

**Автономная некоммерческая организация
дополнительного профессионального образования**

«Центр дополнительного профессионального образования»

ПРИНЯТА

на заседании педагогического
совета АНОДПО «ЦДПО»,
протокол № 6 от 27.08.2019

УТВЕРЖДЕНА

приказом АНОДПО «ЦДПО»
№ 56 от 28.08.2019.

**ДОПОЛНИТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
технической направленности**

«Python. Прикладное программирование»

Возраст обучающихся: 15-17 лет.

Срок реализации: 1 год (108 часов).

Автор-разработчик: Евдокимова Н.И.,
педагог дополнительного образования

г. Выборг

2019 г.

I. Пояснительная записка

Дополнительная общеразвивающая программа технической направленности «Веб-мозаика» разработана на основе:

- Федерального закона от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации»,
- Концепции развития дополнительного образования детей (утверждена Распоряжением Правительства Российской Федерации от 04 сентября 2014 года № 1726-р),
- Приказа Министерства просвещения Российской Федерации от 9 ноября 2018 г. № 196 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».

Направленность общеразвивающей программы «Python. Прикладное программирование»: техническая.

Цель дополнительной общеразвивающей программы «Python. Прикладное программирование»:

- обучение слушателя основам объектно-ориентированного подхода в программировании на примере языка Python;
- формирование у слушателя навыка алгоритмического мышления;
- профориентирование слушателя, подготовка к осознанному выбору профессий в области ИКТ, связанных с программированием.

Задачи дополнительной общеразвивающей программы технической направленности «Python. Прикладное программирование»:

- обучить слушателя базовым теоретическим понятиям, лежащим в основе объектно-ориентированного подхода к разработке программного обеспечения
- изучить правила использования стандартных элементов интерфейса программных продуктов, работающих под Windows;
- изучить синтаксические конструкции языка Python, связанные с построением стандартными алгоритмическими конструкциями;
- познакомить с технологиями программирования и классическими алгоритмами для различных структур данных;
- научить приемам ведения проектной деятельности в сфере разработки программного обеспечения.

Актуальность дополнительной общеразвивающей программы технической направленности «Python. Прикладное программирование»

Знакомство школьников с технологиями программирования на востребованном языке с целью популяризации данных технологий. Возможность познакомиться с востребованной профессией и заложить основу для будущего специализированного образования.

Новизна дополнительной общеразвивающей программы технической направленности «Python. Прикладное программирование» заключается в том, что в процесс обучения включена проектная деятельность с использованием компьютерных технологий.

Педагогическая целесообразность: педагогическая целесообразность данной общеразвивающей программы состоит в возможности использования детьми приобретенных знаний в последующем освоении предметов общеобразовательных школ, а также в последующем профессиональном обучении.

Сроки реализации дополнительной общеразвивающей программы технической направленности «Python. Прикладное программирование»: 1 год.

На сегодняшний день объектно-ориентированный подход в программировании применяется в большинстве разработок программного обеспечения, как для Windows-приложений, так и для мобильных устройств и программируемых веб-проектов. Для изучения основ объектно-ориентированного программирования выбран язык Python, как наиболее доступный для

слушателей данного возраста. Впоследствии, опираясь на полученные знания, слушатели смогут осваивать более сложные языки и технологии.

Одновременно в программе уделяется внимание теории построения алгоритмов. Алгоритмический подход при решении задач на ЭВМ - это залог успешного применения средств вычислительной техники в различных областях человеческой деятельности и, в первую очередь, при разработке программного обеспечения. Умение разрабатывать алгоритмы — это основа процесса программирования, фундамент обучающегося в будущей успешной деятельности в качестве разработчика программного обеспечения. Умение алгоритмически мыслить принесет пользу школьнику не только в сфере узконаправленной деятельности программиста, но и будет способствовать общему интеллектуальному развитию, т.к. учит структурировать информацию, дисциплинирует мышление, способствует развитию памяти, обостряет внимание.

