



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Магнитогорский государственный технический университет им. Г.И. Носова»

УТВЕРЖДАЮ:

Начальник

Центра довузовской подготовки



*З.С. Акмаева*

*01* *2023г.*

РАБОЧАЯ ПРОГРАММА

**РАЗРАБОТЧИК НА PYTHON: С НУЛЯ ДО JUNIOR**  
*Возраст обучающихся от 12 лет*

*Срок реализации 36 часов*

Рабочая программа  
составлена:

Кагарманова Лира  
Альфировна

*(должность, ученая степень,  
ученое звание)*

*Кагарманова*  
*(подпись) И.О. Фамилия)*

Магнитогорск – 2023

## **1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

### ***1.1. Нормативно-правовые основания разработки программы:***

– Федеральный закон от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;

– Приказ Минпросвещения России от 09.11.2018 № 196 (ред. от 30.09.2020) «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» (Зарегистрировано в Минюсте России 29.11.2018 № 52831);

– Приказ Минобрнауки России от 03.10.2014 № 1304 «Об утверждении требований к освоению дополнительных общеобразовательных программ, обеспечивающих подготовку иностранных граждан и лиц без гражданства к освоению профессиональных образовательных программ на русском языке» (Зарегистрировано в Минюсте России 17.11.2014 № 34732);

– Письмо Минобрнауки России от 18.11.2015 № 09-3242 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»);

### ***1.2. Направленность программы***

Данная общеобразовательная программа имеет техническую направленность.

### ***1.3. Новизна, актуальность, педагогическая целесообразность***

Актуальность программы «Разработчик на Python: с нуля до junior» вызвана потребностью современного информационного общества в высокообразованных, адаптированных к изменениям специалистах в IT-сфере. Для удовлетворения данной потребности перед дополнительным образованием стоит задача развития человеческого потенциала через выявление талантливых детей, развитие их мотивации и способностей.

Дополнительная общеобразовательная программа «Разработчик на Python: с нуля до junior» является программой технической направленности и продвинутого образовательного уровня.

Традиционное изучение в общеобразовательных учреждениях города языка программирования «Pascal» и Basic не может в полной мере удовлетворить потребности обучающегося информационно-технологической и физико-математической направленностей в изучении новых и популярных языков программирования.

Изучение языка программирования «Python» поможет ребенку получить более целостное представление о профессии программиста, разработчика, инженера.

Целесообразность программы выражена в подборе интерактивных и практикоориентированных форм занятий, способствующих формированию

основных компетенций (информационных, коммуникативных, компетенций личного развития и др.).

#### ***1.4. Отличительные особенности программы***

Ключевой особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и его реализации с помощью средств программирования.

Программа предполагает участие обучающихся в интеллектуальных соревнованиях по программированию различных уровней, создание образовательных практических или научно-исследовательских проектов, что даст возможность детям полностью реализовать свой интеллектуальный и творческий потенциал.

#### ***1.5. Категории (возраст) обучающихся***

Возраст детей, участвующих в реализации данной общеразвивающей программы: от 12 до 14 лет.

#### ***1.6. Срок освоения программы \_\_\_\_\_ час.***

Сроки реализации (продолжительность обучения): 18 недель с января по май, по 2 часа (всего 36 часов)

#### ***1.7. Форма обучения***

Для реализации программы предусматривается очная форма обучения, при необходимости могут использоваться дистанционные образовательные технологии.

#### ***1.8. Формы и режим занятий обучающихся***

Занятия проводятся 1 раз в неделю по 2 часа.

Основными формами и методами изучения являются лекции, практические занятия, практикумы по решению задач, устные сообщения учащихся с последующей дискуссией.

Практические занятия проходят в форме тестирования, практикума, совместной продуктивной деятельности, моделирования и защиты проектов, проблемном изложении материала, с помощью которого дети сами решают познавательные задачи.

В ходе реализации программы применяется дифференцированный, индивидуальный подход к каждому обучающемуся.

## **2. ЦЕЛИ И ЗАДАЧИ ПРОГРАММЫ**

Основная цель программы – удовлетворение потребностей обучающихся в интеллектуальном совершенствовании посредством подготовки к ликвидации пробелов знаний в области алгоритмизации и программирования на языке «Python».

