

**Образовательное частное учреждение
дополнительного профессионального образования
«Центр компьютерного обучения «Специалист.Ру»
Учебно-научного центра при МГТУ им. Н.Э. Баумана»
(ОЧУ «Специалист.Ру»)**

123317, город Москва, Пресненская набережная, д. 8, строение 1, этаж 48, помещение
484с, комната 4,
ИНН 7701345493, ОГРН 1037701927031

Утверждаю:

Директор ОЧУ «Специалист.Ру»



О.В.Пичугина/

2018__ года

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
(ОБЩЕРАЗВИВАЮЩАЯ) ПРОГРАММА
«Основы программирования на языке "Java" для
школьников. Модуль 2»**

для детей и взрослых

Срок обучения: 4 недели

Форма обучения: очная

город Москва

Аннотация

1. Пояснительная записка

Дополнительное образование – это процесс свободно избранного ребенком освоения знаний, способов деятельности, ценностных ориентаций, направленных на удовлетворение интересов личности, ее склонностей, способностей и содействующей самореализации, и культурной адаптации, выходящих за рамки стандарта общего образования. Дополнительное образование детей направлено на формирование и развитие творческих способностей детей, удовлетворение их индивидуальных потребностей в интеллектуальном, нравственном и физическом совершенствовании, формирование культуры здорового и безопасного образа жизни, а также на организацию их свободного времени. Дополнительное образование детей обеспечивает их адаптацию к жизни в обществе, профессиональную ориентацию, а также выявление и поддержку детей, проявивших выдающиеся способности.

Дополнительное образование обладает большими возможностями для совершенствования общего образования, его гуманизации; позволяет полнее использовать потенциал школьного образования за счет углубления, расширения и применения школьных знаний; позволяет расширить общее образование путем реализации досуговых и индивидуальных образовательных программ, дает возможность каждому ребенку удовлетворить свои индивидуальные познавательные, эстетические, творческие запросы.

Дополнительное образование детей — необходимое звено в воспитании многогранной личности, в ее образовании, в ранней профессиональной ориентации. Ценность дополнительного образования детей состоит в том, что оно усиливает вариативную составляющую общего образования и помогает ребятам в профессиональном самоопределении, способствует реализации их сил, знаний, полученных в базовом компоненте. Дополнительное образование детей создает юному человеку условия, чтобы полноценно прожить пору детства. Ведь если ребенок полноценно живет, реализуя себя, решая задачи социально значимые, выходит даже в профессиональное поле деятельности, то у него будет гораздо больше возможностей достичь в зрелом возрасте больших результатов, сделать безошибочный выбор. Занятость учащихся во внеучебное время содействует укреплению самодисциплины, развитию самоорганизованности и самоконтроля школьников, появлению навыков содержательного проведения досуга, позволяет формировать у детей практические навыки здорового образа жизни, умение противостоять негативному воздействию окружающей среды.

Программа дополнительного образования «ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ JAVA ДЛЯ ШКОЛЬНИКОВ. Модуль 2» имеет научно-техническую направленность.

Возраст обучающихся: 12-16 лет.

После успешного освоения курса и прохождения итоговой аттестации обучающиеся получают свидетельство об обучении, установленного образца ОЧУ «Специалист». Программа реализуется в объеме 24 академических часов, срок обучения: 4 недели.

СОДЕРЖАНИЕ

1	Общая характеристика программы
2	Цель и задачи программы
3	Учебный план
4	Содержание учебно-тематического плана
5	Планируемые результаты обучения
6	Календарный учебный график
7	Организационно-педагогические условия реализации программы
8	Формы аттестации/контроля, оценочные материалы и критерии оценивания результатов обучения
9	Список литературы
Приложение 1. Методические материалы к модулям 1-8	
Приложение 2. Методические материалы «Портфолио»	

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

Курс дополнительной общеобразовательной общеразвивающей программы для детей и взрослых «Основы программирования на языке "Java" для школьников, Модуль 2» относится к программам дополнительных общеобразовательных курсов для детей и взрослых и представляет собой комплекс основных характеристик образования (объем, содержание, планируемые результаты), организационно-педагогических условий, форм аттестаций, разработанный и утвержденный ОЧУ «Специалист.РУ».

Программа разработана на основе следующих **нормативных документов**:

- Федеральный закон от 29 декабря 2012 г. №273-ФЗ «Об образовании в Российской Федерации»;
- Приказ Министерства образования и науки Российской Федерации № 1008 от 29 августа 2013 г «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Постановление Главного государственного санитарного врача Российской Федерации от 4 июля 2014 г. N 41 г. Москва «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей».
- Устав ОЧУ «Специалист.РУ», локальные нормативные акты.

