

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ НЕТИПОВОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОРОДСКОЙ ДВОРЕЦ ТВОРЧЕСТВА ЮНЫХ»

ПРИНЯТО

Протокол Малого педагогического совета
№ 8 от «05» 06 2017 года


/М.Ю. Колганов
Руководитель структурного подразделения

УТВЕРЖДЕНО

Приказ № 1533-04 от «30» 08 2017 года

Генеральный директор

М.Р. Катунцова
М.П.

Дополнительная общеобразовательная общеразвивающая программа

ОСНОВЫ ПРОФЕССИОНАЛЬНОГО ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ C++

Срок реализации программы: 1 год
Возраст учащихся 13-16 лет

Разработчик
Гузенко Петр Юрьевич,
педагог дополнительного образования

ОДОБРЕНО

Протокол Методического совета
№ 14 от «30» 08 2017 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программ «Основы профессионального программирования на языке С++» (далее Программа) имеет **техническую** направленность и предназначена для изучения основ программирования на языке высокого уровня. Данная программа является приведением реализуемой дополнительной общеразвивающей одноименной программы в соответствие с методическими рекомендациями (п.2.2 распоряжения КОбр СПб №617-р от 01.03.2017)

Актуальность программы,

По мнению автора, обусловлена необходимостью для обучающегося, освоившего фундаментальные аспекты программирования, овладеть также самым распространенным языком профессионального программирования, реализованным на всех широко распространенных платформах, -- языком программирования С++ для умения создавать программный продукт в быстро меняющихся (в основном в сторону усложнения) средствах разработки. Данная программа является существенно переработанным и дополненным курсом, преподаваемым автором в рамках центра компьютерных технологий отдела техники СПбГДТЮ с 1994г. Аналогичные по тематике курсы входят в обязательный набор дисциплин при подготовке и переподготовке программистов любого уровня выше ознакомительного.

Новизна данной программы на год начала реализации в представленном виде состояла в подготовке и успешной реализации обучения школьников старшего школьного возраста основам профессионального программирования на языке С++, то есть в достижении повышенного уровня образованности в области разработки приложений на языке С++ в одной или нескольких средах разработки, что составляет основу профессионализма программиста.

Педагогическая целесообразность данной программы обусловлена использованием при её разработке и реализации тезисов о дополнительном образовании как средстве творческого развития и о развитии школьников в личностно-ориентированном учебно-воспитательном процессе, а также учетом многолетнего успешного опыта работы центра компьютерных технологий в области обучения программированию.

Отбор материала для программы в основном обусловлен внутренней логикой промышленного стандарта современной технологии программирования -- объектно-ориентированного программирования, анализа и проектирования (ООР, ООА, ООД). Язык С++ в этой области содержит наиболее полный набор возможностей (кроме обычной триады ООП – инкапсуляция, наследование, полиморфизм – есть еще перегрузка операций, обобщенное программирование в виде шаблонов функций и классов, а также обширная стандартная библиотека шаблонов STL).

Отличительные особенности

- отсутствие образовательной программы с аналогичным подбором материала среди известных автору образовательных программ (отдел техники, другие отделы СПбГДТЮ)
- соответствие отобранного материала основам профессионального программирования, в том числе наличие набора решенных задач из литературы.
- использование одновременных «линий» освоения разделов отобранного материала, облегчающих освоение программы учащимися с несколько различной подготовкой (в пределах освоенных ранее образовательных программ 1-2 годов обучения) и способностями. Этими «линиями» являются: - лекционный материал и задания на практическое занятие по лекционному материалу; задания по освоению выбранной среды визуальной разработки; реализация собственного проекта для представления на конференциях/конкурсах/смотре; задания повышенной сложности (по желанию учащегося); в начале учебного года – предлагаемый к быстрой самостоятельной реализации список стандартных задач, не дублирующих лекционный материал (приложение 3).
- наличие одновременно реализуемых линий требует несколько увеличенного времени для практических занятий: 2 ак.ч./нед. для работы с заданиями по теории, 2 ак.ч./нед. на практические индивидуально-групповые занятия консультативного характера.

Уровень освоения программы: базовый, (в смысле приложения 3 к методическим рекомендациям, утв. распоряжением КОБр СПб №617-р от 01.03.2017), а по сути профессионально-ориентированный уровень освоения, а именно на достижение повышенного уровня образованности в области разработки приложений на языке С++ в одной или нескольких средах разработки, умения формулировать задачи реализации программной разработки и находить средства их решения. В рамках освоения дополнительной общеобразовательной общеразвивающей программы результатом является не только демонстрация собственной компьютерной разработки и представление на итоговой конференции Центра компьютерных технологий, но и участие в конкурсах городского и всероссийского уровня:

-Всероссийская конференция "Будущее сильной России - в высоких технологиях"

-Международная конференция "Школьная информатика и проблемы устойчивого развития"

-Городской конкурс школьников по программированию и компьютерным работам-номинация "Программирование"

Адресат программы данная программа предназначена для учащихся 13-16 лет, учащимся 9-11 (как исключение 8) классов, прошедшие 2 года обучения по языку Паскаль и 1 год программирования на С++ на базе ЦКТ отдела техники либо в других учреждениях, уверенно владеющим основными понятиями программирования на императивных языках в процедурной парадигме (типы данных, операторы управления, простые структуры данных и алгоритмы для них, подпрограммы, передача параметров).

