

Департамент образования администрации Владимирской области
Государственное автономное образовательное учреждение дополнительного
профессионального образования Владимирской области «Владимирский институт
развития образования имени Л.И. Новиковой»

Кафедра цифрового образования и информационной безопасности



«УТВЕРЖДАЮ»


2021

**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА-
ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ**

Основы программирования на языке Python

Владимир

2021

Организация - разработчик: ГАОУ ДПО ВО «Владимирский институт развития образования имени Л.И. Новиковой»

Составители (разработчики):

Мишин Д.В., к.т.н., заведующий кафедрой цифрового образования и информационной безопасности ГАОУ ДПО ВО ВИРО

Мишина И.Ю., доцент кафедры цифрового образования и информационной безопасности ГАОУ ДПО ВО ВИРО

Программа рекомендована кафедрой цифрового образования и информационной безопасности ГАОУ ДПО ВО ВИРО к использованию в учебном процессе для повышения квалификации педагогов образовательных организаций основного общего, среднего общего и дополнительного образования, в частности, учителей информатики и ИКТ.

Протокол №22 от «3» марта 2021 г.

Зав.кафедрой

Д.В.Мишин

I. Общая характеристика программы

1.1. Нормативно-правовые основания разработки программы

Нормативную правовую основу разработки программы составляют:

- Федеральный закон от 29.12.2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Федеральный закон от 27.07.2006 г. № 149-ФЗ «Об информации, информационных технологиях по защите информации»;
- Федеральный закон от с 01.01.2008 г. № 152-ФЗ РФ «О персональных данных»;
- Федеральный закон от 29.12.2010 № 436-ФЗ (ред. от 28.07.2012) "О защите детей от информации, причиняющей вред их здоровью и развитию";
- Федеральный закон от 27.07.2010 г. № 210 «Об организации предоставления государственных и муниципальных услуг»
- Доктрина информационной безопасности Российской Федерации от 05.12.2016 № 646 (утверждённая указом Президента РФ);
- Указ Президента РФ от 09.05.2017 № 203 "О Стратегии развития информационного общества в Российской Федерации на 2017 - 2030 годы";
- Государственная программа РФ «Развитие образования» на 2018-2025 гг. (Утверждено постановлением правительства РФ 26.12.2017.№1642);
- Паспорт национального проекта «Образование» (УТВЕРЖДЕН президиумом Совета при Президенте Российской Федерации по стратегическому развитию и национальным проектам (протокол от 3 сентября 2018 г. №10);
- Постановление Правительства РФ от 26.12.2017 г. № 1642 "Об утверждении государственной программы Российской Федерации "Развитие образования";
- Распоряжение Правительства Российской Федерации от 25.10.2014 г. № 2125-р «Об утверждении Концепции создания единой федеральной межведомственной системы учета контингента обучающихся по основным образовательным программам и дополнительным общеобразовательным программам»;
- Распоряжение Правительства РФ от 14.02.2015 г. №236-р «Об утверждении плана мероприятий ("дорожной карты") по созданию единой федеральной межведомственной системы учета контингента обучающихся по основным образовательным программам и дополнительным общеобразовательным программам»;
- Приказ Минобрнауки России от 1.07.2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;
- Приказ Минобрнауки России от 15.01.2013 №10 «Федеральные государственные требования к минимуму содержания дополнительных профессиональных образовательных программ профессиональной переподготовки и повышения квалификации педагогических работников, а также к уровню профессиональной переподготовки педагогических работников»;
- Приказ Министерства образования и науки РФ от 23.08.2017 г. № 816 "Об утверждении Порядка применения организациями осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ";
- Приказ Минтруда России от 18.10.2013 N 544н (ред. от 05.08.2016) Об утверждении профессионального стандарта "Педагог (педагогическая деятельность в сфере дошкольного, начального общего, основного общего, среднего общего образования) (воспитатель, учитель);
- Приказ Министерства образования и науки РФ от 6.10.2009 г. N 373 "Об утверждении и введении в действие федерального государственного образовательного стандарта начального общего образования" (с изменениями и дополнениями);
- Приказ Министерства образования и науки РФ от 17.12.2010 г. № 1897 "Об утверждении и введении в действие федерального государственного образовательного стандарта основного общего образования» (с изменениями и дополнениями);