В процессе реализации программы решаются следующие **задачи**:

- выявить и устранить пробелы в знаниях общеобразовательного предмета;
- актуализировать, систематизировать и углубить знания обучающихся;
- сформировать связанные с учебной дисциплиной компетенции;
- повысить психологическую готовность обучающихся к сдаче экзаменов и последующему обучению;
- расширять кругозор обучающихся в области программирования;
- способствовать возможностям получения новых знаний в области компьютерного программирования;
- профориентировать обучающихся.

### **3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ПРОГРАММЫ**

В результате освоения дополнительной образовательной программы обучающиеся должны:

*знать:*

- роль компьютерного программирования в развитии общества, изменении содержания и характера деятельности человека;
- историю, эволюцию и место языка Python среди языков программирования высокого уровня;
- синтаксис, основные алгоритмические конструкции и парадигмы программирования языка программирования Python;
- основные подходы к созданию программ на высокоуровневом языке программирования Python;
- применение полученных знаний в области прикладной математики и информатики;

*уметь:*

- распознавать необходимость применения той или иной алгоритмической конструкции при решении задач;
- организовывать данные для эффективной алгоритмической обработки;
- реализовывать полученные навыки для решения задач в олимпиадах по программированию различного уровня;

*владеть:*

- разными способами работы с информацией.

### **4. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН**

№ п/п	Наимено вание тем	Трудоемк ость, час	Все го, ауд.	В том числе		Дистанцио нные занятия,	Само ст. рабо	Форм а контр
				лекц ии	практ ич.			

			Час ов		занят ия	час	та, час	оля
1	Введение в Python		12	5	7			
1.1	Техника безопасн ость. Общие с ведения о языках програм мирован ия.		2	1	1			Опрос
1.2	Язык Python. Среда програм мирован ия. Особенн ости ввода- вывода		4	2	2			Практ ическ ое зadan ие
1.3	Типы данных, операции . Оператор присваи вания		3	1	2			Выпо лнени е упраж нений
1.4	Числа. Стандарт ные операции		3	1	2			Практ ическ ое зadan ие
2	Алгорит мические инструкц ии		15	6	9			
2.1	Логическ ие выражен ия и опер аторы.		2	1	1			Практ ическ ое зadan ие

2.2	Условный оператор		3	1	2			Опрос
2.3	Реализация ветвления в языке Python.		3	1	2			Практическое задание
2.4	Цикл while. Теория чисел		4	2	2			Практическое задание
2.5	Цикл for		3	1	2			Практическое задание
3	Строки		9	3	6			
3.1	Литералы строк		3	1	2			Практическое задание
3.2	Срезы строк		3	1	2			Практическое задание
3.3	Методы строк		3	1	2			Практическое задание

## 5. СОДЕРЖАНИЕ ПРОГРАММЫ

№ п/п	Наименование темы	Содержание обучения по темам, наименование и тематика практических занятий, самостоятельной работы
1.1	Инструктаж по технике безопасности. Общие сведения о языках программирования.	<b>Теория:</b> техника безопасности в компьютерном классе. Общие сведения о языках программирования. Что такое программирование. Что такое программа и программный код.

		<b>Практика.</b> Первая программа в Python 3.5.
1.2	Язык Python. Среда программирования. Особенности ввода-вывода	<b>Теория.</b> Язык программирования Python. Достоинства и недостатки. Области применения. Интерактивный режим работы программы. <b>Практика.</b> Установка среды программирования PyCharm
1.3	Типы данных, операции. Оператор присваивания	<b>Теория.</b> Ввод и вывод числовой информации. <b>Практика.</b> Тренировочное задание на ввод и вывод числовой информации.
1.4	Числа. Стандартные операции	<b>Теория.</b> Стандартные операции с целыми и действительными числами. Стиль программирования Python (PEP 8). <b>Практика.</b> Решение простых задач в интерактивном режиме.
2.1	Логические выражения и операторы	<b>Теория.</b> Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python. <b>Практика.</b> Практическая работа 1 Логические выражения
2.2	Условный оператор	<b>Теория.</b> Условная и циклическая инструкции. Каскадность и вложенность алгоритмических инструкций. <b>Практика.</b> Решение задач по теме "Условные операторы"
2.4	Цикл while	<b>Теория.</b> Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума. <b>Практика.</b> Решение задач на анализ чисел потока и целочисленной арифметики
2.5	Цикл for	<b>Теория.</b> Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом. <b>Практика.</b> Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности.
3.1	Литералы строк	<b>Теория.</b> Понятие «литералы строк».

