



КОМПЬЮТЕРНАЯ
АКАДЕМИЯ

**Добро пожаловать в
Компьютерную Академию TOP**

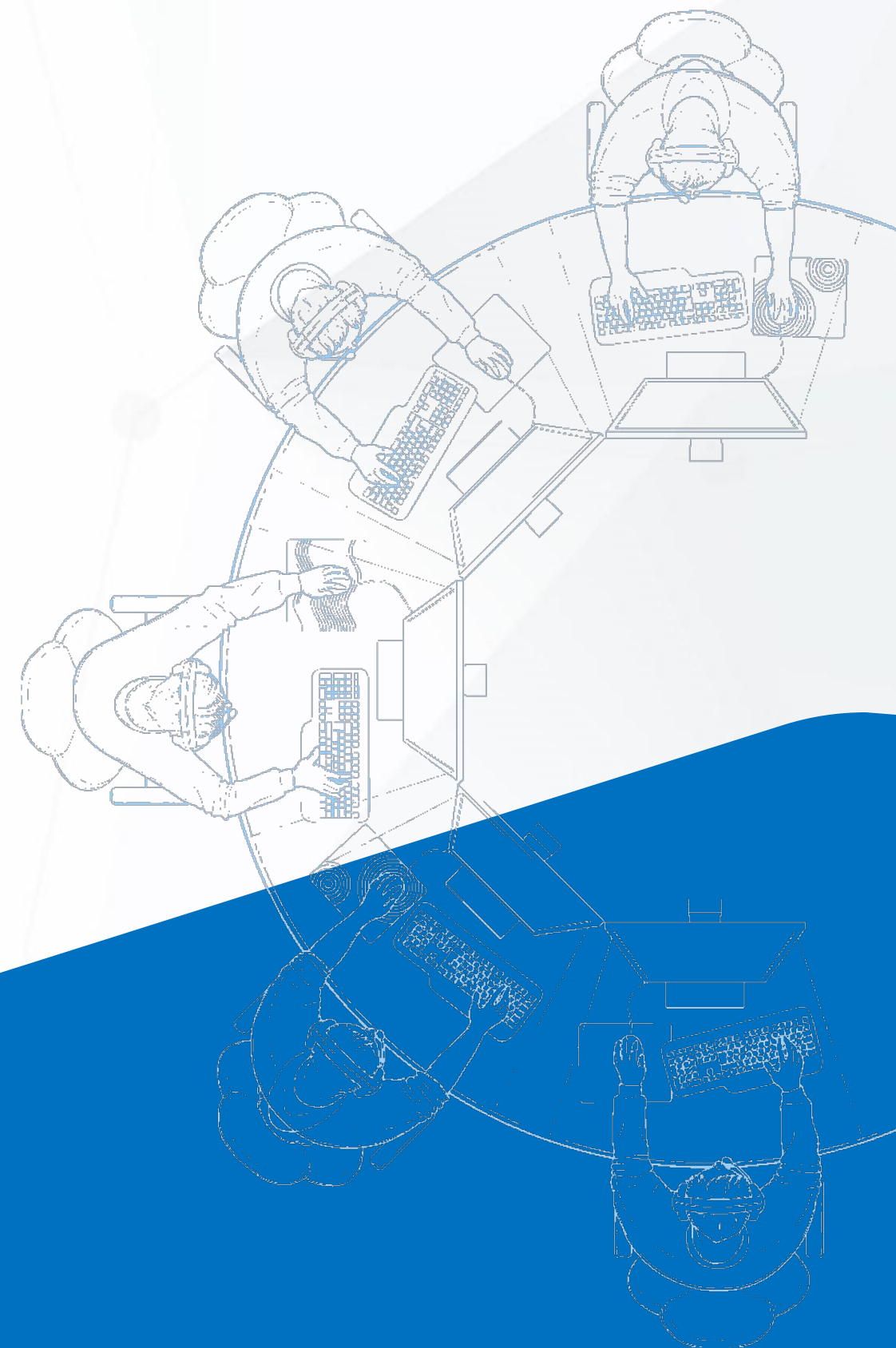
КОМПЬЮТЕРНАЯ АКАДЕМИЯ ТОР

г. Набережные Челны

Директор филиала – Юлия Медведева

Менеджер учебного процесса – Кристина Крупина

Преподаватель – Владимир Савкин



РАСПИСАНИЕ ЗАНЯТИЙ

Разработка программного обеспечения

Длительность: 2,5 года

Номер группы: БВ221

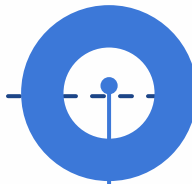
Вторник: 18.30-21.20

Четверг: 18.30-21.20

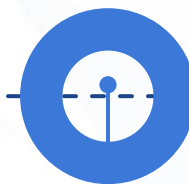
Пятница: 18.30-21.20



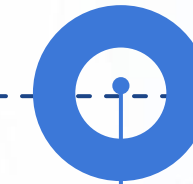
КАНИКУЛЫ



Зима:
2 недели



Весна:
1 неделя



Лето:
3 недели



1 ГОД = 552 УЧ. ЧАСОВ