- **Направленность** (профиль) программы: научно-техническая;

Срок обучения: 4 недели; программа реализуется в объеме 24 академических часов. Программа курса содержит учебный план, календарный график, рабочую программу, методические материалы, материалы итоговой аттестации.

Актуальность программы, соответствие государственному, социальному заказу/запросам.

В результате прохождения обучения слушатель сформирует знаний в области программирования на языке Java, разовьет логическое мышление, активизирует навыки использования полученных знаний и умений в практической деятельности, проявит интерес к науке, фантазии, изобретательности, создание оптимальных условий для всестороннего удовлетворения потребностей обучающихся и развития их индивидуальных склонностей и способностей, появится мотивация личности к познанию и творчеству.

Курс предназначен для всех желающих в возрасте 12-16 лет. Курс будет полезен для тех, кто интересуется сферами в области научно-технической направленности; будущих программистов, аналитиков, инженеров.

Отличительные особенности программы В процессе освоения программы, обучающиеся смогут в раннем возрасте получить профессиональную ориентацию.

Адресат программы и примерный портрет слушателя курсов: программа разработана для детей и взрослых (от 12 лет), которых интересует научно-техническая направленность. Специальные требования к входным знаниям, умениям и компетенциям обучающихся - не предъявляются.

2. ЦЕЛЬ И ЗАДАЧИ ПРОГРАММЫ

Цель и задачи курса:

В результате освоения курса, обучающиеся научатся формировать знания в области программирования на языке Java, разовьют логическое мышление, активизируют навыки использования полученных знаний и умения в практической деятельности, разовьют интерес к науке, фантазии, изобретательности, создадут оптимальные условия для всестороннего удовлетворения потребностей обучающихся и развития их индивидуальных склонностей и способностей, мотивации личности к познанию и творчеству.

Задачи:

- формирование системы знаний и умений в области программирования на языке Java;
- изучение интересов и потребностей учащихся в дополнительном образовании;
- расширение различных видов деятельности в системе дополнительного образования детей для наиболее полного удовлетворения интересов и потребностей, учащихся в объединениях по интересам;
- создание условий для привлечения к занятиям в системе дополнительного образования детей большего числа учащихся среднего и старшего возраста;
- определение содержания дополнительного образования детей, его форм и методов работы с учащимися с учетом их возраста и интересов;
- развитие творческого потенциала личности и формирование нового социального опыта.

3. УЧЕБНЫЙ ПЛАН

В процессе преподавания курса «Основы программирования на языке "Java" для школьников. Модуль 2» используются как классические методы обучения (лекции), так и различные виды практической работы обучающихся по заданию преподавателя, которые направлены на развитие в области программирования.

Трудоемкость программы: 24 академических часа (20 аудиторных).

Срок обучения: 4 недели

Режим занятий: 8 академических часов в неделю (2-я половина дня).

Занятия начинаются не ранее 8.00 часов утра и заканчиваются не позднее 20.00 часов. Для обучающихся в возрасте 14-18 лет допускается окончание занятий в 21.00 часов. При наличии двух смен занятий организуется не менее 30-минутный перерыв между сменами для уборки и проветривания помещений.

Продолжительность занятий детей в учебные дни - не более 3-х академических часов в день, в выходные и каникулярные дни - не более 4 академических часов в день. После 30-45 минут теоретических занятий организуется перерыв длительностью не менее 10 мин¹.

Формы организации образовательного процесса (индивидуальные, групповые и т.д.) и другие виды занятий по программе определяются содержанием программы. По программе могут предусматриваться лекции, практические и семинарские занятия, лабораторные работы, круглые столы, мастер-классы, мастерские, деловые и ролевые игры, тренинги, выездные тематические занятия, творческие отчеты и другие виды учебных занятий и учебных работ.

3.1. Учебный план

№ п/п	Наименование модулей по программе	Общая трудоемкость (акад. часов)	Аудиторные часы			СРС, ч
			Всего	В том числе:		
				Лекций	Практических занятий	
1	Модуль 1. Обработка событий, связанных с устройством управления "мышь"	1,5	1,5	1	0,5	0
2	Модуль 2. Проектирование интерфейса пользователя	2	1,5	1	0,5	0,5
3	Модуль 3. Создание приложения "Калькулятор". Преобразование приложения в апплет	2	1,5	1	0,5	0,5
4	Модуль 4. Верстка WEB-страницы с помощью HTML и CSS	2	1,5	1	0,5	0,5
5	Модуль 5. Использование JavaScript на WEB-странице	2,5	2	1,5	0,5	0,5
6	Модуль 6. Создание поздравительного блокировщика WINDOWS. Реализация интерфейса	2,5	2	1,5	0,5	0,5
7	Модуль 7. Создание поздравительного блокировщика WINDOWS. Завершение	2	1,5	1	0,5	0,5
8	Модуль 8. Визуальные средства разработки. Создание интерфейса приложения	2	1,5	1	0,5	0,5

¹ Согласно П. VIII. Требования к организации образовательного процесса. Постановления Главного государственного санитарного врача Российской Федерации от 4 июля 2014 г. N 41 г. Москва «Об утверждении СанПиН 2.4.4.3172-14 "Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей».