Цель программы

Основной целью данной программы применительно к конкретным условиям обучения, сложившимся в центре компьютерных технологий отдела техники СПбГДТЮ, является формирование и развитие мотивации учащихся к техническому творчеству и овладению основами профессионального программирования.

Задачи программы

Для достижения поставленной цели в процессе обучения по данной программе решаются следующие задачи:

- обучающие:

-- знакомство с реализацией языка программирования С++ при использовании различных сред разработки, от устаревшей среды начального уровня BorlandC++ до относительно современных сред фирм Borland (C Builder v.6+) и/или Microsoft (Visual Studio 2005-2013).

-- знакомство с концепцией объектно-ориентированного анализа и проектирования, паттернами проектирования

-- доведение до профессионального уровня навыков формализации и реализации решения поставленных задач по созданию работающих программных продуктов для ЭВМ

-- дальнейшее развитие навыков постановки задач по созданию работающих приложений для ПЭВМ

- развивающие:

-- развитие навыков формализации и реализации решения поставленных задач по созданию работающих программ для ЭВМ

-- развитие навыков постановки задач по созданию работающих программ для ПЭВМ

-- развитие способностей к логическому мышлению с помощью задач программирования

-- развитие способности к долговременной упорной работе над проектами малого и среднего (по меркам реальных проектов) размера

- воспитательные:

-- воспитание целеустремленности, силы воли и настойчивости при овладении абстрактными математическими и алгоритмическими конструкциями как при участии педагога, так и

самостоятельно

- формирование навыков алгоритмизации деятельности как средства достижения результатов не только в области программирования
- понимание неизбежности постоянного овладения вновь появляющимися технологиями разработки программного обеспечения как условия самореализации в этой сфере деятельности
- формирование коммуникативных навыков при представлении своих разработок на мероприятиях
- формирование навыков сетевой коммуникативной культуры в сфере профессионального общения программистов через Интернет.

Условия реализации программы

Условия набора и формирование групп: Принимаются учащиеся 13-16 лет, имеющие подготовку 2 года обучения по языку Паскаль и 1 год программирования на С++ на базе ЦКТ отдела техники либо в других учреждениях, уверенно владеющим основными понятиями программирования на императивных языках в процедурной парадигме (типы данных, операторы управления, простые структуры данных и алгоритмы для них, подпрограммы, передача параметров).

Группы учащихся набираются без конкурсного отбора с учетом наличия знаний и навыков, по рекомендации, полученной на предыдущем году обучения программированию или по результатам собеседования. Учебная нагрузка в неделю составляет в соответствии с требованиями СанПиН 2.2.2/2.4.1340-03 и 2.4.4.3172-14: 2 ак.ч. на теоретическое занятие и по 2 ак.ч. на каждое практическое занятие. Списочный состав формируется в соответствии с нормативно-правовыми основами проектирования общеобразовательных программ в т.ч. СанПиН 2.4.4.3172-14. Допускается проведение практических занятий подгруппами в разное время при проведении теоретических занятий для обеих подгрупп одновременно.

Срок реализации программы: один учебный год, 216 часов.

Особенности образовательной среды по преподаванию языков программирования Pascal и С++

В секторе информатики отдела техники сложились определенные традиции преподавания программирования. По опыту преподавания языков программирования Pascal (с 1989г.) и С++ (с 1994г.) автор считает возможным охарактеризовать образовательную среду этой части преподавания программирования следующим образом. Большинство педагогов работают по авторским программам, отражающим личный опыт и предпочтения их авторов. Имеет место также сложившаяся специализация по годам обучения. Следует отметить возрастание возможностей реализации творческой деятельности учащегося по годам обучения, связанной с ростом освоенных знаний. Кульминацией этих возможностей (в рамках нормативных документов СПбГДТЮ и отдела техники) традиционно является третий год обучения программированию (по одной из соответствующих программ).

Имеет место также расслоение учащихся на превосходящих средний уровень группы, соответствующих ему и на учащихся, с трудом осваивающих предлагаемые темы.