- Приказ Министерства образования и науки РФ от 17.05 2012 г. № 413 "Об утверждении и введении в действие федерального государственного образовательного стандарта среднего общего образования» (с изменениями и дополнениями);
- Приказ департамента образования администрации Владимирской области от 7.05.2014 г. № 675 «О введении в рабочую эксплуатацию АИС «Информационный портал системы образования Владимирской области»;
- Приказ департамента образования администрации Владимирской области от 31.12.2014 г. № 1688 «Об утверждении Концепции создания и развития регионального информационного портала Владимирской области»;
- Методические рекомендации по разработке основных профессиональных образовательных программ и дополнительных профессиональных программ с учетом соответствующих профессиональных стандартов (утв. Минобрнауки России от 22.01.2015 г. № ДЛ-1/ 05 ВН);
- Методические рекомендации-разъяснения по разработке основных профессиональных образовательных программ и дополнительных профессиональных программ на основе профессиональных стандартов (письмо Минобрнауки России от 22.04.2015 г. № ВК-1030/ 06);
- Государственная программа Владимирской области «Информационное общество (2014-2020 годы)»;
- СанПиН 2.4.2.2821-10 «Санитарно-эпидемиологические требования к условиям и организации обучения в общеобразовательных учреждениях».

Локальные акты

- Положение об итоговой аттестации слушателей по программам повышения квалификации в ГАОУ ДПО ВО ВИРО.
- Положение об организации дополнительного профессионального образования слушателей ГАОУ ДПО ВО ВИРО.

1.2. Область применения программы

Настоящая программа предназначена для повышения квалификации педагогов образовательных организаций основного общего, среднего общего и дополнительного образования, в частности, учителей информатики и ИКТ, ориентированных на обучение детей программированию на языке Python, их профориентацию и решение задач, связанных с разработкой программного обеспечения.

1.3. Требования к обучающимся

Высшее профессиональное образование или среднее профессиональное образование по направлениям подготовки «Образование и педагогика» или в области, соответствующей преподаваемому предмету, либо высшее профессиональное образование или среднее профессиональное образование и дополнительное профессиональное образование по направлению деятельности в образовательной организации.

1.4. Цель и планируемые результаты освоения программы

Повышение уровня предметной и профессиональной компетентности педагогов образовательных организаций, обучение педагогов основам языка программирования Python.

Обучающийся в результате освоения программы должен владеть:

Код ТФ	ТФ	Практический опыт (Трудовые действия)	Умения	Знания
A/01.6	Педагогическая деятельность по проектированию и реализации образовательного процесса в образовательных организациях дошкольного, начального общего, основного общего, среднего общего образования Общепедагогическая функция. Обучение	– Формирование навыков, связанных с информационно-коммуникационными технологиями (далее - ИКТ)	– Устанавливать интерпретатор и среды разработки языка высокого уровня Python – Разрабатывать простые программы на языке высокого уровня Python, содержащие циклы, условия и другие управляющие конструкции – Работать со структурированными типами данных и файлами средствами Python – Подключать дополнительные модули и пакеты Python	– Место языка высокого уровня Python в современном программировании – Особенности синтаксиса и лексики Python – Типы данных, операторы, особенности работы с переменными – Особенности и целесообразность применения подпрограмм, работы с файловыми объектами –

Сокращения:

ОО – образовательная организация

1.5. **Форма обучения:** с применением ЭО и ДОТ.

Срок обучения: 36 часов.

Режим занятий: индивидуально в соответствии с графиком курса.

1.6. **Форма документа, выдаваемого по результатам освоения программы:** лицам, успешно освоившим программу и прошедшим итоговую аттестацию, выдается сертификат о повышении квалификации.