9	Модуль 9. Работа с файловой системой из Java. Создание, удаление, переименование файлов. Чтение и запись в файл	3,5	3	2	1	0,5
	Модуль 10. Базы данных. Язык запросов SQL. Работа с базой данных MySQL из JAVA	2	2	1,5	0,5	-
	Модуль 11. Создание игры "Предсказание будущего" с использованием базы данных	2	2	1,5	0,5	-
	Итог:	24	20	14	6	4
	Итоговая аттестация					

Академический час, предусмотренный для всех видов занятий – 45 мин.

3.2. Рабочая программа курса/модулей

Модуль 1. Обработка событий, связанных с устройством управления "мышь"

Виды событий для устройства "мышь".

Принципы обработки событий мыши из Java.

Обработка нажатий кнопок мыши и перемещения курсора мыши.

Создание приложения для рисования мышью.

Выбор цвета для рисования мышью.

Рисование мышью.

Модуль 2. Проектирование интерфейса пользователя

Общие принципы добавления элементов интерфейса пользователя.

Обработка событий.

Знакомство с основными элементами интерфейса: кнопка, надпись, текстовое поле.

Проектирование интерфейса калькулятора.

Модуль 3. Создание приложения "Калькулятор". Преобразование приложения в апплет

Программирование нажатий кнопок калькулятора.

Знакомство с апплетами.

Преобразование приложения калькулятора в апплет.

Создание JAR-архива апплета.

Подключение апплета к HTML-странице.

Запуск апплета в браузере.

Модуль 4. Верстка WEB-страницы с помощью HTML и CSS

Схема работы веб-приложений.

Знакомство с HTML, CSS.

Форма записи HTML-тегов.

Понятие верстки веб-страницы, типы верстки.

Изучение табличной верстки.

Создание формы для ввода логина и пароля пользователем.

Настройка внешнего вида формы с помощью CSS.

Модуль 5. Использование JavaScript на WEB-странице

Знакомство с JavaScript. Предназначение JavaScript.
Некоторые отличия JavaScript от Java.
Доступ к элементам веб-страницы из JavaScript.
Вывод диалоговых окон из JavaScript, переход по адресу.
Проверка логина, пароля пользователя с помощью JavaScript.

Модуль 6. Создание поздравительного блокировщика WINDOWS. Реализация интерфейса

Общий подход к созданию приложений Windows, блокирующих экран.
Создание окна с размерами текущего разрешения экрана.
Удаление рамки окна.
Добавление поздравительных текстовых надписей.
Создание текстового поля для ввода пароля.
Создание кнопки закрытия блокировщика.

Модуль 7. Создание поздравительного блокировщика WINDOWS. Завершение Знакомство с реестром Windows.

Использование WinAPI в Java приложении.
Программирование таймера для поднятия окна блокировщика поверх других окон. Применение WinAPI для обращения к окну блокировщика.

Модуль 8. Визуальные средства разработки. Создание интерфейса приложения Изучение визуального конструктора Eclipse для построения интерфейса пользователя. Добавление элементов интерфейса с помощью мыши.

Синхронизация программного кода и визуального конструктора.
Изменение свойств и подключение обработчиков событий в визуальном конструкторе.

Модуль 9. Работа с файловой системой из Java. Создание, удаление, переименование файлов. Чтение и запись в файл

Основы работы с файлами и файловой системой.
Создание интерфейса приложения для операций с файлами.
Создание нового файла из Java.
Удаление существующего файла.
Изменение имени файла.
Принципы хранения данных файлах.
Потоки ввода-вывода Java для работы с файлами.
Запись данных многострочного текстового поля в файл.
Считывание содержимого файла и вывод в многострочное текстовое поле.

Модуль 10. Базы данных. Язык запросов SQL. Работа с базой данных MySQL из JAVA

Основы работы с базами данных.
Реляционная база данных.
Понятие таблицы, поля, записи, первичного ключа.
Язык структурированных запросов SQL: добавление, удаление, изменение, чтение строк таблицы.
Знакомство с СУБД MySQL и интерфейсом PHPMyAdmin.
Взаимодействие с базой данных MySQL из Java.

Модуль 11. Создание игры "Предсказание будущего" с использованием базы данных

Создание базы данных и таблицы для хранения данных игры.
Добавление данных в таблицу.

Построение интерфейса пользователя игры.
Получение случайной строки таблицы базы данных из приложения Java.