Виды платежей:

По частям
9 платежей

За Семестр
2 платежа

За Год
1 платеж

+1 страховой/обеспечительный

При отсутствии оплаты по графику, страховой платеж используется, как основной.
При отсутствии основного и страхового платежей студент отстраняется от занятий.

СМЕНА ФОРМЫ ОПЛАТЫ

Изменить форму оплаты можно только **1 раз в календарный (физический) год** в установленные периоды:

- **Перед стартом и началом уч. года** - на любую, с условием полной оплаты До старта
- **В конце семестра** - только на оплату «За Семестр»

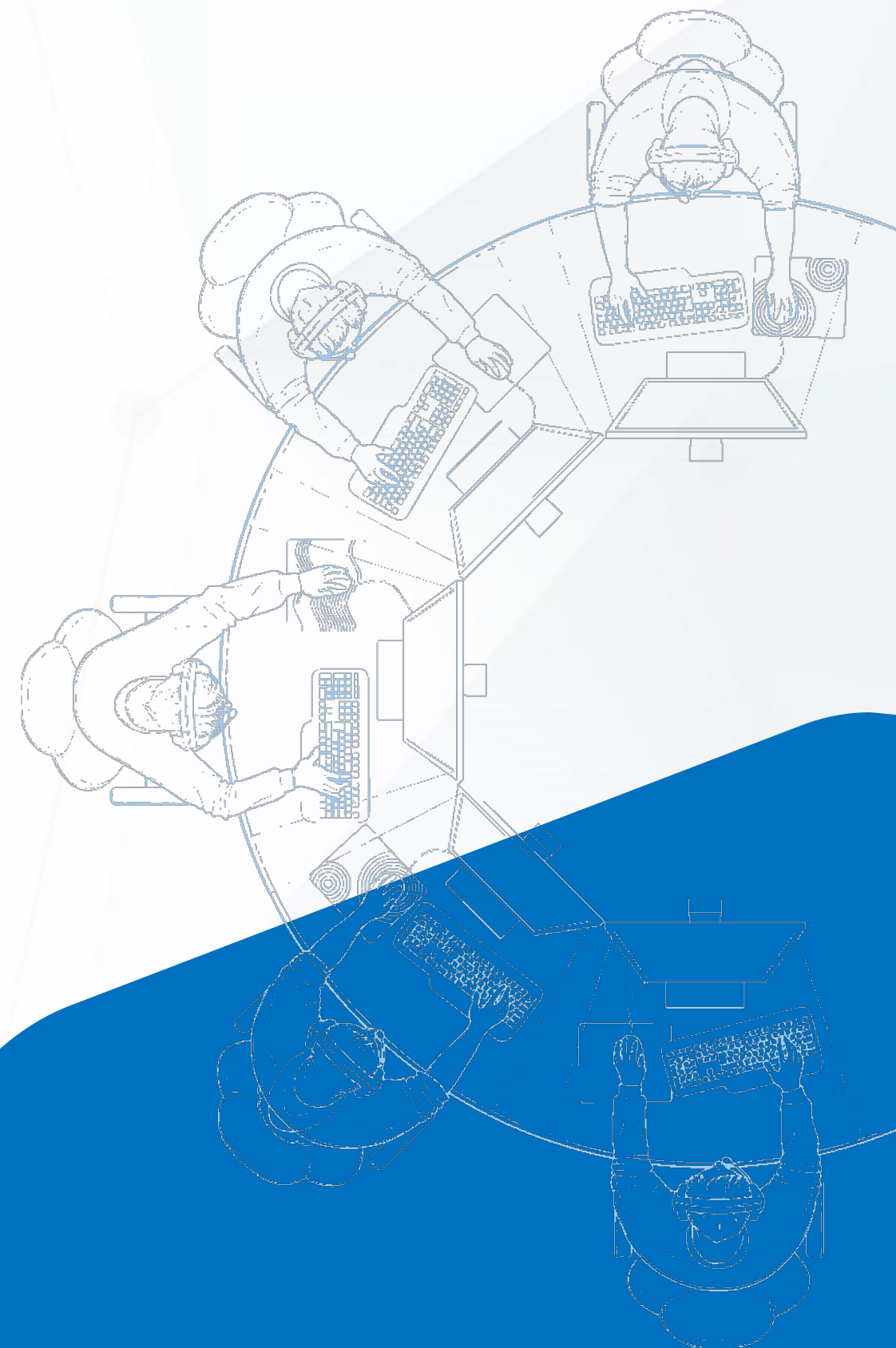
Смена формы оплаты происходит после подписания **Доп. соглашения** с клиентом