		<b>Практика.</b> Ввод-вывод строки. Решение задач на ввод строки, поиск подстроки.
3.2	Срезы строк	<b>Теория.</b> Форматирование строки. <b>Практика.</b> Преобразование строки. Решение упражнений.
3.3	Методы строк	<b>Теория.</b> Методы работы со строкой. <b>Практика.</b> Применение методов строки. Решение задач.
Практические занятия	<p>Примерный перечень практических работ:</p> <ol style="list-style-type: none"> <li>1. Написание простейших программ (нахождение суммы двух чисел, четность числа, произведение, разность и др.)</li> <li>2. Написание простейших программ имеющих линейную структуру</li> <li>3. Решение задач, на нахождение наибольшего/наименьшего из чисел и др.</li> <li>4. Написание программ нахождения суммы n-чисел и др.</li> <li>5. Задачи на использование строк и символьных переменных.</li> </ol> <p>Практические занятия проходят в форме тестирования, практикума, совместной продуктивной деятельности, моделирования, проблемном изложении материала, с помощью которого дети сами решают познавательные задачи.</p>	
Самостоятельная работа	Самостоятельная работа проводится в форме подготовки и разработки индивидуального/группового проекта к концу изучения курса.	

## 6. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

### 6.1. Материально-техническое обеспечение

Материально-техническое обеспечение дисциплины включает:

Учебные аудитории для проведения занятий лекционного типа	Доска, мультимедийные средства хранения, передачи и представления информации
Учебные аудитории для проведения практических занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	Доска, мультимедийный проектор, экран
Помещения для самостоятельной	Персональные компьютеры с пакетом

работы обучающихся	MS Office, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета
Помещение для хранения и профилактического обслуживания учебного оборудования	Стеллажи для хранения учебно-наглядных пособий и учебно-методической документации

## ***6.2. Информационное и учебно-методическое обеспечение***

Материально-техническое обеспечение:

- столы для компьютера;
- компьютерные стулья;
- шкафы для дидактических материалов, пособий;
- специальная и научно-популярная литература для педагога и учащихся;
- канцтовары;

Информационное обеспечение:

- персональный компьютер (на каждого участника);
- мультимедийный проектор;
- видеоматериалы разной тематики по программе;
- оргтехника;
- выход в сеть Internet;

Аппаратное обеспечение:

- Процессор не ниже Core2 Duo;
- Объем оперативной памяти не ниже 4 Гб DDR3;
- Дисковое пространство не менее 128 Гб;
- Монитор диагональю не менее 19”;

Программное обеспечение:

- Операционная система Windows 7 Профессиональная или выше;
- Интерпретатор Python версии 3.7 и выше;
- IDE Jupyter PyCharm;
- Foxit Reader или другой просмотрщик PDF файлов;
- WinRAR;
- Пакет офисных программ;
- Adobe Photoshop или другой растровый графический редактор;
- Любой браузер для интернет серфинга.

### **а) Основная литература:**

1. М. Лутц «Изучаем Питон», С-Пб, Вильямс, 2019.

2. Билл Любанович «Простой Python. Современный стиль программирования», С- Пб, Питер, 2019

3. Прохоренок, Н. А. Python 3. Самое необходимое./ Н. А. Прохоренок, В. А. Дронов. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2019. — 608 с.

4. Бэрри Пол. Изучаем программирование на Python. – М.: Эксмо, 2022. – 624 с.

5. Алеен Б. Дауни. Основы Python. Научитесь думать как программист. – М.: Манн, Иванов и Фербер, 2021. – 304 с.

**б) Дополнительная литература:**

1. Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2010.

2. Кольцов Д.М. Python Полное руководство. – СПб.: Наука и техника, 2022. – 480 с.