3.3. Практические занятия

<i>№ n/n</i>	<i>№ и название темы дисциплины</i>	<i>Тематика семинарских (практических) занятий</i>	<i>СРС</i>
1	Модуль 1 Обработка событий, связанных с устройством управления "мышь"	Принципы обработки событий мыши из Java. Обработка нажатий кнопок мыши и перемещения курсора мыши. Создание приложения для рисования мышью. Выбор цвета для рисования мышью	Рисование мышью
2	Модуль 2. Проектирование интерфейса пользователя	Обработка событий. Знакомство с основными элементами интерфейса: кнопка, надпись, текстовое поле. Проектирование интерфейса калькулятора.	основными элементами интерфейса: кнопка, надпись, текстовое поле
3	Модуль 3 Создание приложения "Калькулятор". Преобразование приложения в апплет	Программирование нажатий кнопок калькулятора. Знакомство с апплетами. Преобразование приложения калькулятора в апплет. Создание JAR-архива апплета.	Подключение апплета к HTML-странице.
4	Модуль 4. Верстка WEB-страницы с помощью HTML и CSS	Знакомство с HTML, CSS. Форма записи HTML-тегов. Понятие верстки веб-страницы, типы верстки. Изучение табличной верстки. Создание формы для ввода логина и пароля пользователем	Настройка внешнего вида формы с помощью CSS.
5	Модуль 5. Использование JavaScript на WEB-странице	Знакомство с JavaScript. Предназначение JavaScript. Некоторые отличия JavaScript от Java. Доступ к элементам веб-страницы из JavaScript. Вывод диалоговых окон из JavaScript, переход по адресу	Проверка логина, пароля пользователя с помощью JavaScript.
6	Модуль 6. Создание поздравительного блокировщика WINDOWS. Реализация интерфейса	Создание окна с размерами текущего разрешения экрана. Удаление рамки окна. Добавление поздравительных текстовых надписей. Создание текстового поля для ввода пароля.	Создание кнопки закрытия блокировщика.
7	Модуль 7. Создание поздравительного блокировщика WINDOWS. Завершение	Использование WinAPI в Java приложении. Программирование таймера для поднятия окна блокировщика поверх других окон.	Применение WinAPI для обращения к окну блокировщика. графического объекта.

<i>№ n/n</i>	<i>№ и название темы дисциплины</i>	<i>Тематика семинарских (практических) занятий</i>	<i>СРС</i>
8	Модуль 8. Визуальные средства разработки. Создание интерфейса приложения	Изучение визуального конструктора Eclipse для построения интерфейса пользователя. Добавление элементов интерфейса с помощью мыши. Синхронизация программного кода и визуального конструктора.	Изменение свойств и подключение обработчиков событий в визуальном конструкторе.
9	Модуль 9. Работа с файловой системой из Java. Создание, удаление, переименование файлов. Чтение и запись в файл	Создание интерфейса приложения для операций с файлами. Создание нового файла из Java. Удаление существующего файла. Изменение имени файла. Принципы хранения данных файла. Потоки ввода-вывода Java для работы с файлами. Запись данных многострочного текстового поля в файл.	Считывание содержимого файла и вывод в многострочное текстовое поле.
10	Модуль 10. Базы данных. Язык запросов SQL. Работа с базой данных MySQL из JAVA	Понятие таблицы, поля, записи, первичного ключа. Язык структурированных запросов SQL: добавление, удаление, изменение, чтение строк таблицы. Знакомство с СУБД MySQL и интерфейсом PHPMyAdmin. Взаимодействие с базой данных MySQL из Java.	добавление, удаление, изменение, чтение строк таблицы.
11	Модуль 10. Создание игры "Предсказание будущего" с использованием базы данных	Добавление данных в таблицу. Построение интерфейса пользователя игры. Получение случайной строки таблицы базы данных из приложения Java.	Создание базы данных и таблицы для хранения данных игры.

4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

<i>№</i>	<i>Этапы в процессе освоения курса (темы/задачи)</i>	<i>Компоненты</i>		
		<i>знания</i>	<i>умения</i>	<i>навыки</i>
1	Модуль 1. Обработка событий, связанных с устройством управления "мышь"	<u>Знать:</u> Виды событий для устройства "мышь". Принципы обработки событий мыши из Java.	<u>Уметь:</u> Обработка нажатий кнопок мыши и перемещения курсора мыши. Создание приложения для рисования мышью.	<u>Владеть навыками:</u> Рисование мышью
2	Модуль 2. Проектирование интерфейса пользователя	<u>Знать:</u> Общие принципы добавления элементов	<u>Уметь:</u> Работать с основными элементами интерфейса:	<u>Владеть навыками:</u> Применять кнопка, надпись, текстовое поле