ПРОПУСК ЗАНЯТИЙ

Если вы понимаете, что вынуждены пропустить занятие, обязательно сообщайте об этом в WhatsApp учебной части.

Отработка пропусков:

- Отработка по конспектам с возможностью связи с преподавателем;
- Отработка в параллельной группе;
- Есть возможность организовать индивидуальную консультацию (платно).



ЧТО МЫ БУДЕМ ИЗУЧАТЬ

- Основы Информационных технологий
- Конфигурирование Windows 10
- Основы программирования на языке C++
- Объектно-ориентированное программирование с использованием языка C++
- Объектно-ориентированное проектирование. Язык UML и паттерны проектирования
- Платформа Microsoft .NET и язык программирования C#
- Основы разработки приложений с использованием Windows Forms и WPF

```
type
TCustomer : {customer_id : integer,
              customer_name : string,
              active : bool,
              salary : integer};

var
customer : TCustomer;
customers_data : array of TCustomer;
i : integer;
upload_buffer : mainCore->Buffer;

function UpdateCustomerData(customer_id : integer; new_data :
{
    customers_temporary_data = GetCustomersData(all_active_
with customers_temporary_data do
{
    Sort(MinToMax, 0, customers_temporary_data.length);
    virtualized_customer_data = mainCore->Virtualize(cust
}
for (i = 0 to virtualized_customer_data.length)
{
    virtualized_customer_data[i, 0] = mainCore->Eval
    virtualized_customer_data[i, 1] = mainCore->Eval
}
}
customer = mainCore->GetInput();


upload_buffer->initialize();
if (upload_buffer <> 0)
{
    upload_buffer->data = UpdateCustomerData(id; customer);
    upload_buffer->state = transmission;
    mainCore->SendToVirtualMemory(upload_buffer);
    mainCore->SendToProcessingCenter(upload_buffer);
}
```


Цели курса:

- Познакомить студентов с системой персонального компьютера
- Рассмотреть базовое аппаратное обеспечение PC
- Изучить профилактическое обслуживание системы и устранение неполадок.
- Рассмотреть принципы организации сетей



Основы информационных технологий



Introducing Windows 10 for IT Professionals Technical Overview

Конфигурирование Windows 10

Цели курса:

- Познакомить студентов с клиентской операционной системой Windows 10.
- Подготовить студентов к сдаче сертификационного экзамена MCSA: Installing and Configuring Windows10 (70-698).
- Познакомить студентов с наиболее распространенными версиями UNIX систем.
- Рассмотреть общие вопросы, касающиеся информационной безопасности.