Содержание программы призвано содействовать профессиональному самоопределению обучающегося, реализации индивидуального потенциала, достижению сбалансированности между профессиональными интересами обучающегося, его психофизическими особенностями и потребностями рынка труда. Упор сделан на развитие у обучающихся творческого потенциала и самостоятельности.

Программа конкретизирует содержание предметных тем, дает распределение учебных часов по разделам курса и последовательность изучения тем и разделов.

Календарное планирование учебного процесса осуществляется с учетом даты начала занятий, каникулярного периода и реализуется в течение учебного года путем последовательного изложения тем в соответствии с учебным планом.

Категория обучающихся - учащиеся 9- 11-х классов.

Общее количество учебных часов обучения - 108 часов.

Форма обучения - очная.

Форма проведения занятий - аудиторные.

Режим занятий - один раз в неделю по 3 аудиторных часа. Продолжительность академического часа установлена в соответствии с требованиями СанПиНа (СанПиН 2.4.4.3172-14 "Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей").

Программа оснащена современными техническими средствами, учебно-методическими пособиями и электронными учебно-методическими материалами для обучающихся.

Условия реализации программы:

- Требования к исходному уровню подготовки учащихся – пользователь ПК под управлением ОС Windows..
- Условия формирования групп: разновозрастные.
- Допускается дополнительный набор в группу в соответствии с технологическим регламентом и с учетом санитарных норм.
- Количество детей в группе: не более 14 человек.
- Формы организации деятельности обучающихся на занятии: фронтальная.

Материально-техническое оснащение: программа оснащена современными техническими средствами, каждый обучающийся обеспечен отдельным рабочим местом, персональным компьютером с необходимым набором программ. Кабинет обеспечен видеосистемой для демонстрации работы с программным обеспечением.

II. Ожидаемые результаты и способы определения результативности

2.1. Личностные. У учащихся должны быть сформированы понятия о правовом использовании информационных технологий, соблюдении авторских прав, этике поведения в информационном пространстве (социальных сетях, форумах и т.п.).

Участие в конкурсах, презентациях позволит повысить социальную активность учащихся, а также их вовлеченность в социальную жизнь города.

2.2. **Метапредметные. Информационная компетенция.** При помощи реальных объектов (телевизор, магнитофон, телефон, факс, компьютер, принтер, модем, копир) и информационных технологий (аудио-, видеозапись, электронная почта, СМИ, Интернет) формируются умения самостоятельно искать, анализировать и отбирать необходимую информацию, организовывать, преобразовывать, сохранять и передавать ее. Данная компетенция обеспечивает навыки деятельности ученика по отношению к информации, содержащейся в учебных предметах и образовательных областях, а также в окружающем мире.

2.3. **Предметные.** В результате изучения программы обучающийся должен:

знать:

- базовую терминологию, связанную с программированием на языке Python;
- методы работы и настройки интегрированной среды разработки языка Python;
- базовые теоретические понятия, лежащие в основе объектно-ориентированного подхода к разработке программного обеспечения;
- основные элементы синтаксиса языка Python, классические алгоритмические конструкции, элементарные типы данных и способы их преобразований;
- правила использования стандартных элементов интерфейса программных продуктов, работающих под Windows;
- методики разработки, набора и отладки программных модулей;
- технологии программирования и классические алгоритмы для различных структур данных.

уметь:

- разработать прикладной модуль, работающую в операционной системе Windows. Для этого, используя стандартные элементы управления, разработать интерфейс программы, разработать алгоритм работы программы, реализовать его на языке Python, отладить приложение, используя средства IDE Eclipse PyDef.
- вести проектную деятельность по разработке программного обеспечения начиная от разработки алгоритма, заканчивая отладкой, оформлением сопровождающей документации и защитой проекта.

Система оценки результатов освоения общеразвивающей программы:

- текущий контроль успеваемости
- промежуточная аттестация.

Реализации данной программы являются: самостоятельное выполнение творческих и проектных работ с использованием программного обеспечения, изученного на данном курсе, участие в конкурсах образовательной организации, а также и на муниципальном уровне. В состав курса входят практические работы, позволяющие оценить степень усвоения материала.