**в) Электронные и Internet-ресурсы:**

1. Официальный сайт языка Python <https://www.python.org/>

2. Материалы для изучения программирования на языке Python <http://kpolyakov.spb.ru/school/probook/python.htm>

3. Архив задач с проверяющей системой - <https://acmp.ru/>

4. Архив задач с проверяющей системой - <http://acm.timus.ru>

**6.3. Организация образовательного процесса**

Одним из важнейших условий реализации программы «Разработчик на Python: с нуля до junior» является активная позиция каждого слушателя, его инициатива, осмысление собственного опыта. В процессе изучения программы применяются как традиционные (объяснительно-иллюстративное, репродуктивно-воспроизводящее, предметно-ориентированное обучение), так и инновационные технологии обучения. Для достижения целей изучения программы используются активные (лекции, практическая работа с организацией совместной деятельности) и интерактивные формы проведения занятий (интерактивные лекции, взаимное рецензирование).

Предусмотрено широкое использование активных и интерактивных форм занятий (интерактивных лекций с использованием электронных образовательных ресурсов, дискуссий, разбора конкретных ресурсов, документов, ситуаций).

Процесс обучения в рамках реализации программы осуществляется с позиций обучения действием, данный курс дает знания и умения, а навыки отрабатываются на практике.

Использование приемов технологии проблемного обучения (создание проблемных ситуаций, активная самостоятельная деятельность) способствуют формированию у слушателей способности самостоятельного усвоения новых понятий и умения анализировать определенные действия.

Использование в рамках курсов повышения квалификации групповых технологий обучения позволяет решать задачи организации совместных действий, ведущих к активизации познавательных процессов; распределения начальных действий и операций; коммуникации, общения, без которых невозможны распределение, обмен и взаимопонимание и благодаря которым планируются адекватные задаче условия деятельности и выбор соответствующих способов действия.

## **7. ФОРМЫ КОНТРОЛЯ**

### **7.1. Входной контроль**

В начале изучения курса, дисциплины (модуля) проводится входной контроль знаний и умений с целью установления базового уровня знаний и умений обучающихся.

Оценочные средства входного контроля могут быть в виде тестов, проверочных упражнений, задач.

### **7.2. Текущий контроль**

Текущий контроль успеваемости учащихся и слушателей обеспечивает оперативное управление через преподавателя учебной деятельностью слушателей и её корректировку; стимулирует регулярную и целенаправленную работу слушателей, активизирует их познавательную деятельность и проводится в течение всего периода обучения.

К основным видам текущего контроля относятся: устный опрос, письменные работы, контроль с помощью технических средств и информационных систем.

Оценочные средства текущего контроля могут быть в виде тестов, проверочных упражнений, задач.

### **7.3. Итоговый зачёт**

Итоговый зачёт, как правило, проводится по результатам текущего контроля знаний слушателей, может быть проведён в виде специального зачётного контрольного мероприятия (теста; собеседования, интернет-тестирования и т.п.).

Условия, процедура подготовки и проведения итогового зачёта по отдельной дисциплине самостоятельно разрабатываются преподавателем, ведущим дисциплину.

Итоговый зачёт проводится за счёт объёма времени, выделенного преподавателю учебной нагрузкой.

Итоговый зачёт оценивается отметкой: «зачтено», «не зачтено».

Основные критерии оценки знаний, практических умений и владений обучающегося:

– отметка «зачтено» ставится обучающемуся, успешно занимавшемуся по данной дисциплине в период обучения и успешно прошедшему контрольное мероприятие;

– отметка «не зачтено» ставится обучающемуся, имеющему задолженности по результатам текущих аттестаций по данной дисциплине.

Результаты итогового зачёта выставляются в электронном журнале на образовательном портале «Интернет-лицей МГТУ» (<https://dpklms.magtu.ru>).

В случае получения отметки «не зачтено» обучающемуся предоставляется возможность *один раз повторно выполнить контрольное задание*.