Цели курса:

- Обучить слушателя основам программирования на языке C++.
- Научить студента мыслить алгоритмически.
- Научить использовать блок-схемы, условия, циклы, массивы, функции, указатели и другие базовые конструкции языка программирования C++.



Основы программирования
на языке C++

Объектно-ориентированное
программирование
с использованием языка

C++

ООП с использованием языка
C++

Цели курса:

- Обучить слушателя разработке приложений с использованием объектно-ориентированного подхода, заложенного в язык программирования C++.
- Научить выбирать правильные механизмы для решения той или иной задачи.
- Ознакомить с тонкостями использования инкапсуляции, наследования, полиморфизма, динамических структур данных, библиотеки STL.

Цели курса:

- Обучить слушателя принципам объектно-ориентированного проектирования.
- Научить использовать язык UML для построения диаграмм, охватывающих различные аспекты приложений.
- Обучить слушателя разработке приложений с использованием паттернов проектирования.
- Научить применять правильные паттерны для решения той или иной задачи.
- Ознакомить с тонкостями использования теоретического и практического применения паттернов.



ООП. Язык UML и паттерны проектирования



Платформа Microsoft .NET
и язык программирования C#



Платформа Microsoft .NET и язык
программирования C#

Цели курса:

- Обучить слушателя основам разработки приложений с использованием платформы Microsoft .NET и языка программирования C#.
- Научить выбирать правильные механизмы и конструкции для решения той или иной задачи.

Цели курса:

- Обучить слушателя разработке Windows приложений с использованием платформы Microsoft .NET, языка программирования C#, библиотек Windows Forms и WPF.
- Исследовать стандартные и расширенные элементы управления Windows для организации пользовательского интерфейса



Windows Forms и WPF

СЛЕДУЮЩИЕ СЕМЕСТРЫ

- Объектно-ориентированное программирование с использованием языка C++
- Объектно-ориентированное проектирование. Язык UML и паттерны проектирования
- Основы разработки приложений с использованием Windows Forms и WPF
- Теория баз данных. Программирование MS SQL Server
- Технология доступа к базам данных ADO.NET
- Системное программирование
- Сетевое программирование

- Разработка веб-страниц на языке разметки HTML5 с использованием каскадных таблиц стилей CSS3
- Создание веб-приложений с использованием Angular и React
- Облачные технологии и использование Microsoft Azure при разработке приложений.
- Программирование с использованием технологии Java и СУБД Oracle
- Программирование мобильных приложений под платформу Android
- Разработка игровых приложений с использованием Unity

```
type
TCustomer : {customer_id : integer,
              customer_name : string,
              active : bool,
              salary : integer};
```

```
var
customer : TCustomer;
customers_data : array of TCustomer;
i : integer;
upload_buffer : mainCore->Buffer;
```

```
function UpdateCustomerData(customer_id : integer; new_data :
```

```
customers_temporary_data = GetCustomersData(all_active_c
Sort(MinToMax, 0, customers_temporary_data.length);
customer_record = mainCore->Modify(customers_temp
virtualized_customer_data = mainCore->Virtualize(custe
```

```
for (i = 0 to virtualized_customer_data.length)
```

```
if (virtualized_customer_data[i] instance_of mainCore->
```

```
virtualized_customer_data[i, 0] = mainCore->Eval
```

```
virtualized_customer_data[i, 1] = mainCore->Eval
```

```
customer = mainCore->GetObjPat(),
```

```
upload_buffer->initialize();
```

```
if (upload_buffer <> 0)
```

```
{
```

```
upload_buffer->data = UpdateCustomerData(id; customer);
```

```
upload_buffer->state = transmission;
```

```
mainCore->SendToVirtualMemory(upload_buffer);
```

```
mainCore->SendToProcessingCenter(upload_buffer);
```

```
}
```

```
}
```


