



**Добро пожаловать в
Компьютерную Академию ШАГ**

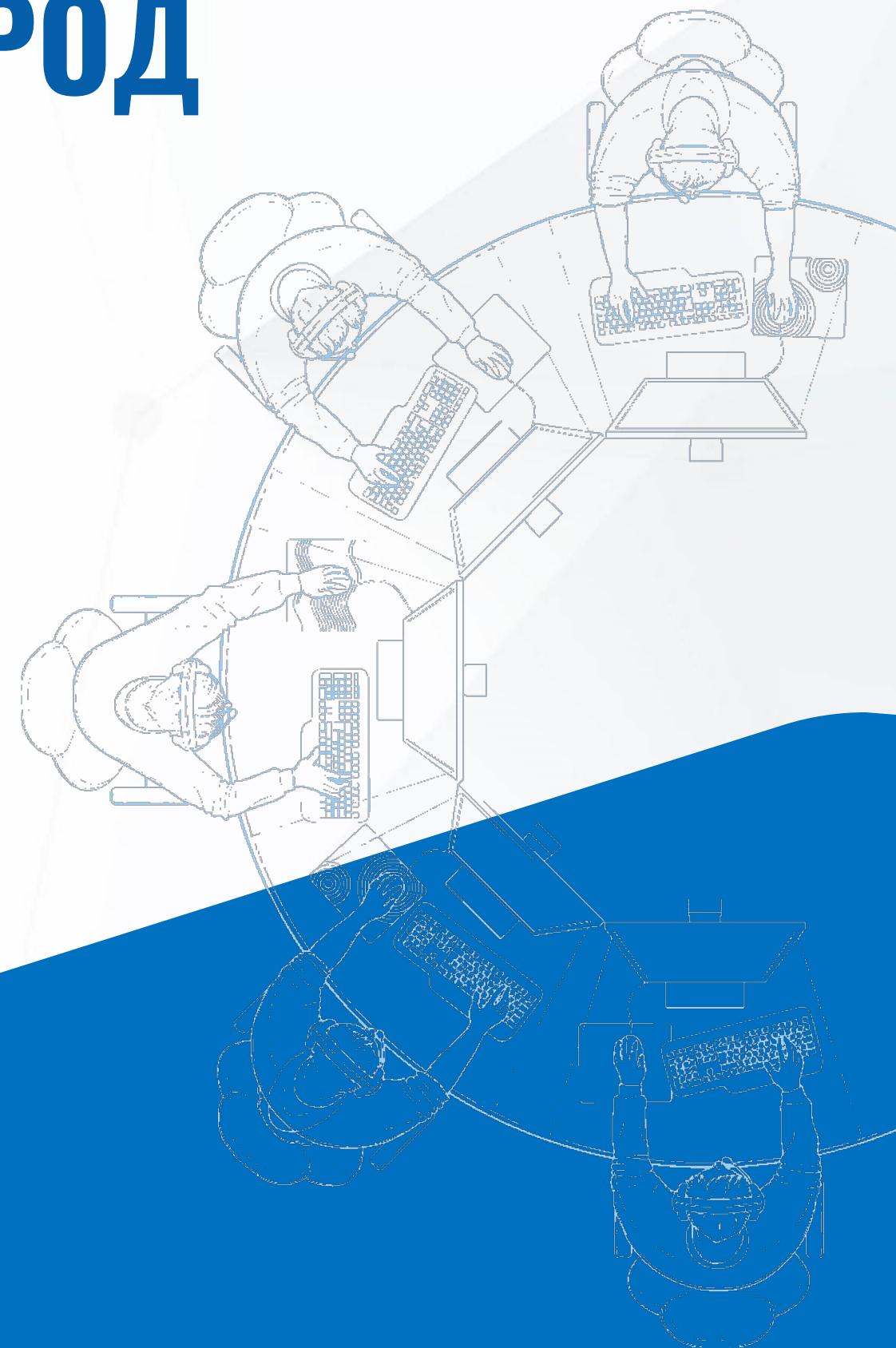
КОМПЬЮТЕРНАЯ АКАДЕМИЯ ШАГ ГОРОД

Директор филиала – Тимофеева
Регина

Руководитель учебной части –
Муратова Лилия

Менеджеры учебного процесса –
Юсупова Яна, Нуржанова Анара

Преподаватель – Иваненко Иван



РАСПИСАНИЕ ЗАНЯТИЙ

Малая Компьютерная Академия

Длительность: 5 лет

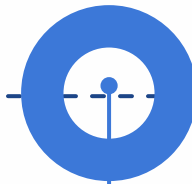
Номер группы: В2113

Первое занятие: 21.11.2021

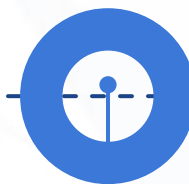
Воскресенье: 09:00-11:50



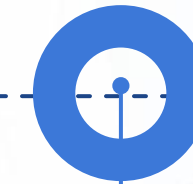
КАНИКУЛЫ



Зима:
С 30.12 по 9.01



Весна:
С 1.05 по 3.05
С 8.05 по 10.05.