2. Учебный план

№ п.п	Наименование разделов	Инвариантная часть	Вариативная часть	Всего	Форма аттестации
1.	Модуль 1. Введение в Python	4	-	4	Тестирование, Практическая работа 1
2.	Модуль 2. Структура программы	2	-	2	Тестирование
3.	Модуль 3. Переменные, типы данных и операторы Python	8	-	8	Тестирование, Практическая работа 2
4.	Модуль 4. Управляющие структуры Python	8	-	8	Тестирование, Практическая работа 3
5.	Модуль 5. Структурированные типы данных Python	6	-	6	Тестирование, Практическая работа 4
6.	Модуль 6. Подпрограммы в Python	4	-	4	Тестирование, Практическая работа 5
7.	Модуль 7. Файлы и модули в Python	4	-	4	Тестирование,

					Практическая работа 6
	Итоговая аттестация				Зачет по совокупности выполненных практических работ и тестов
	Итого	36	-	36	

3. Календарный учебный график

Компоненты программы	Модули						
	1 м	2 м	3 м	4 м	5 м	6 м	7 м
Модуль 1-7	Л/П	Л	Л/П	Л/П	Л/П	Л/П	Л/П
Практика (учебная)	-	-	-	-	-	-	-
Итоговая аттестация							+

4. Рабочие программы учебных модулей

Наименование модулей, практики, тем программы	Вид учебного занятия	Содержание учебного материала	Всего часов
<u>Наименование компонента программы:</u> <i>Модуль 1. Введение в Python</i>			
Тема 1.1. Знакомство с языком высокого уровня Python	Лекция с/р	История языка Python, отличие Python 2 и Python 3, статистика, преимущества и недостатки Python, философия Python, решаемые с помощью Python задачи, использующие Python проекты, Python на олимпиадах и ЕГЭ.	1
Тема 1.2. Инструменты для работы с кодом на Python	Лекция с/р	Онлайн платформы, требования для работы с кодом на Python на персональном компьютере, установка интерпретатора Python в Windows и GNU/Linux, установка среды разработки; требования для работы с кодом на Python на мобильном устройстве.	1
Тема 1.3. Начало работы с Python	Лекция с/р	Режимы работы Python: интерактивный и программный, первые программы, получение помощи, правила оформления кода (PEP 8). Прохождение теста модуля 1. Выполнение практической работы модуля 1.	1
	Практическое занятие с/р		1
<u>Наименование компонента программы:</u> <i>Модуль 2. Структура программы</i>			
Тема 2.1. Лексика и синтаксис Python	Лекция с/р	Алфавит Python, физические и логические строки, структура строк и отступы, лексический анализ, лексическая структура программы на Python, синтаксическая структура.	1

Тема 2.1. Структура программы на языке Python	Лекция с/р	Модульное, структурное и объектно-ориентированное программирование и Python, конструкции в Python, концептуальная иерархия. Прохождение теста модуля 2.	1
	Практическое занятие с/р		
<u>Наименование компонента программы:</u> <i>Модуль 3. Переменные, типы данных и операторы Python</i>			
Тема 3.1. Объекты в Python. Переменные.	Лекция с/р	Объекты в Python, общие функции, переменные, инструкция присваивания, динамическая типизация, управление памятью и сборка мусора. Выполнение практической работы модуля 3	1
	Практическое занятие с/р		1
Тема 3.2. Типы данных Python.	Лекция с/р	Классификация типов данных Python, проверка типов, скалярные типы, коллекции, строки и кратко о других последовательностях, кратко о множествах и отображениях, взаимное преобразование типов. Выполнение практической работы модуля 3 (продолжение)	1
	Практическое занятие с/р		2
Тема 3.3. Операторы Python	Лекция с/р	Арифметические операторы, операторы сравнения, логические операторы, побитовые операторы, операторы присваивания, поверхностное и глубокое копирование, операторы принадлежности, операторы тождественности, приоритет операторов, ассоциативность, изменение порядка вычисления, оператор доступа к атрибутам и методам, ввод / вывод. Выполнение практической работы модуля 3 (продолжение) Прохождение теста модуля 3.	1
	Практическое занятие с/р		2
<u>Наименование компонента программы:</u> <i>Модуль 4. Управляющие структуры Python</i>			
Тема 4.1. Условия	Лекция	Управляющие структуры, условная инструкция, её синтаксис, отступы, варианты использования, короткая запись, инструкция pass, вложенные условия, примеры программ. Выполнение практической работы модуля 4.	1
	Практическое занятие		2
Тема 4.2. Циклы	Лекция	Цикл с предусловием, его синтаксис, бесконечный цикл, короткая запись цикла while, цикл по коллекциям, его синтаксис, примеры использования, прерывание и продолжение циклов (break, continue). Выполнение практической работы модуля 4 (продолжение).	1
	Практическое занятие		2
Тема 4.3. Комбинация циклов и условий	Лекция	Примеры применения комбинаций циклов и условий в программе Прохождение теста модуля 4.	1
	Практическое занятие		1