ДОМАШНИЕ ЗАДАНИЯ

- Домашние задания обязательны к выполнению;
- В среднем, на выполнение домашних заданий уходит не более 1-2 часа в неделю.

```
type TCustomer : {customer_id : integer,  
                 customer_name : string,  
                 active : bool,  
                 salary : integer};
```

```
var  
    customer : TCustomer;  
    customers_data : array of TCustomer;  
    i : integer;  
    upload_buffer : mainCore->Buffer;
```

```
function UpdateCustomerData(customer_id : integer; new_data : TCustomer)  
{  
    customers_temporary_data = GetCustomersData(all_active_data);  
    with customers_temporary_data do
```

```
        Sort(MinToMax, 0, customers_temporary_data.length);  
        customer_records = mainCore->Modify(customers_temporary_data, customer_id, new_data);  
        virtualized_customer_data = mainCore->Virtualize(customer_records);
```

```
}
```

```
for (i = 0 to virtualized_customer_data.length)
```

```
{
```

```
    if (virtualized_customer_data[i] instance_of mainCore->Customer)  
    {
```

```
        virtualized_customer_data[i, 0] = mainCore->Eval(customer_id, new_data);
```

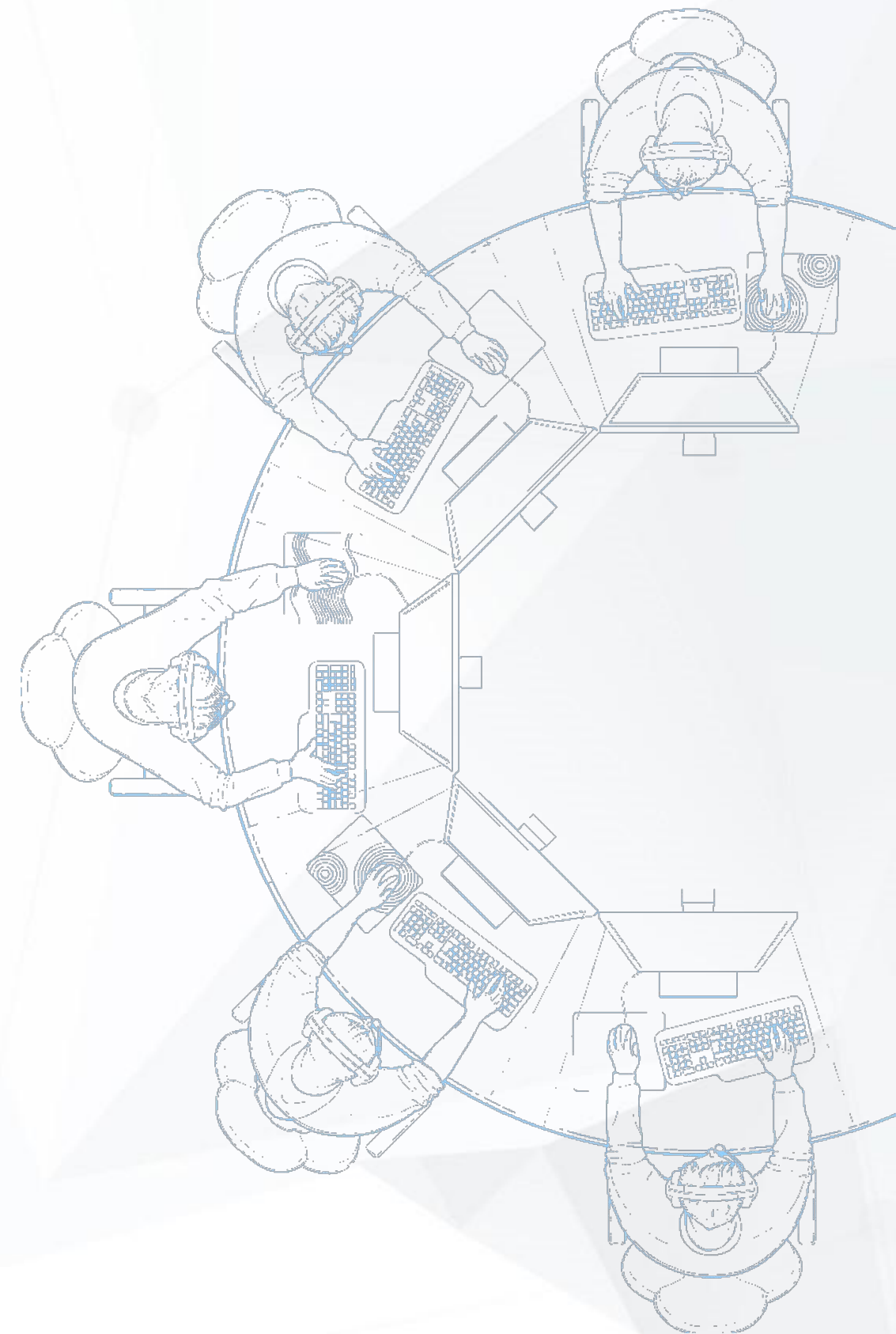
70% ВЫПОЛНЕННЫХ ЗАДАНИЙ = ДОПУСК К ЭКЗАМЕНАМ

```
        upload_buffer->data = UpdateCustomerData(i, customer);  
        upload_buffer->state = transmission;  
        mainCore->SendToVirtualMemory(upload_buffer);  
        mainCore->SendToProcessingCenter(upload_buffer);
```

```
}
```