III. Учебный план

№ п/п	Наименование разделов	Всего, часов	В том числе	
			лекции	практические занятия
1.	Изучение основ языка Python 3	42	13	29
2.	Объектно-ориентированный подход в программировании	18	4	14
3.	Создание графических 2D приложений с помощью Pygame на языке Python	21	6	15
4.	Изучение подходов к разработке своих модулей	6	2	4
5.	Индивидуальное проектирование: " Создание 2D приложения средствами Python "	21	5	16
ИТОГО:		108	30	78

IV. Организационно-педагогические условия реализации программы:

Формы организации образовательной деятельности: групповая (8-14 человек) совместная образовательная деятельность.

Форма проведения аудиторных занятий – теоретическая и практическая. Количество часов с использованием компьютерной техники соответствуют требованиям СанПиН и возрасту обучающихся.

Продолжительность одного занятия: 3 академических часа.

Объем нагрузки в неделю: 1 раз в неделю.

Принципы реализации программы:

- Доступности - процесс усвоения знаний, умений и навыков при реализации данной программы, учитывает возрастные особенности детей. Материалы располагаются от простых к более сложным или многократно повторяются.
- Сознательности и активности – для активизации учащихся программой предусмотрена форма групповой деятельности. Здесь важны мнения, суждения, оценки других. Используемая форма может проходить в виде конкурсов, викторин, открытых занятий, соревнований, выставок и т.д.
- Деятельностного обучения – учебные планы взаимодополняемы, имеют практическую направленность. Каждая тема подкрепляется проведением практических занятий.
- Преемственности - содержание более сложного материала основывается на знаниях, умениях и навыках, полученных на более ранних этапах обучения.

Формы и методы работы:

Теоретический материал на занятиях излагается в виде лекций, бесед, комментирования демонстрации работы с программным обеспечением.

Практический материал на занятиях преподается в виде практических творческих работ.

Для успешной реализации образовательного процесса используются следующие методы обучения:

- объяснительно-иллюстративный: беседы, рассказы с иллюстрациями или демонстрацией работы программного обеспечения.
- репродуктивный: учащиеся делают работы по образцу.
- диалогический: диалог между педагогом и учащимся, совместное обсуждение вариантов выполнения работ.
- частично-поисковый: учащиеся совместно с педагогом проводят поиск новых решений, поиск новых объектов изучения по темам занятий.

Формой подведения итогов реализации данной программы являются: самостоятельное выполнение творческих работ с использованием программного обеспечения, изученного на данном курсе, выполнение проектной работы, разработанной специально для данного курса, участие в

конкурсах образовательной организации, а также и на муниципальном уровне. В состав курса входят практические работы, позволяющие оценить степень усвоения материала.

Материально-технические условия реализации программы: программа оснащена современными техническими средствами, каждый обучающийся обеспечен отдельным рабочим местом, персональным компьютером с необходимым набором программ. Кабинет обеспечен видеосистемой для демонстрации работы с программным обеспечением, доской. Специально для курса разработаны раздаточные материалы.

Учебно-методическое и информационное обеспечение: программа обеспечена учебно-методическими пособиями и электронными учебно-методическими материалами для обучающихся, педагоги имеют возможность посещения методических семинаров по программе в течение всего учебного года, а также курсов по повышению квалификации.