Лето:
С 29.05 по
31.08



1 ГОД = 144 УЧ. ЧАСА

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

Написать заявление на смену формы оплаты **МОЖНО ТОЛЬКО** в определенный период:

- Перед началом учебного года - на любую форму

Смена формы возможна только 1 раз за учебный год.

```
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(customer_id, new_data);
            virtualized_customer_data[i, 1] = mainCore->Evaluate(customer_id, new_data);
        }
    }

customer = mainCore->GetInput();

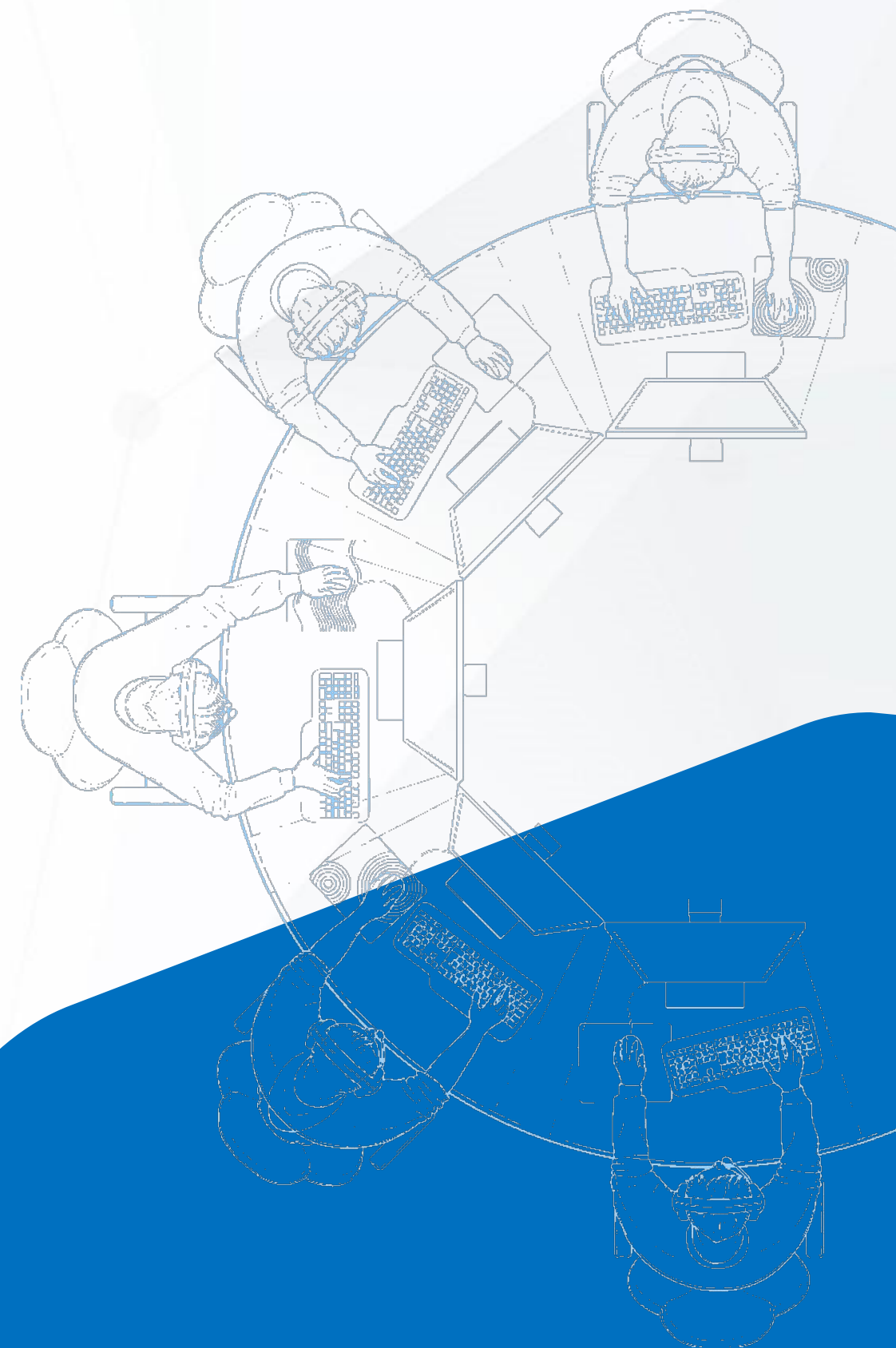
upload_buffer->initialize();
if (upload_buffer <> 0)
{
    upload_buffer->data = UpdateCustomerData(id; customer);
    upload_buffer->state = transmission;
    mainCore->SendToVirtualMemory(upload_buffer);
    mainCore->SendToProcessingCenter(upload_buffer);
}
}
```