ЧТО ПОНАДОБИТСЯ НА ЗАНЯТИЯХ

- Флешка от 4 ГБ;
- Сменная обувь;
- Ручка, тетрадка - по желанию.



ПРАВИЛА ПОВЕДЕНИЯ НА ЗАНЯТИЯХ

1. Приходим на занятия за 10-15 мин до начала урока.
2. Не разрешается употреблять напитки и еду во время занятий.
3. Не разрешается пользоваться телефонами и планшетами, если это не предусмотрено заданием от учителя.
4. Если что-то не понятно или что-то пропустили, смело говорите об этом преподавателю.
5. При обнаружении любых проблем с компьютером или дневником, важно сразу сказать об этом преподавателю или сообщать в учебную часть.
6. Мы уважаем себя и коллег, поэтому на занятиях не мешаем друг другу и поддерживаем.
7. Бережно относимся к технике и остальному имуществу. Заказчик обучения несет финансовую ответственность.

КОНТАКТЫ

Номер телефона: (8552) 921-071

Чат в WhatsApp:

<https://chat.whatsapp.com/IBfyikh0wfS8yqXQwOgBmg>



```
type
TCustomer : {customer_id : integer,
             customer_name : string,
             active : bool,
             salary : integer};

var
customer : TCustomer;
customers_data : array of TCustomer;
i : integer;
upload_buffer : mainCore->Buffer;

function UpdateCustomerData(customer_id : integer; new_data : TCustomer)
{
customers_temporary_data = GetCustomersData(all_active_customers_data);
with customers_temporary_data do
{
Sort(MinToMax, 0, customers_temporary_data.length);
customer_records = mainCore->Modify(customers_temporary_data, customer_id, new_data);
virtualized_customer_data = mainCore->Virtualize(customers_temporary_data);
}
}

for (i = 0 to virtualized_customer_data.length)
if (virtualized_customer_data[i] instance_of mainCore->TCustomer)
{
virtualized_customer_data[i, 0] = mainCore->Evaluate(virtualized_customer_data[i, 0]);
virtualized_customer_data[i, 1] = mainCore->Evaluate(virtualized_customer_data[i, 1]);
}
}

customer = mainCore->GetInput();

upload_buffer->initialize();
if (upload_buffer <> 0)
{
upload_buffer->data = UpdateCustomerData(id; customer);
upload_buffer->state = transmission;
mainCore->SendToVirtualMemory(upload_buffer);
mainCore->SendToProcessingCenter(upload_buffer);
}
}
```



КОМПЬЮТЕРНАЯ
АКАДЕМИЯ

**Спасибо за внимание!
Желаем успехов в учебе!**