## 8. ПЕРЕЧЕНЬ ТИПОВЫХ ЗАДАНИЙ

Вопросы для промежуточного контроля по усвоению материала
<p>Составьте выражение для вычисления в интерпретаторе Python 3 и вставьте в поле ответа результат вычисления:</p> <p style="text-align: center;">11111·1111111</p> <p>— произведение чисел 11111 (5 единиц) и 1111111 (7 единиц)</p>
<p>Запишите число 1.2345e3 в виде десятичной дроби.</p>
<p>Составьте и запишите выражение для вычисления:</p> <p style="text-align: center;">2014.0^14(Возвестив 14 степень)</p> <p>Обратите внимание на запись числа: это вещественное число.</p>
<p>Приведите к целому типу число 2.99</p>
<p>Расставьте скобки в выражении</p> <p style="text-align: center;">a and b or not a and not b</p> <p>в соответствии с порядком вычисления выражения (приоритетом операций). Всего потребуется 5 пар скобок (внешние скобки входят в их число).</p>
<p>Найдите результат выражения для заданных значений <i>a</i> и <i>b</i> Учитывайте регистр символов при ответе.</p> <p><b>a = True</b> <b>b = False</b> <b>a and b or not a and not b</b></p>
<p>Отметьте выражения, значения которых равны True:</p> <ul style="list-style-type: none"><li>• "239" &lt; "30" and 239 &lt; 30</li><li>• "239" &lt; "30" and 239 &gt; 30</li><li>• "239" &gt; "30" and 239 &lt; 30</li><li>• "239" &gt; "30" and 239 &gt; 30</li></ul>
<p>Укажите результат выражения: "123" + "42"</p>
<p>Какое значение будет у переменной <i>i</i> после выполнения фрагмента программы?</p> <pre>i = 0 while i &lt;= 10:     i = i + 1     if i &gt; 7:         i = i + 2</pre>
<p>Сколько итераций цикла будет выполнено в этом фрагменте программы?</p> <pre>i = 0 while i &lt;= 10:     i = i + 1     if i &gt; 7:</pre>

```
i = i + 2
```

Сколько всего знаков \* будет выведено после исполнения фрагмента программы:

```
i = 0
while i < 5:
    print('*')
    if i % 2 == 0:
        print('**')
    if i > 2:
        print('***')
    i = i + 1
```

Определите, какое значение будет иметь переменная i после выполнения следующего фрагмента программы:

```
i = 0
s = 0
while i < 10:
    i = i + 1
    s = s + i
    if s > 15:
        break
    i = i + 1
```

Определите, какое значение будет иметь переменная i после выполнения следующего фрагмента программы:

```
i = 0
s = 0
while i < 10:
    i = i + 1
    s = s + i
    if s > 15:
        continue
    i = i + 1
```

### Задачи для промежуточного контроля по усвоению материала

Напишите простой калькулятор, который считывает с пользовательского ввода три строки: первое число, второе число и операцию, после чего применяет операцию к введённым числам ("первое число" "операция" "второе число") и выводит результат на экран.

Поддерживаемые операции: +, -, /, \*, mod, pow, div, где  
mod — это взятие остатка от деления,  
pow — возведение в степень,  
div — целочисленное деление.

Если выполняется деление и второе число равно 0, необходимо выводить строку "Деление на 0!".

Обратите внимание, что на вход программе приходят вещественные числа.

Жители страны Малевии часто экспериментируют с планировкой комнат. Комнаты бывают треугольные, прямоугольные и круглые. Чтобы быстро вычислять жилплощадь, требуется написать программу, на вход которой подаётся тип фигуры комнаты и соответствующие параметры, которая бы выводила площадь получившейся комнаты. Для числа  $\pi$  в стране Малевии используют значение 3.14.

Формат ввода, который используют Малевийцы:

Треугольник

a

b

c

где a, b и c — длины сторон треугольника

прямоугольник

a

b

где a и b — длины сторон прямоугольника

круг

r

где r — радиус окружности

Напишите программу, которая получает на вход три целых числа, по одному числу в строке, и выводит на консоль в три строки сначала максимальное, потом минимальное, после чего оставшееся число.

На ввод могут подаваться и повторяющиеся числа.

Паша очень любит кататься на общественном транспорте, а получая билет, сразу проверяет, счастливый ли ему попался. Билет считается счастливым, если сумма первых трех цифр совпадает с суммой последних трех цифр номера билета.

Однако Паша очень плохо считает в уме, поэтому попросил вас написать программу, которая проверит равенство сумм и выведет "Счастливый", если суммы совпадают, и "Обычный", если суммы различны.