		интерфейса пользователя	кнопка, надпись, текстовое поле	
3	Модуль 3. Создание приложения "Калькулятор". Преобразование приложения в апплет	<u>Знать:</u> Программирование нажатий кнопок калькулятора	<u>Уметь:</u> Преобразование приложения калькулятора в апплет	<u>Владеть навыками:</u> Подключение апплета к HTML-странице
4	Модуль 4. Основы объектно-ориентированного программирования	<u>Знать:</u> Схема работы веб-приложений. Знакомство с HTML, CSS. Форма записи HTML-тегов.	<u>Уметь:</u> Создание формы для ввода логина и пароля пользователем.	<u>Владеть навыками:</u> Настройка внешнего вида формы с помощью CSS.
5	Модуль 5. Использование JavaScript на WEB-странице	<u>Знать:</u> Предназначение JavaScript. Некоторые отличия JavaScript от Java	<u>Уметь:</u> Вывод диалоговых окон из JavaScript, переход по адресу	<u>Владеть навыками:</u> Проверка логина, пароля пользователя с помощью JavaScript
6	Модуль 6. Создание поздравительного блокировщика WINDOWS. Реализация интерфейса	<u>Знать:</u> Создание окна с размерами текущего разрешения экрана.	<u>Уметь:</u> Удаление рамки окна. Добавление поздравительных текстовых надписей	<u>Владеть навыками:</u> Создание текстового поля для ввода пароля. Создание кнопки закрытия блокировщика.
7	Модуль 7. Создание поздравительного блокировщика WINDOWS. Завершение	<u>Знать:</u> Использование WinAPI в Java приложении	<u>Уметь:</u> Программирование таймера для поднятия окна блокировщика поверх других окон	<u>Владеть навыками:</u> Применение WinAPI для обращения к окну блокировщика
8	Модуль 8. Визуальные средства разработки. Создание интерфейса приложения	<u>Знать:</u> Изучение визуального конструктора Eclipse для построения интерфейса пользователя	<u>Уметь:</u> Добавление элементов интерфейса с помощью мыши. Синхронизация программного кода и визуального конструктора.	<u>Владеть навыками:</u> Изменение свойств и подключение обработчиков событий в визуальном конструкторе.
9	Модуль 9. Работа с файловой системой из Java. Создание,	<u>Знать:</u> Основы работы с файлами и файловой	<u>Знать:</u> Создание нового файла из Java. Удаление	<u>Владеть навыками:</u> Считывание

	удаление, переименование файлов. Чтение и запись в файл	системой. Создание интерфейса приложения для операций с файлами.	существующего файла. Изменение имени файла.	содержимого файла и вывод в многострочное текстовое поле.
10	Модуль 10. Базы данных. Язык запросов SQL. Работа с базой данных MySQL из JAVA	<u>Знать:</u> Основы работы с базами данных. Реляционная база данных.	<u>Уметь:</u> Язык структурированных запросов SQL: добавление, удаление, изменение, чтение строк таблицы.	<u>Владеть навыками:</u> Взаимодействие с базой данных MySQL из Java
11	Модуль 11. Создание игры "Предсказание будущего" с использованием базы данных	<u>Знать:</u> Создание базы данных и таблицы для хранения данных игры. Добавление данных в таблицу.	<u>Уметь:</u> Построение интерфейса пользователя игры	<u>Владеть навыками:</u> Получение случайной строки таблицы базы данных из приложения Java.

5. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Календарный учебный график формируется при осуществлении обучения в течение всего календарного года. По мере набора групп слушателей по программе составляется календарный график, учитывающий объемы лекций, практики, самоподготовки, выезды на объекты.

Неделя обучения	1	2	3	4	5	6	7	Итого часов
	пн	вт	ср	чт	пт	сб	вс	
1 неделя	-	-	-	-	-	4	-	4
СРС	-	-	-	-	-	1	-	1
2 неделя	-	-	-	-	-	4	-	4
СРС	-	-	-	-	-	1	-	1
3 неделя	-	-	-	-	-	4	-	4
СРС	-	-	-	-	-	1	-	1
4 неделя	-	-	-	-	-	4	-	4
СРС	-	-	-	-	-	1	-	1
5 неделя	-	-	-	-	-	4ИА	-	4
Итого:	-	-	-	-	-	24	-	24
Примечание: ИА – Итоговая аттестация (портфолио, практическое занятие)								

6. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Реализация дополнительной общеобразовательной (общеразвивающей) программы для детей и взрослых осуществляется в соответствии с требованиями российского законодательства, нормативными актами Министерства образования науки РФ, уставом ОЧУ «Специалист.РУ», лицензией на осуществление образовательной деятельности.

6.1. Требования к квалификации педагогических кадров

Реализация дополнительной общеобразовательной (общеразвивающей) программы для детей и взрослых обеспечивается руководящими и научно-педагогическими работниками организации, а также лицами, привлекаемыми к реализации программы на условиях гражданско-правового договора.