<u>Наименование компонента программы:</u>			
<i>Модуль 5. Структурированные типы данных Python</i>			
Тема 5.1. Последовательности	Лекция	Список, коллекционные включения, цикл по списку, вложенные списки, методы split и join, вывод списка на экран, работа со списками, копирование списка в Python, примеры задач со списками, кортеж, числовой диапазон, байтовые строки, байтовый массив. Выполнение практической работы модуля 5.	1
	Практическое занятие		1
Тема 5.2. Множества	Лекция	Множества в Python, создание множеств, операции над множествами. Выполнение практической работы модуля 5 (продолжение).	1
	Практическое занятие		1
Тема 5.3. Отображения	Лекция	Отображения: словари, создание словарей, операции со словарями. Прохождение теста модуля 5.	1
	Практическое занятие		1
<u>Наименование компонента программы:</u>			
<i>Модуль 6. Подпрограммы в Python</i>			
Тема 6.1. Основные понятия. Функции в Python	Лекция	Подпрограммы, функции в Python, позиционные и ключевые параметры/ аргументы в Python, упаковка и распаковка аргументов, передача аргументов по значению и по ссылке, область видимости, возврат нескольких значений, строки документации, анонимные функции, побочный эффект и чистые функции, время выполнения функции. Выполнение практической работы модуля 6.	1
	Практическое занятие		1
Тема 6.2. Рекурсия	Лекция	Примеры применения рекурсии в программе Прохождение теста модуля 6.	1
	Практическое занятие		1
<u>Наименование компонента программы:</u>			
<i>Модуль 7. Файлы и модули в Python</i>			
Тема 7.1. Файловый объект в Python. Работа с файлами в Python	Лекция	Файлы и целесообразность их применения, файловый объект в Python, работа с файлами в Python: основные свойства и методы, простое чтение и запись, чтение файла целиком. Выполнение практической работы модуля 7.	1
	Практическое занятие		1
Тема 7.2. Модули и пакеты в Python и их особенности. Собственные модули в Python	Лекция	Модули и пакеты в Python, подключение и использование, область поиска, особенности модулей в Python, собственные модули, использование модуля: запуск или импорт. Прохождение теста модуля 7.	1
	Практическое занятие		1

5. Организационно-педагогические условия реализации программы

5.1. Организация образовательного процесса

Реализация программы подразумевает наличие базового уровня ИКТ компетентности слушателей, знакомство с базовыми технологиями программирования и алгоритмизации.

Программой предусмотрена итоговая аттестация по результатам выполнения практических работ и прохождения тестов модулей.

Индивидуальные и групповые консультации проходят при непосредственном общении преподавателя и обучающихся средствами СДО.

5.2. Материально-техническое обеспечение

Реализация программы требует наличия:

- технических средств обучения: компьютер, подключенный к сети Интернет (для применения ДОТ), актуальные версии браузера.

- возможность установки интерпретатора Python в Windows или GNU/Linux.