На вход программе подаётся строка из шести цифр. (Пример:123321)

<p>Выводить нужно только слово "Счастливый" или "Обычный", с большой буквы.</p>
<p>Напишите программу, которая считывает с консоли числа (по одному в строке) до тех пор, пока сумма введённых чисел не будет равна 0 и <b>сразу после этого</b> выводит сумму квадратов всех считанных чисел.</p> <p>Гарантируется, что в какой-то момент сумма введённых чисел окажется равной 0, <b>после этого считывание продолжать не нужно.</b></p> <p>В примере мы считываем числа 1, -3, 5, -6, -10, 13; в этот момент замечаем, что сумма этих чисел равна нулю и выводим сумму их квадратов, не обращая внимания на то, что остались ещё не прочитанные значения.</p>
<p>Напишите программу, которая выводит часть последовательности 1 2 2 3 3 3 4 4 4 4 5 5 5 5 ... (число повторяется столько раз, чему равно). На вход программе передаётся неотрицательное целое число <math>n</math> — столько элементов последовательности должна отобразить программа. На выходе ожидается последовательность чисел, записанных через пробел в одну строку.</p> <p>Например, если <math>n = 7</math>, то программа должна вывести 1 2 2 3 3 3 4.</p>
<p>Выведите таблицу размером <math>n \times n</math>, заполненную числами от 1 до <math>n^2</math> по спирали, выходящей из левого верхнего угла и закрученной по часовой стрелке, как показано в примере (здесь <math>n=5</math>)</p>

### Пример практической работы (проводится после усвоения материала каждого модуля)

1. Напишите программу, которая выводит на экран текст «Здравствуй, мир!» (без кавычек).
2. В популярном сериале «Остаться в живых» использовалась последовательность чисел 4 8 15 16 23 42, которая принесла героям удачу и помогла сорвать джекпот в лотерее. Напишите программу, которая выводит данную последовательность чисел с одним пробелом между ними.
3. Измените предыдущую программу так, чтобы каждое число последовательности 4 8 15 16 23 42 печаталось на отдельной строке.
4. Напишите программу, которая выводит указанный квадрат, состоящий из звездочек (\*).

```

* * * *
*      *
*      *
* * * *

```

5. На вход программе подается строка текста – имя человека. Напишите программу, которая выводит на экран приветствие в виде слова «Привет» (без кавычек), после которого должна стоять запятая и пробел, а затем введенное имя

6. На вход программе подается строка текста – название футбольной команды. Напишите программу, которая повторяет ее на экране со словами « - чемпион!» (без кавычек).

*Пример: Барселона - чемпион! (Барселона введено с клавиатуры через команду input)*

7. Напишите программу, которая считывает три строки по очереди, а затем выводит их в той же последовательности, каждую на отдельной строчке. (Текст может быть совершенно любой)

*Пример:*

Привет

Привет

Привет

8. Напишите программу, которая считывает три строки по очереди, а затем выводит их в обратной последовательности, каждую на отдельной строчке.

9. Напишите программу, которая считывает три целых числа и выводит на экран их сумму. Каждое число записано в отдельной строке.

10. Напишите программу, которая считывает целое число, после чего на экран выводится следующее и предыдущее целое число с пояснительным текстом.

*Пример:*

*Введено число 10*

*Выводится:*

*Следующее за числом 20 число: 21*

*Для числа 20 предыдущее число: 19*

11. Напишите программу, которая считает стоимость трех компьютеров, состоящих из монитора, системного блока, клавиатуры и мыши.

Формат входных данных

На вход программе подаётся четыре целых числа, каждое на отдельной строке. В первой строке — стоимость монитора, во второй строке — стоимость системного блока, в третьей строке — стоимость клавиатуры и в четвертой строке — стоимость мыши.

Формат выходных данных

Программа должна вывести одно число – стоимость покупки (трех компьютеров).

12. Напишите программу, в которой вычисляется сумма, разность и произведение двух целых чисел, введенных с клавиатуры.

Формат входных данных

На вход программе подаётся два целых числа, каждое на отдельной строке.

Формат выходных данных

Программа должна вывести сумму, разность и произведение введенных чисел, каждое на отдельной строке.

**Приложение 1**  
**Состав преподавателей, участвующих в реализации программы**

№	ФИО	Место работы, должность, ученое звание	Дата рождения	Реализуемые программы	Наименование группы	Образование (высшее/высшее педагогическое/среднее профессиональное)
1	Кагарманова Лира Альфировна	МГТУ им.Г.И.Носова, ст.лаборант БИиИТ	13.01.1999	Разработчик Python: с нуля до junior	Python-22-1	высшее педагогическое