V. Календарный график учебного процесса

на первое полугодие:

Месяцы	1-й месяц				2-й месяц				3-й месяц				4-й месяц				Всего
Недели	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	16
Теория	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	15
Практика	2	2	2	2	2	2	2	2	2	2	2	2	2	3	2	2	33
Итого:																48	

на второе полугодие:

Месяцы	1-й месяц				2-й месяц				3-й месяц				4-й месяц				5-й месяц				Всего
Недели	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	20
Теория	1	1	0	0	1	1	1	1	1	1	0	1	1	2	1	0	1	1	0	0	15
Практика	2	2	3	3	2	2	2	2	2	2	3	2	2	1	2	3	2	2	3	3	45
Итого:																				60	

VI. Учебно-тематический план

Код темы	Наименование разделов и перечень тематики занятий	В том числе		
		всего, часов	лекции	практические занятия
1	Изучение основ языка Python 3	42	13	29
1.1	Технология ООП	3	1	2
1.2.	Eclipse PyDef- интегрированная среда разработки для Python. Python в вопросах и ответах.	2	2	4
1.3	Введение в программирование на языке Python	3	1	2
1.4.	Основные алгоритмические конструкции	9	3	6
1.5	Встроенные типы данных	15	5	10
1.6	Стиль программирования	3	1	2
1.7	Промежуточная аттестация	3	0	3
2	Объектно-ориентированный подход в программировании	18	4	14
2.1	Базовые возможности классов	3	1	2
2.2	Инкапсуляция и взаимодействие классов	3	1	2
2.3	Наследование и полиморфизм	3	1	2

2.4.1	Перегрузка операторов и встроенных методов. Вложенные объекты и деструкторы	6	1	5
2.5	Промежуточная аттестация	3	0	3
3	Создание графических 2D приложений с помощью Pygame на языке Python	21	6	15
3.1	Спрайты и графические примитивы	3	1	2
3.2	Движение и поворот спрайтов, пользовательское управление	3	1	2
3.3	Базовая физика в Pygame: гравитация и столкновения	3	1	2
3.4	Создание анимации и камеры	3	1	2
3.5	Динамические объекты, базовый искусственный интеллект	3	1	2
3.6	Звук и эффекты	3	1	2
3.7	Промежуточная аттестация	3	0	3
4	Изучение подходов к разработке своих модулей	6	2	4
4.1	Пользовательские интерфейсы и меню	3	1	2
4.2	Сохранение данных и создание исполняемого файла	3	1	2
5	Индивидуальное проектирование: «Создание 2D приложения средствами Python»	21	5	16
5.1	Разработка интерфейса разрабатываемого модуля, системы именования объектов в программе и вариантов реализации основных алгоритмов на бумаге. Создание основных элементов интерфейса	3	2	1
5.2	Реализация основных алгоритмов	6	1	5
5.3	Отладка модуля	9	2	7
5.4	Публичный показ и защита индивидуального приложения	3	0	3

VII. Содержание программы по темам

Раздел 1. Изучение основ языка Python 3.

Введение в объектно-ориентированное программирование (ООП): понятия объекты, свойства объекта, функции, события, обработчик события. Базовые концепции ООП: инкапсуляция, наследование, полиморфизм.

Eclipse PyDef -интегрированная среда разработки для Python: Настройка Eclipse, доступ к расширенным справочным данным средствами Python.doc, Меню Window/Preferences, раздел General. Editors. Text Editors/Spelling, установка VE 1.4 для Eclipse 3.5. Galileo, Создание нового модуля на Python.

Последовательность операторов, встроенные типы данных, выражении, последовательности, имена.

В результате обучения Слушатель должен

знать:

- интерфейс JDK Eclipse PyDef и правила его настройки
- основы языка Python, ориентироваться в областях его применения
- работу основных алгоритмических конструкций

уметь: -

- ориентироваться в углубленных технологиях применения языка Python в современном программировании, таких, как: работа с базами данных, создание графических пользовательских интерфейсов и т.п..

должен владеть:

- - практическими навыками создания модуля в Python;
- - практическими навыками создания модулей в Python с применением продвинутых технологий

Раздел 2 Объектно-ориентированный подход в программировании

Написание классов с атрибутами, методами и конструкторами. Создание экземпляров класса. Взаимодействие классов, инкапсуляция методов и атрибутов. Наследование классов, перегрузка функций родительского класса. Перегрузка операторов и встроенных функций. Проектирование вложенных структур классов, деструкторы.