Требования к квалификации педагогических кадров, обеспечивающих реализацию образовательного процесса:

- наличие ученой степени (ученого звания);
- опыт работы в образовательной организации высшего образования не менее 5 лет;
- наличие документа о прохождении повышения квалификации или профессиональной переподготовки за последние 3 года.

Сведения о профессорско-преподавательском составе, обеспечивающего реализацию программы профессиональной переподготовки содержатся в Приложении 2.

6.2. Материально-техническая и ресурсная база

Для реализации программы используется собственная материально-техническая база.

Для реализации программы предполагается использование имеющихся в ОЧУ «Специалист.РУ» учебных аудитории для проведения занятий лекционного типа, занятий семинарского типа, выполнения проектных работ, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования:

- Компьютерные классы, обеспечивающие доступ в Интернет
- Лекционная аудитория, оснащенная проектором или иными средствами визуализации учебного материала.
- Доступ в электронную информационно-образовательную среду ОЧУ «Специалист.РУ» в систему дистанционного обучения.
- Стандартное программное обеспечение для работы над разработкой учебно-методических материалов.
- Мастерские и аудитории для проведения открытых занятий.

Специальных помещений, предполагающих наличие какого-либо специального оборудования для реализации данной программы, не предусматривается.

6.3. Общие требования к организации образовательного процесса

Обучение предполагает контактную работу с преподавателем (лекции, практические занятия), и асинхронное взаимодействие, предполагающее использование современных дистанционных образовательных технологий.

Активные формы занятий включают как теоретическую часть (направленную на получение или пополнение имеющихся знаний), так и практическую. Предусматривается активное использование в учебном процессе инновационных образовательных технологий, активных и интерактивных форм проведения занятий.

В процессе освоения дополнительной общеобразовательной (общеразвивающей) программы для детей и взрослых, обучающиеся формируют портфолио по результатам освоения учебных модулей.

Итоговая аттестация предполагает определение уровня подготовленности обучающихся на основе оценки портфолио, практической работы (результат деятельности) и/или теста.

Образовательная деятельность обучающихся предусматривает следующие виды учебных занятий и учебных работ: лекции, практические и семинарские занятия, круглые столы, мастер-классы, тренинги, семинары по обмену опытом, проведение открытых занятий, консультации и другие виды учебных занятий и методической работы, определенные учебным планом.

По завершении реализации программ, как правило, проводится анкетирование обучающихся с целью изучения мнения по вопросу эффективности и информативности проведенного обучения, уровню организации учебного процесса, удовлетворенности учебно-методическим материалом, работниками образовательной организации проводится анализ высказанных предложений и пожеланий.

7. ФОРМЫ АТТЕСТАЦИИ, ОЦЕНОЧНЫЕ МАТЕРИАЛЫ И КРИТЕРИИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ

7.1. Освоение образовательной программы, в том числе отдельной части или всего объема учебного предмета, курса, дисциплины (модуля) образовательной программы, сопровождается текущим контролем успеваемости и промежуточной аттестацией слушателей².

Итоговая аттестация настоящей программой предусмотрена, предполагает проверку результатов освоения общеразвивающей программы для детей и взрослых «Основы программирования на языке "Java" для школьников. Модуль 2» и проводится в форме **практической работы по созданию игры «Предсказание будущего»**, самостоятельно разработанного обучающимся.

Создание игры "Предсказание будущего" с использованием базы данных.

Создание базы данных и таблицы для хранения данных игры.

Добавление данных в таблицу.

Построение интерфейса пользователя игры.

Получение случайной строки таблицы базы данных из приложения Java.

Практическая работа обучающегося – это файловая папка (электронная) с документально подтвержденной информацией о личных, общественно значимых и академических достижениях учащегося, продемонстрированных за время освоения ДО программы. Эти сведения служат одним из инструментов объективного оценивания

² «Положение о проведении промежуточной аттестации слушателей и осуществлении текущего контроля их успеваемости» ОЧУ «Специалист.Ру», п.1.3., п.3.3.

результатов обучения. Методика особенно эффективна для прикладных направлений: архитектуры, дизайна, технологий, программирования, строительства, журналистики и т. д. Портфолио/практическая работа или папка личных достижений, способствует объективности оценки результатов, достигнутых обучающимся в ходе учебной деятельности. Этот своеобразный презентационный портфель документов (кодов, классов и др.) может формироваться в бумажном и/или электронном виде.