5.3. Информационное обеспечение обучения

Основные источники:

1. Лутц М. Изучаем Python. 5-е издание. - Пер. с англ. - Диалектика, 2019 - 2020.
2. Марк Саммерфилд. Программирование на Python 3. Подробное руководство - 2009.
3. Дэвид Бизли. Python. Подробный справочник. 4-е издание. - М.: Символ-Плюс, 2010.
4. Доусон М. Програмируем на Python. - СПб.: Питер, 2019.
5. Swaroop С Н, A Byte of Python. - это свободная книга по программированию на языке Python. В переводе Владимира Смоляра: <http://wombat.org.ua/AByteOfPython/>.
6. Брайсон Пэйн. Python для детей и родителей. Играй и программируй. - М.: 2017.

Интернет ресурсы:

1. Сайт для обучения программированию на Python, позволяющий писать и запускать программы прямо в браузере: <http://pythontutor.ru>
2. Сайт дистанционной подготовки, содержащий множество задач самого различного уровня, от простейших до международных олимпиад (для решения задач необходима регистрация): <http://informatics.mccme.ru>. Имеется автоматическая система проверки с поддержкой различных языков, в том числе Python.
3. Крупнейший в России архив задач по программированию с автоматической проверяющей системой Timus Online Judge. Основной источник задач для архива — соревнования Уральского федерального университета, Чемпионаты Урала, Уральские четвертьфиналы ICPC, Петрозаводские сборы по программированию. <https://acm.timus.ru/>
4. Сайт с материалами для изучения программирования на языке Python на основе учебника «Информатика. Углублённый уровень» для 10-11 классов К.Ю. Полякова и Е.А. Еремина. Здесь размещены варианты глав учебника по программированию, написанные на основе языка Python, презентации и отлаженные примеры всех программ, рассмотренных в учебнике. <http://kpolyakov.spb.ru/school/probook/python.htm>
5. Учебник Фоксфорда - <http://foxford.ru/wiki/informatika/o-yazyke-python>

6. Курсы Stepik <https://stepik.org/course/67>, <https://stepik.org/course/58638> и др.
7. Сайт для обучения программированию на Python, позволяющий писать и запускать программы прямо в браузере: <http://pythontutor.ru>
8. Сайт дистанционной подготовки, содержащий множество задач самого различного уровня, от простейших до международных олимпиад (для решения задач необходима регистрация): <http://informatics.mccme.ru>. Имеется автоматическая система проверки с поддержкой различных языков, в том числе Python.
9. Крупнейший в России архив задач по программированию с автоматической проверяющей системой Timus Online Judge. Основной источник задач для архива — соревнования Уральского федерального университета, Чемпионаты Урала, Уральские четвертьфиналы ICPC, Петрозаводские сборы по программированию. <https://acm.timus.ru/>
10. Сайт с материалами для изучения программирования на языке Python на основе учебника «Информатика. Углублённый уровень» для 10-11 классов К.Ю. Полякова и Е.А. Еремина. Здесь размещены варианты глав учебника по программированию, написанные на основе языка Python, презентации и отлаженные примеры всех программ, рассмотренных в учебнике. <http://kpolyakov.spb.ru/school/probook/python.htm>
11. Учебник Фоксфорда - <http://foxford.ru/wiki/informatika/o-yazyke-python>
12. Курсы Stepik <https://stepik.org/course/67>, <https://stepik.org/course/58638> и др.

5.4. Кадровое обеспечение образовательного процесса

Педагогические работники, реализующие дополнительную профессиональную программу, должны удовлетворять квалификационным требованиям, указанным в квалификационных справочниках по соответствующим должностям.

6. Контроль и оценка результатов освоения программы

6.1. Промежуточная аттестация:

промежуточная аттестация слушателей предусматривает прохождение тестирований и выполнение практических работ по результатам изучения модулей.

Для успешного завершения курса обучающимся необходимо выполнить от 60% заданий практических работ и дать не менее 70% правильных ответов в рамках каждого теста. **(Приложение 1. Примеры заданий практических работ модулей №№1-7; Приложение 2 Примеры тестовых вопросов модулей №№1-7)**

6.2. Итоговая аттестация:

итоговая аттестация слушателей проводится после освоения всех модулей программы в форме зачета по совокупности положительных результатов выполнения практических работ и прохождения промежуточных тестов. Оценивание: «зачет/незачет».