В результате обучения Слушатель должен

знать:

- принципы работы и свойства элементов управления классов.
- способы оптимизировать структуру проекта наследованием и полиморфизмом
- принципы взаимодействия классов и приватизации атрибутов

уметь:

- реализовывать объектно-классовую структуру проекта
- оптимизировать объем кода, заменяя схожую логику наследованными классами и полиморфными функциями.
- реализовывать вложенную структуру классов, контролируя удаление объектов.

Раздел 3. Создание графических 2D приложений

Создание ООП – заготовки для графических приложений. Создание и отображение средствами библиотеки Pygame окна. Управление движением спрайта: моделирование скорости и пользовательское управление с клавиатуры. Реализация поворота спрайта с помощью мыши или клавиатуры.

В результате обучения Слушатель должен

знать:

- Основные встроенные функции и структуры данных модуля Pygame
- Техники программирования графики и анимации
- Техники моделирования физики и имитирования поведения искусственного интеллекта

уметь:

- Проектировать структуру классов графического приложения
- Оптимизировать расчеты игрового цикла и рендеринг
- Обрабатывать игровые события

Раздел 4. Изучение подходов к разработке своих модулей

Создание элементов пользовательского интерфейса, техника создания игрового инвентаря. Разделение приложения на модули, импорт собственных модулей. Сохранение сложных структур данных в файл. Написание собственного меню и организация работы нескольких уровней.

В результате обучения Слушатель должен

знать:

- Технику разработки пользовательских интерфейсов
- Способы организации работы множества уровней

- Алгоритм создания исполняемого exe файла

уметь:

- Создавать игровое меню
- Сохранять результат работы приложения в файл
- Импортировать собственные модули

Раздел 5. Индивидуальное проектирование: «Создание 2D приложения средствами Python»

Индивидуальное проектирование проводится в виде выполнения курсового проекта-модуля. Каждый Слушатель выполняет модуль по индивидуальному заданию. Модуль имеет обязательную и творческую часть. Модуль призван интегрировать все знания и умения, полученные слушателем в процессе обучения. В результате Слушатель разработает программный продукт, самостоятельно пройдя все этапы, начиная с разработки интерфейса, заканчивая тестированием и отладкой. Курсовой проект предполагает публичный показ Слушателем своей работы и защиту проекта. Итоговая работа имеет следующие этапы:

- Разработка интерфейса разрабатываемой программы, системы именования объектов в программе и вариантов реализации основных алгоритмов на бумаге.
- Создание основных элементов интерфейса.
- Реализация основных алгоритмов.
- Отладка проекта.
- Публичный показ и защита индивидуального проекта.

VIII. Оценочные материалы

Оценка уровня подготовки слушателей осуществляется в форме текущего (фронтальная и индивидуальная проверка, выполнение практических и самостоятельных работ, устный опрос, тестирование, срезы) и итогового контролей знаний и умений. Результаты регулярно отражаются в Журнале учёта успеваемости.

Упражнения по теме «Изучение основ языка Python 3»

Практическая работа

Герой с начальной силой 10 входит в башню с несколькими этажами. На каждом этаже находится некоторое количество комнат, расположенных по кругу.

В каждой комнате героя ожидает один из трех вариантов:

- 1) - монстр с силой k_1 ,
- 2) - зелье с силой k_2
- 3) - артефакт с силой k_3

Параметры k_1 k_2 k_3 могут быть не одинаковы для разных комнат одного типа.

Если герой наткнется на монстра, сила которого k_1 больше силы героя, то это поражение. В противном случае герой проходит комнату. Если герой находит артефакт, его сила повышается на k_3 . Если герой находит зелье, его сила также повышается на k_2 , но только на одну следующую комнату, и не важно, что в ней находится.

Герой может перемещаться по этажу только в одном из двух направлений (по или против часовой стрелки), проверяя все комнаты подряд. Прохождение этажа может начаться с любой комнаты.

Перейти к следующему этажу можно лишь полностью пройдя все комнаты на текущем уровне.

Успешно завершив этаж, герой переходит к следующему и сохраняет все свои артефакты.