№/п	Конкретные задания/выполнение работы	Баллы за выполнение
1	Создание базы данных и таблицы для хранения данных игры	3 балл
2	Добавление данных в таблицу	2 балл
3	Построение интерфейса пользователя игры.	3 балл
4	Получение случайной строки таблицы базы данных из приложения Java.	3 балла
ИТОГО		0-7 не зачтено 8-11 -зачтено

7.2. Оценочные материалы для итоговой аттестации

7.2.1. Критерии и показатели результата обучения

Компоненты освоения программы	Результат обучения			
	«Не зачтено» (0-7 баллов)		«Зачтено» (8-11 баллов)	
знания	Отсутствие знаний	Неполные знания	Полные знания с небольшими пробелами	Системные и глубокие знания
умения	Отсутствие умений	Частичные умения	Умения с частичными пробелами	Полностью сформированные умения
навыки	Отсутствие навыков	Частичные навыки	Отдельные пробелы в навыках	Полностью сформированные навыки

7.2.2. Балльная система оценивания освоения программы

Содержание элементов оценивания освоения программы	Баллы
Освоение модулей №1-№8	0-7
Практическая работа (ИА – отражает результат освоения программы на основе создания игры)	8-11
Итоговая оценка: «Не зачтено» «Зачтено»	0-7 баллов 8-11 баллов

8. СПИСОК ЛИТЕРАТУРЫ

Методические материалы к курсу обучения.

1. "Основы программирования на языке Java для школьников" Модуль 1
 2. "Основы программирования на языке Java для школьников" Модуль 2
- Интернет ресурс: <http://v8.1c.ru>

- Нормативная документация в электронной форме:
 - Федеральный закон от 27 июля 2006 г. N 152-ФЗ "О персональных данных"
 - План мероприятий "Развитие отрасли информационных технологий", утвержденный распоряжением Правительства России от 30.12.2013 №2602-р.

- Профильные периодические электронные издания:
 - ИТС:ИТС (Информационно-технологическое сопровождение пользователей ИС:Предприятия), <http://its.1c.ru/>
 - <http://v8.1c.ru/>

- Список рекомендуемой литературы:

- Герберт Шилдт, "Java. Руководство для начинающих"

Приложение 1.

Методические материалы к модулям 1-11.

Java является объектно-ориентированным языком, поэтому такие понятия как "класс" и "объект" играют в нем ключевую роль. Любую программу на Java можно представить, как набор взаимодействующих между собой объектов.

Шаблон или описанием объекта является класс, а объект представляет экземпляр этого класса. Можно еще провести следующую аналогию. У нас у всех есть некоторое представление о человеке - наличие двух рук, двух ног, головы, туловища и т.д. Есть некоторый шаблон - этот шаблон можно назвать классом. Реально же существующий человек (фактически экземпляр данного класса) является объектом этого класса.

Класс определяется с помощью ключевого слова `class`:

```
class Person{  
  
}
```

В данном случае класс называется `Person`. После названия класса идут фигурные скобки, между которыми помещается тело класса - то есть его поля и методы.

Вся функциональность класса представлена его членами - полями (полями называются переменные класса), которые хранят состояние объекта, и методами, которые определяют поведение объекта. Например, класс `Person`, который представляет человека, мог бы иметь следующее определение:

```
class Person{  
  
    String name;           // имя  
    int age;               // возраст  
    void displayInfo(){
```

```

        System.out.printf("Name: %s \tAge: %d\n", name, age);
    }
}

```

В классе `Person` определены два поля: `name` представляет имя человека, а `age` - его возраст. И также определен метод `displayInfo`, который ничего не возвращает и просто выводит эти данные на консоль.

Теперь используем данный класс. Для этого определим следующую программу:

```

public class Program{

    public static void main(String[] args) {

        Person tom;
    }
}
class Person{

    String name;        // имя
    int age;            // возраст
    void displayInfo(){
        System.out.printf("Name: %s \tAge: %d\n", name, age);
    }
}

```

Как правило, файлы определяются в разных файлах. В данном случае для простоты мы определяем два класса в одном файле. Стоит отметить, что в этом случае только один класс может иметь модификатор `public` (в данном случае это класс `Program`), а сам файл кода должен называться по имени этого класса, то есть в данном случае файл должен называться `Program.java`.

Класс представляет новый тип, поэтому мы можем определять переменные, которые представляют данный тип. Так, здесь в методе `main` определена переменная `tom`, которая представляет класс `Person`. Но пока эта переменная не указывает ни на какой объект и по умолчанию она имеет значение `null`. По большому счету мы ее пока не можем использовать, поэтому вначале необходимо создать объект класса `Person`.

Конструкторы

Кроме обычных методов классы могут определять специальные методы, которые называются конструкторами. Конструкторы вызываются при создании нового объекта данного класса. Конструкторы выполняют инициализацию объекта.

Если в классе не определено ни одного конструктора, то для этого класса автоматически создается конструктор без параметров.