<p>Результаты Сформированы навыки, связанные с решением профессиональных задач работы с современными языками программирования на примере Python с целью подготовки учащихся к сдаче ЕГЭ и участию в олимпиадах по информатике:</p>	<p>Основные показатели оценки результата В ходе тестирования проходит проверка полученных знаний: – В области применимости языка Python для решения широкого круга задач разработки программного обеспечения; – В области средств разработки на языках высокого уровня; – В области структуры типовой программы на языке Python, применению</p>
---	--

	<p>типов данных и операторов;</p> <ul style="list-style-type: none">– В области управляющих структур и конструкций языка Python;– В области функций и файловых объектов Python.
--	--

Приложение 1 – примеры заданий Практических работы

Модуль 3. Переменные, типы данных и операторы (20 заданий по написанию небольших программ)

Задание 1. В интерактивном режиме: Переменной `var_int` присвойте значение 100, `var_float` - значение 27.4, `var_str` - "La". Значение, хранимое в переменной `var_int`, увеличьте в 4.3 раза. Полученный результат свяжите с переменной `var_big`. Измените значение, хранимое в переменной `var_float`, уменьшив его на единицу, результат свяжите с той же переменной. Разделите `var_int` на `var_float`, а затем `var_big` на `var_float`. Результат данных выражений не привязывайте ни к каким переменным. Измените значение переменной `var_str` на "TuTuTuLaLaLa". При формировании нового значения используйте операции конкатенации (+) и повторения строки (*). Выведите значения всех переменных с пояснением, один раз используя `print()`. Сделайте подтверждающий скриншот работы в интерактивном режиме.

Задание 2. Составьте программу, которая запрашивает у пользователя 2 целых числа и выполняет операции: - арифметические: +, -, *, /, //, %, **; - сравнения: >, >=, !=, ==, выводя на экран результат каждого действия. В случае получения вещественного результата, округлите его до 2-х знаков после запятой (используя функцию `round()`).
Пример входных данных: 2 3 Пример выходных данных: 5 -1 6 0.67 0 2 8 True True False False True False

Задание 3. Напишите программу, которая считывает длины двух катетов в прямоугольном треугольнике и выводит его площадь. Каждое число записано в отдельной строке. 5
Пример входных данных: 3 5 Пример выходных данных: 7.5

Модуль 4. Управляющие структуры (16 заданий по написанию небольших программ)

1 - Напишите программу, которая считывает целые числа с консоли по одному числу в строке. Для каждого введённого числа проверить: – если число меньше 10, то пропускаем это число; – если число больше 100, то прекращаем считывать числа; – в остальных случаях вывести это число обратно на консоль в отдельной строке.

2. Требуется определить, является ли данный год високосным. Напомним, что високосными годами считаются те годы, порядковый номер которых либо кратен 4, но при этом не кратен 100, либо кратен 400 (например, 2000-й год являлся високосным, а 2100-й будет невисокосным годом). Программа должна корректно работать на числах $1900 \leq n \leq 3000$. Выведите "Високосный" в случае, если введённый пользователем год является високосным и "Обычный" в обратном случае. Пример входных данных: 2100
Пример выходных данных: 2 Обычный Пример входных данных: 2000 Пример выходных данных: Високосный

3. Заданы две клетки шахматной доски. Если они покрашены в один цвет, то выведите слово "YES", а если в разные цвета — то "NO". Программа получает на вход четыре числа от 1 до 8 каждое, задающие номер столбца и номер строки сначала для первой клетки, потом для второй клетки. Пример входных данных: 1 1 2 3 6 Пример выходных данных: YES

Модуль 6. Подпрограммы (5 заданий по написанию небольших программ)

1) Напишите функцию `modify_list(lst)`, которая принимает на вход список целых чисел, удаляет из него все нечётные значения, а чётные нацело делит на два. Функция не должна ничего возвращать, требуется только 2 изменение переданного списка. Функция не должна осуществлять ввод/вывод информации. После объявления функции запишите следующий код: `lst = [1, 2, 3, 4, 5, 6] print(modify_list(lst)) print(lst) modify_list(lst) print(lst)`
`lst = [10, 5, 8, 3] modify_list(lst) print(lst)` Результат: None [1, 2, 3] [1] [5, 4]