Помогите герою пройти всю башню целиком, указав, с какой двери ему начать и в каком направлении двигаться. Предоставить такую информацию для всех этажей.

Если этаж пройти невозможно (при любых стартовых условиях герой попадает в комнату с более сильным монстром), нет смысла проходить оставшиеся этажи: программа завершает работу. Если этаж можно пройти несколькими способами, программа указывает любой из них в решении.

Входные данные записываются в файл KR.in (создается вручную) следующим образом: пишется номер этажа, и после него

описание каждой комнаты в виде текста: в строке обязательно содержится одно число и одно из слов: "артефакт", "зелье", "монстр".

Далее, расположены записи схожего формата. Количество этажей и комнат заранее неизвестно.

Этажи идут строго в порядке номеров (номера нужны лишь для удобства чтения файла).

Программа записывает ответ в файл KR.out (создается вручную). В ответе содержатся строчки, как пройти каждый этаж: номер стартовой двери и направление. Если какой-то этаж невозможно пройти, запись в файл обрывается на строчке "Этаж № j пройти невозможно" Функция проверки обхода этажа с заданными параметрами: стартовой комнатой и направлением

Упражнения по теме «Объектно-ориентированный подход в программировании»

Практическая работа

Напишите программу для игры в Блэкджек: используйте старые классы для карт, руки, игрока и колоды.

Правила игры:

Участвуют несколько игроков (имена указываются пользователем) и раздающий.

Из перемешанной колоды каждый игрок получает по две открытые карты, и две получает дилер, но первая из них перевернута рубашкой вверх.

Каждая карта имеет свою ценность: цифры от 2 до 10 имеют ценность соответственно от 2 до 10, Валет, Дама и Король имеют ценность 10, а туз имеет ценность 1, если очков вместе с ним больше 11, иначе, его ценность 11 (например, 2+туз = 13, и дама+2+туз = 13).

Далее каждого игрока спрашивают, желает ли он добрать карт: если игрок набирает больше 21 очка, он проигрывает, иначе остается в игре.

После того, как все добрали карт, раздающий открывает свою вторую карту. Пока у него меньше 17 очков, он добирает карты.

Если раздающий перебрал, все оставшиеся игроки побеждают. Если раздающий и один или несколько игроков еще в игре, сравнивается количество очков: все, у кого очков больше, чем у раздающего побеждают, меньше - проигрывают, равно - ничья.

Если все игроки перебрали, то раздающий сразу выигрывает, без добора.

Добавьте классу карты свойство - количество очков за нее (туз как 1). Добавьте классу карты атрибут, скрывающий карту (благодаря ему, закрытая карта печатается как XX).

Классу руки добавьте метод, возвращающий сумму очков имеющихся карт. Класс игрок, наследуемый от руки, получает методы для контролирования добора карт. Класс раздающий, наследуемый от руки, может переворачивать свою 1 карту и может добирать, если у него меньше 17 очков.

В главной функции смоделируйте одну игровую сессию: пользователь вводит имена и отвечает за каждого игрока: берет он карты или нет. Далее, все как по правилам игры: после добора игроков, раздающий показывает свои карты, и определяется победитель.

Упражнения по теме «Создание графических 2D приложений»

Практическая работа

Создайте уровень, на котором расположены: трава, разные виды деревьев и домов.

Уровень должен генерироваться в соответствии с массивом строк, заполненным в программе. массив имеет 10 строк по 10 символов в каждой и может выглядеть так:

```
"-----",  
"- + -",  
"---- -",  
"-++ -",  
"- - -",  
"- --"
```

```
"-- ++ -",  
"- ----",  
"- -- +",  
"-----"
```

Здесь минус означает дерево, плюс - дом, пробел - траву (можете выбрать другие обозначения)
Размер экрана - 640 на 640. Каждый символ обозначает блок соответствующего спрайта 64 на 64. В случае травы и дома, файл спрайта выбирается случайно из трех предоставленных.