Выше определенный класс `Person` не имеет никаких конструкторов. Поэтому для него автоматически создается конструктор по умолчанию, который мы можем использовать для создания объекта `Person`. В частности, создадим один объект:

```

public class Program{

```

```

public static void main(String[] args) {

    Person tom = new Person(); // создание объекта
        tom.displayInfo();

        // изменяем имя и возраст
        tom.name = "Tom";
        tom.age = 34;
        tom.displayInfo();
    }
}
class Person{

    String name;        // имя
    int age;            // возраст
    void displayInfo(){
        System.out.printf("Name: %s \tAge: %d\n", name, age);
    }
}

```

Для создания объекта `Person` используется выражение `new Person()`. Оператор `new` выделяет память для объекта `Person`. И затем вызывается конструктор по умолчанию, который не принимает никаких параметров. В итоге после выполнения данного выражения в памяти будет выделен участок, где будут храниться все данные объекта `Person`. А переменная `tom` получит ссылку на созданный объект.

Если конструктор не инициализирует значения переменных объекта, то они получают значения по умолчанию. Для переменных числовых типов это число 0, а для типа `string` и классов - это значение `null` (то есть фактически отсутствие значения).

После создания объекта мы можем обратиться к переменным объекта `Person` через переменную `tom` и установить или получить их значения, например, `tom.name = "Tom"`.

В итоге мы увидим на консоли:

```

Name: null           Age: 0
Name: Tom            Age: 34

```

Если необходимо, что при создании объекта производилась какая-то логика, например, чтобы поля класса получали какие-то определенные значения, то можно определить в классе свои конструкторы. Например:

```

public class Program{

    public static void main(String[] args) {

        Person bob = new Person();           // вызов первого конструктора без параметров
            bob.displayInfo();

        Person tom = new Person("Tom");     // вызов второго конструктора с одним
параметром
            tom.displayInfo();
    }
}

```

```

        Person sam = new Person("Sam", 25); // вызов третьего конструктора с двумя
параметрами
        sam.displayInfo();
    }
}
class Person{
    String name;    // имя
    int age;        // возраст
    Person()
    {
        name = "Undefined";
        age = 18;
    }
    Person(String n)
    {
        name = n;
        age = 18;
    }
    Person(String n, int a)
    {
        name = n;
        age = a;
    }
    void displayInfo(){
        System.out.printf("Name: %s \tAge: %d\n", name, age);
    }
}

```

Теперь в классе определено три конструктора, каждый из которых принимает различное количество параметров и устанавливает значения полей класса.

Консольный вывод программы:

```

Name: Undefined          Age: 18
Name: Tom                Age: 18
Name: Sam                Age: 25

```

Ключевое слово this

Ключевое слово `this` представляет ссылку на текущий экземпляр класса. Через это ключевое слово мы можем обращаться к переменным, методам объекта, а также вызывать его конструкторы. Например:

```

public class Program{

    public static void main(String[] args) {

        Person undef = new Person();
        undef.displayInfo();
    }
}

```

```

        Person tom = new Person("Tom");
        tom.displayInfo();

        Person sam = new Person("Sam", 25);
        sam.displayInfo();
    }
}
class Person{
    String name;        // имя
    int age;            // возраст
    Person()
    {
        this("Undefined", 18);
    }
    Person(String name)
    {
        this(name, 18);
    }
    Person(String name, int age)
    {
        this.name = name;
        this.age = age;
    }
    void displayInfo(){
        System.out.printf("Name: %s \tAge: %d\n", name, age);
    }
}

```

В третьем конструкторе параметры называются так же, как и поля класса. И чтобы разграничить поля и параметры, применяется ключевое слово `this`:

```
this.name = name;
```

Так, в данном случае указываем, что значение параметра `name` присваивается полю `name`.

Кроме того, у нас три конструктора, которые выполняют идентичные действия: устанавливают поля `name` и `age`. Чтобы избежать повторов, с помощью `this` можно вызвать один из конструкторов класса и передать для его параметров необходимые значения:

```

Person(String name)
{
    this(name, 18);
}

```

В итоге результат программы будет тот же, что и в предыдущем примере.

Инициализаторы

Кроме конструктора начальную инициализацию объекта вполне можно было проводить с помощью инициализатора объекта. Инициализатор выполняется до любого конструктора. То есть в инициализатор мы можем поместить код, общий для всех конструкторов:

```
public class Program{

    public static void main(String[] args) {

        Person undef = new Person();
            undef.displayInfo();

            Person tom = new Person("Tom");
            tom.displayInfo();

        }
    }
class Person{

    String name;        // имя
    int age;            // возраст

    /*начало блока инициализатора*/
    {
        name = "Undefined";
        age = 18;
    }
    /*конец блока инициализатора*/
    Person(){

    }
    Person(String name){

        this.name = name;
    }
    Person(String name, int age){

        this.name = name;
        this.age = age;
    }

    void displayInfo(){
        System.out.printf("Name: %s \tAge: %d\n", name, age);
    }
}
}
```