2) Группа биологов в институте биоинформатики завела себе черепашку. После дрессировки черепашка научилась понимать и запоминать указания биологов следующего вида: север 10 запад 20 юг 30 восток 40 где первое слово — это направление, в котором должна двигаться черепашка, а число после слова — это положительное расстояние в сантиметрах, которое должна пройти черепашка. Но команды даются быстро, а черепашка ползёт медленно, и программисты догадались, что можно написать программу, которая определит, куда в итоге биологи приведут черепашку. Для этого программисты просят вас написать программу, которая выведет точку, в которой окажется черепашка после всех команд. Для простоты они решили считать, что движение начинается в точке (0, 0), и движение на восток увеличивает первую координату, а на север — вторую. 4 Программе подаётся на вход число команд n , которые нужно выполнить черепашке, после чего n строк с самими командами. Вывести нужно два числа в одну строку: первую и вторую координату конечной точки черепашки. Все координаты целочисленные. Решение необходимо оформить с помощью функций! Пример входных данных: 4 север 10 запад 20 юг 30 восток 40 Пример выходных данных: 20 -20

Модуль 7. Файлы и модули (4 задания по написанию небольших программ)

1) Напишите программу, которая считывает текст из файла (в файле может быть больше одной строки) и выводит в новый файл самое частое слово в этом тексте и через пробел то, сколько раз оно встретилось. Если таких слов несколько, вывести лексикографически первое (можно использовать оператор $<$ для строк). Слова, написанные в разных регистрах, считаются одинаковыми. Пример входных данных (содержимое файла): abc a bCd bC AbC BC BCD bcd ABC Пример выходных данных (содержимое файла): abc 3

2) Имеется файл с данными по успеваемости абитуриентов. Он представляет из себя набор строк, где в каждой строке записана следующая информация:
 Фамилия;Оценка_по_математике;Оценка_по_физике;Оценка_по_русскому_языку
 Поля внутри строки разделены точкой с запятой (после ; нет пробела), оценки — целые числа. Для разбиения строки на части по символу ';' можно использовать метод split. 2 Напишите программу, которая считывает исходный файл с подобной структурой и для каждого абитуриента записывает его среднюю оценку по трём предметам на отдельной строке, соответствующей этому абитуриенту, в файл с ответом. Также вычислите средние баллы по математике, физике и русскому языку по всем абитуриентам и добавьте полученные значения, разделённые пробелом, последней строкой в файл с ответом. Пример входных данных (содержимое файла): Петров;85;92;78 Сидоров;100;88;94 Иванов;58;72;85 Пример выходных данных (содержимое файла): 85.0 94.0 71.666666667 81.0 84.0 85.666666667

Приложение 2 – примеры вопросов тестов

Модуль 1. Введение в Python (18 вопросов. 30 минут)

1) Отступы расставлены верно?

```
number = 23
```

```
guess = int(input('Введите целое число : '))
```

```
if guess == number:
```

```
    print("Поздравляю, вы угадали.") # Здесь начинается новый блок
```

```
        print('(хотя и не выиграли никакого приза!))' # Здесь заканчивается новый блок
```

```
elif guess < number:
```

```
    print("Нет, загаданное число немного больше этого.") # Ещё один блок
```

```
    # Внутри блока вы можете выполнять всё, что угодно ...
```

```
else:
```

```
    print("Нет, загаданное число немного меньше этого.")
```

```
    # чтобы попасть сюда, guess должно быть больше, чем number
```

Выберите один ответ:

Верно

Неверно

2) Python 3 несовместим с предыдущими версиями: программы, написанные на более ранних версиях языка, оказалось невозможно запускать на новой версии языка - Python 3.

Выберите один ответ:

Верно

Неверно

3)

В алфавит языка программирования Python входят (выберите все верные утверждения):

Выберите один или несколько ответов:

a. 10 арабских цифр от 0 до 9.

b. Латинские прописные буквы.

c. Буквы национальных алфавитов в Unicode.

d. Римские цифры.

e. Специальные символы (знак нижнего подчёркивания; знаки препинания; круглые, квадратные скобки; знаки арифметических операций, # -знак комментариев и др.)

f. Латинские строчные буквы.