Воспользуйтесь предыдущей программой для генерации уровня указанным способом.

Результат работы модуля:

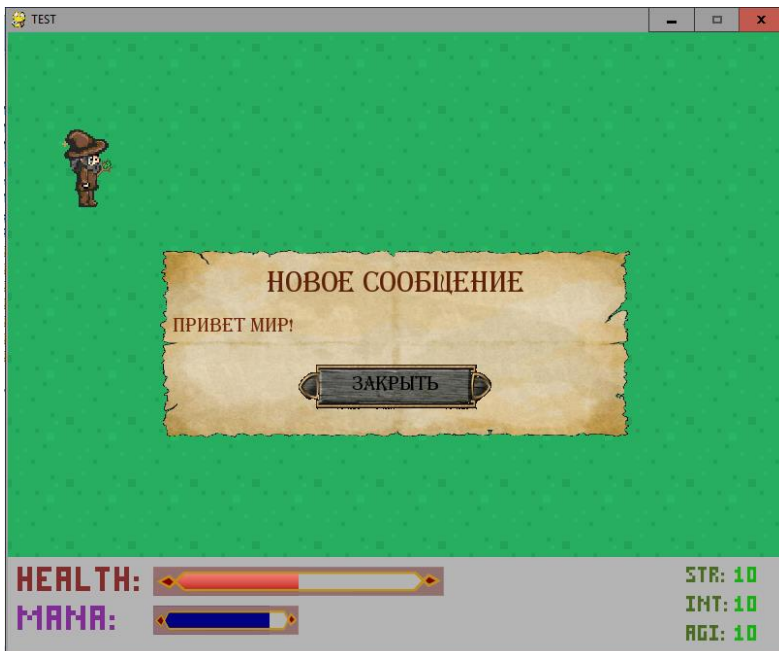


Упражнения по теме «Изучение подходов к разработке своих модулей»

Практическая работа

Напишите класс кнопки, которая обладает текстом и тремя спрайтами (обычное состояние, нажатие, наведение). Напишите еще один класс для отображения игровых сообщений с заголовком, текстом и кнопкой. Сообщение приостанавливает игровой процесс и лишь обрабатывает нажатие своей кнопки (при нажатии сообщение закрывается и игра продолжается). Для тестирования, организуйте вывод сообщения по кнопке пробел

Результат



Упражнения по теме «Индивидуальное проектирование: «Создание 2D приложения средствами Python»»

Индивидуальный проект по курсу «Создание 2D приложения средствами Python» объем учебной программы – 21 час

Постановка задачи:

1. Согласно индивидуальному заданию создать 2D приложение на языке Python, используя библиотеку pygame.
2. Составить пояснительную записку к проекту.

Общие требования к выполнению работы:

1. Проект должен быть сохранен в папку **Z:\Python_itog\смысловое имя_Фамилия_Группа**.
 - Папка проекта должна содержать:
 - Ресурсы приложения (спрайты, звуки, шрифты), распределенные во вложенных папках по смыслу (например, ресурсы для игрока, уровня и врагов в папках с соответствующими названиями)
 - Модули, содержащие классы – компоненты приложения (также распределенные по смыслу, например, модуль animation содержит класс для создания анимации с любыми спрайтами, а модуль player.py содержит классы для персонажа, камеры и полоски здоровья)
 - Модуль main.py, который организует запуск приложения (при его компиляции можно увидеть результат работы программы)
2. Все модули должны компилироваться без ошибок, а при компиляции модуля main.py должен быть виден результат работы приложения.
3. Демо-версия может содержать расширенный функционал приложения. Допускается пренебрежение некоторыми особенностями приложения (подробнее см. задание). Творческие дополнения на вкус разработчика приветствуются.
4. Все имена переменных и элементов управления должны быть осмысленными.

