Алгоритм действий. Дополнительная мотивация



- Место в постоянно обновляемом рейтинге
- Символические призы по итогам микромодуля
- Бонусные баллы за участие в соревнованиях по программированию
- Общение через ВК



- Турнир для начинающих программистов

- задания придумывают сами студенты
- командный формат
- призы от института





## *Факультатив*

- Для тех, у кого есть трудности:
  - разбор заданий;
  - дорешивание
- У кого все хорошо:
  - задания повышенной сложности

## *Ликвидация задолженностей*

- Контролируемое решение задач (написание программ)
- Повторное прохождение дисциплины со следующим курсом

# Результаты

- Промежуточные итоги



# Промежуточные итоги (4 цикла)

- Больше программирования (в 5..6 раз)
- Объективность оценивания
- Возможность работы с большими группами
- Ритмичность и регулярность работы студента во время семестра, высокая посещаемость занятий
- Эффективная самостоятельная работа
- Поддержание вовлеченности студентов в процесс освоения дисциплины в течение всего семестра
- Оценки от студентов (за 2 года, из 10): собственный прогресс: 7,1-7,5; реализация ожиданий от дисциплины: 7,8-8,1; Повышение самооценки наших студентов (по сравнению с другими образовательными программами)
- Около 10% студентов уходят сразу по окончании семестра, освобождая места для перевода к нам

- Не получается прививать хороший “стиль кодирования”
- Большой объем внеаудиторной нагрузки на преподавателя:
  - необходимость ежегодной разработки новых заданий для контрольных работ и проектного задания;
  - “ручная” проверка проектного задания и выборочный контроль д/з и лаб на плагиат
- На факультатив приходят в основном те, у которых и так все получается. На консультации почти не приходят
- Количество задолжников по итогам семестра уменьшилось незначительно





**Хусаинов Наиль Шавкятович** - зав. кафедрой, к.т.н., [khusainov@sfedu.ru](mailto:khusainov@sfedu.ru)

**Балабаева Ирина Юрьевна** - старший преподаватель, [ibalabaeva@sfedu.ru](mailto:ibalabaeva@sfedu.ru)

**Кривша Виталий Владимирович** - доцент, к.т.н., [krivsha@sfedu.ru](mailto:krivsha@sfedu.ru)



Кафедра математического обеспечения и применения ЭВМ  
Институт компьютерных технологий и информационной безопасности  
Южный федеральный университет