ПРОПУСК ЗАНЯТИЙ

Если вы понимаете, что вынуждены пропустить занятие, обязательно сообщайте об этом в Viber или по телефону 220-20-41 (учебная часть)

Отработка пропусков:

- Отработка по пн и чт 18:00-19:20 (по предварительной записи);
- Отработка в параллельной группе (если будут места);
- Есть возможность организовать индивидуальную консультацию (платно).



ЧТО МЫ БУДЕМ ИЗУЧАТЬ

- Создание сайтов Wix, Canva 2.0
- Разработка игр на KODU
- 3D-моделирование и виртуальная реальность (Tinkercad и CoSpaces)
- Игровой дизайн
- Разработка игр – Junior (Construct 3)
- Робототехника и техника (LEGO)

```
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->global.TCustomer)
        {
            virtualized_customer_data[i, 0] = mainCore->Evaluate(some_expression);
            virtualized_customer_data[i, 1] = mainCore->Evaluate(some_expression);
        }
    }
}

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);
}
```


СЛЕДУЮЩИЕ СЕМЕСТРЫ

- Разработка мобильных приложений дополненной реальности
- Создание видео – YouTube
- LEGO Pro
- Программирование на Python –Junior
- Аркадные игры на Click Fusion 2.5
- Веб-дизайн – Junior (HTML & CSS)
- Программирование микроконтроллера BBC Micro:bit
- Фотолаборатория
- Разработка сайтов на WordPress
- Цифровая архитектура – SketchUp
- Разработка приложений на Python – Middle
- Анимация и мультипликация
- Диджитал арт
- Создание гаджетов на Arduino
- Искусственный интеллект и большие данные (Python – Senior)
- 3D-анимация
- Разработка игр – Middle (Unreal Engine)
- Видеолаборатория
- Создание виртуальных миров
- Инновационные технологии
- Стартап и фриланс
- Разработка игр – Senior (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 : 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->global.TCustomer)
        {
            virtualized_customer_data[i, 0] = mainCore->Evaluate(some_condition);
            virtualized_customer_data[i, 1] = mainCore->Evaluate(some_condition);
        }
    }
}

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);
}
```

ДОМАШНИЕ ЗАДАНИЯ

- Домашние задания обязательны к выполнению;
- Все задания можно найти в MyStat.

```
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->Customer)
    {
      virtualized_customer_data[i, 0] = mainCore->Eval(
        mainCore->Code(
          'upload_buffer->data = UpdateCustomerData(i, customer);
          upload_buffer->state = transmission;
          mainCore->SendToVirtualMemory(upload_buffer);
          mainCore->SendToProcessingCenter(upload_buffer);
        ')
      );
    }
  }
}
```

60% ВЫПОЛНЕННЫХ ЗАДАНИЙ = ДОПУСК К ЭКЗАМЕНАМ



MyStat - электронный дневник студента

Web-версия mystat.itstep.org



GET IT ON
Google Play



Download on the
App Store

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. При обнаружении любых проблем с компьютером или MyStat, важно сразу сказать об этом преподавателю или сообщать в учебную часть.
5. Мы уважаем себя и коллег, поэтому на занятиях не мешаем друг другу и поддерживаем.
6. Бережно относимся к технике и остальному имуществу. Заказчик обучения несет финансовую ответственность.

ЧТО ПОНАДОБИТСЯ НА ЗАНЯТИЯХ

- Сменная обувь;
- Ручка, тетрадка - по желанию.

На первое занятие:

- Соглашение на то, что ребёнок может сам уходить домой.
- Соглашение на размещение фото учебного процесса в соцсетях.

Соглашения нужно отдать МУП или оставить на ресепшене.



КОНТАКТЫ

Почта: zavuch_chel@itstep.org

**Номер телефона: пн - пт с 9:00 до 18:00
+7 (351) 220-20-41**

Чат в Viber: +79993337964

```
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(customer_records);
    }

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

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);
}
}
```



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