5. Код вашей программы должен быть комментирован. На каждые 5-6 строчек кода, описание переменных или законченный смысловой фрагмент должен стоять комментарий.
6. Код вашей программы должен быть структурированным.
7. Приложение может использовать предоставленные в задании ресурсы, или же сторонние, на усмотрение разработчика, если они не противоречат общей концепции проекта.
8. Приложение должно быть хорошо оптимизировано (отсутствие зависаний).
9. Приложение должно содержать стартовое меню. Также необходимо предусмотреть возможность выхода из программы в любой момент времени.
10. Управление в приложении должно быть интуитивно-понятным. Точную инструкцию по управлению необходимо предоставить внутри самой программы (пункт меню) или в текстовом файле readme.

Порядок проведения защиты

На защиту необходимо представить пояснительную записку и папку с исходниками и ресурсами проекта.

При наличии ошибок компиляции главного файла комиссия вправе не рассматривать далее работу слушателя и поставить ему «неудовлетворительно».

Оценка складывается из двух оценок, полученных за каждый описанный ниже этап защиты:

I этап. Слушатель компилирует главный файл проекта и показывает приемной комиссии реализованную функциональность своего приложения. За этот этап слушатель получает оценку, согласно заданию.

II этап. К этому этапу допускаются только те слушатели, которые успешно сдали предыдущий этап защиты. Слушатель должен продемонстрировать приемной комиссии реализованную часть проекта, сопровождая демонстрацию своими комментариями, рассказать об основных алгоритмах, как они реализованы, как выполнялась разработка, показать свой проект с лучшей стороны. Приемная комиссия задает возникшие по ходу демонстрации вопросы.

Факторы, повышающие оценку за индивидуальный проект:

1. Наличие усовершенствований приложения, не указанных в задании
2. Дополненный дизайн приложения, с использованием собственных изображений и анимаций (не прилагавшихся к заданию)
3. Наличие дополнительных способов оптимизации приложения

Факторы, понижающие оценку за индивидуальный проект:

1. Приложение плохо оптимизировано (заметные торможения).
2. Не выполнены «Общие требования к выполнению работы» (см. лист 1 задания)
3. Скучное содержимое уровня
4. Сильное отклонение от главной идеи приложения

IX. Методические материалы и рекомендации

Программа «Программирование на Python» знакомит слушателя с объектно-ориентированным подходом в программировании, что является залогом к успешному дальнейшему переходу на более сложные языки программирования. Рассматривая различные элементы управления и встроенные классы, преподавателю следует выявлять, обращать внимание слушателей на приметы работы базисных основ объектно-ориентированного подходов в программировании.

Программа «Программирование на Python» нацелена на выработку навыка разработки алгоритмов для решения задач различного назначения, как вычислительных, так и прикладных. Курс помимо теоретического базиса, носящего фундаментальный характер, насыщен тренинговым контентом в виде большого количества прикладных упражнений. Идея алгоритмизация должна проходить через все этапы программы обучения, поэтому преподавателю нужно, разбирая со слушателями прикладные задачи, упор делать именно на построение алгоритма решения.

Преподавателю, ведущему курс, рекомендуется обращать особое внимание на выработку у слушателей умения составлять корректные, быстродействующие алгоритмы, экономящие ресурсы компьютера.

Построение программы и наполнение методических материалов учитывает наличие как мотивированных и опытных в программировании слушателей, так и начинающих.

IX. Литература

- *Коэльё Л. П., Ричерт В.* Построение систем машинного обучения на языке Python.- Перевод с английского. — М.: ДМК Пресс, 2015.
- *Маккинли У.* Python и анализ данных.- Перевод с английского.- М.: ДМК Пресс, 2015.— 482с .
- *Марк Саммерфилд.* Python на практике.- Перевод с английского. – М. ДМК Пресс, 2014.- 338 с.
- *Марк Лутц.* Программирование на Python / Пер. с англ. - 4-е изд. – СПб, 2011
- *Марк Лутц.* [Программирование на Python](#) / Пер. с англ. - 4-е изд.- СПб, 2011. Т.І. 992с.-
- *Марк Лутц.* [Изучаем Python, 4-е издание](#). — Перевод с английского. — СПб, 2010 1280