Модуль 2. Структура программы 29 вопросов. 60 минут

1) Как можно разместить в программе на Python следующий текст из двух строк в качестве комментария?

```
Это комментарий
```

```
из двух строк
```

Выберите один или несколько ответов:

a.

```
# Это комментарий
```

```
# из двух строк
```

b.

```
# Это комментарий
```

```
из двух строк
```

c.

```
'''
```

```
Это комментарий
```

```
из двух строк
```

```
'''
```

d.

*# Это комментарий
из двух строк #*

e.

*'Это комментарий
из двух строк'*

2)

Python чувствителен к регистру, т.е.: BookId, bookID, Bookid, bookid, bookId и т. д., — разные идентификаторы.

Выберите один ответ:

Верно

Неверно

3)

Каков будет результат программы?

print("Ветер на море гуляет\nИ кораблик подгоняет;\nОн бежит себе в волнах\nНа раздутых парусах.")

Выберите один ответ:

a.

Ветер на море гуляет И кораблик подгоняет; Он бежит себе в волнах На раздутых парусах.

b. *Ветер на море гуляет*

И кораблик подгоняет;

Он бежит себе в волнах

На раздутых парусах.

c.

Ветер на море гуляет\nИ кораблик подгоняет;\nОн бежит себе в волнах\nНа раздутых парусах.

d. *Ветер на море гуляет*

И кораблик подгоняет;\nОн бежит себе в волнах

На раздутых парусах.

Модуль 4. Управляющие структуры (20 вопросов. 40 минут)

1. В Python отсутствует инструкция switch...case.

Выберите один ответ:

Верно

Неверно

2. Выберите неверное утверждение об условной инструкции.

Выберите один ответ:

a. Для каждой из частей существует ассоциированный с ней блок инструкций, которые выполняются в случае истинности соответствующего им условного выражения.

b. Завершается запись условной инструкции обязательной частью else (иначе).

c. Блок кода, соответствующий else, выполняется, если все условные выражения ложны (False).

d. За частью if может следовать одна или несколько необязательных частей elif с условными выражениями.

3. Определите результат работы программы.

```
number = -7
abs_number = number if number >= 0 else -number
print(abs_number)
```

Модуль 5. Структурированные типы данных (30 вопросов. 60 минут)

1) Как можно создать список?

Выберите один ответ:

- a. `s = list[1, 2, 3]`
- b. `s = list(1, 2, 3)`
- c. `s = [1, 2, 3]`
- d. Все варианты верны

2) Как получить ['bar', 'baz'] из списка

`a = ['foo', 'bar', 'baz', 'qux', 'quux']`

?

Выберите один ответ:

- a. `print(a[1], a[2])`
- b. `print(a[2:3])`
- c. `print(a[1:-2])`
- d. `print(a[2:4])`
- e. `print(a[-4:-3])`

3) Сколько элементов будет содержать список *students* после следующих операций?

`students = ['Ivan', 'Masha', 'Sasha']`

`students += ['Olga']`

`students += 'Olga'`

Введите в поле ответа одно число.

Модуль 6. Подпрограммы (22 вопроса. 60 минут)

1) Выберите основные способы передачи аргументов.

Выберите один или несколько ответов:

- a. По ссылке
- b. По имени
- c. По параметру
- d. По значению

2) Выберите недостаток использования подпрограмм.

Выберите один ответ:

- a. Скрытие деталей реализации от пользователей подпрограммы
- b. Накладные расходы на вызов подпрограммы
- c. Возможность абстрагироваться от природы рассматриваемых объектов
- d. Декомпозиция сложной задачи на несколько более простых подзадач
- e. Уменьшение дублирования кода и возможность повторного использования кода

3) Какое ключевое слово используется для возврата значения из функции?

Выберите один ответ:

- a. `postget`
- b. `answer`
- c. `get`
- d. `return`