Характеризуя выдаваемый учащимися программный продукт, можно разделить учащихся на склонных к решению алгоритмически сложных (и как правило практически малополезных) задач и на склонных к созданию алгоритмически менее сложных проектов, имеющих преимущественно практическую направленность (судя по названиям представленных на итоговой конференции разработок). Для первых из упомянутых (их, как правило, немного) естественный путь продолжения обучения – группы решения олимпиадных задач, функционирующие в отделе техники. При этом традицией преподавания программирования, видимо, следует считать прикладную ориентацию разработок учащихся, оцениваемую жюри конференций.

Стандартной для второго десятилетия XXI века особенностью образовательной среды данного коллектива следует считать наличие доступа в глобальную сеть Интернет,

имеющегося в СПбГДТЮ и регулируемого нормативными документами отдела техники. Автор данной программы считает необходимым предоставление ограниченного доступа в Интернет в учебных целях (прежде всего – для вовлечения в профессиональное сообщество программистов путем участия в соответствующих форумах по различным средам разработки на C++ и поиска решений конкретных неочевидных проблем при реализации итоговой программы). Рекомендуемые сетевые ресурсы: <http://www.programmersclub.ru>, <http://www.firststeps.com>, <http://www.progclub.ru>, <http://kychka-pc.ru/category/sfml>.

Материально-техническое оснащение Аудитория для теоретических занятий и классы ПЭВМ для практических занятий должны удовлетворять требованиям СанПиН 2.2.2/2.4.1340-03 и 2.4.4.3172-14.

В классе для теоретических занятий желательно наличие двух фломастерных досок, т.к. конспект одного теоретического занятия полностью уместится на них.

В компьютерном классе необходимо наличие достаточного для удовлетворения требованиям СанПиН 2.2.2/2.4.1340-03 и 2.4.4.3172-14 числа объединенных в локальную сеть ПЭВМ клона PC конфигурации, достаточной для эксплуатации Windows-подобной операционной системы (Windows XP/7/8) с функционирующей средой разработки Borland CBuilder v.5+ и/или Microsoft Visual Studio 2005 – 2013 (достаточно свободно распространяемых express-версий с приложением Windows Forms в списке шаблонов проектов VisualStudio).

В обоих классах желательно наличие проектора и интерактивной доски.

Для реализации раздачи заданий и учебных материалов на практические занятия и контроля результатов педагог должен иметь возможность просматривать содержимое рабочих каталогов учащихся (независимо от местонахождения каталогов – на отдельных машинах или на файл-сервере) и иметь один каталог (напр., с именем FOR103), доступного учащимся только для чтения и предназначенного для помещения педагогом заданий, учебных материалов, комментариев и ответов к заданиям. Доступ в сеть Интернет необходим для реализации некоторых задач программы, однако освоение собственно программирования на языке C++ возможно и без доступа в Интернет. Весьма желательно наличие одного принтера и устройств чтения/записи оптических дисков на компьютерный класс.

Особенности организации образовательного процесса: заключаются в применении современных образовательных технологий, а именно применение технологии проектного обучения в Теме 8. Практические консультационные занятия по освоению выбранной среды визуальной разработки приложений (CBuilder v.6 или VisualStudio 2005-2013), выполнению заданий повышенной сложности и по подготовке компьютерных программ учащихся к “Ежегодной итоговой конференции Компьютерного центра” и/или городской конференции по программированию

Форма занятий:

“инструктаж” проводится на первом занятии согласно действующим инструкциям “теоретическое занятие” (оно же “лекция”) проводится в классе для аудиторных занятий и служит основной формой сообщения нового материала согласно конспекту лекций “практическое занятие” (оно же “практика”) проводится в компьютерном классе и служит в основном для реализации приобретенных теоретических сведений в виде выполнения заданий из плана практических занятий, приобретения навыков самостоятельной работы под руководством педагога. Учащиеся индивидуально выполняют обязательные задания по пройденному материалу с предъявлением результата педагогу. Сценарий выполнения таких заданий предусматривает изучение примера с лекции и на его основе решение поставленной задачи. Как правило, предусмотрено 2-3 последовательных шага для выполнения поставленной задачи модификацией программы, реализующей предыдущий шаг. Сложность выполнения последующих шагов в таких заданиях может возрасти нелинейно. “практическое занятие - индивидуальная консультация” проводится в компьютерном классе во время практики и служит для освоения выбранной среды визуальной разработки, для подготовки собственной разработки к итоговой конференции, для ответов на вопросы

учащихся, возникших при реализации задания и/или при подготовке своей программы к к.-л. конференции и для самостоятельного изучения дополнительных тем в рамках изучаемого направления.

"конференция" проводится администрацией при участии педагога и позволяет не только продемонстрировать результат своей работы, но и сравнить его с результатами других участников конференции – в т.ч. и по оценке жюри.

На теоретических занятиях используются словесный и наглядный метод (по типу источника), метод получения и закрепления знаний (по виду дидактических задач). На практических занятиях используются практический метод (по виду источника), метод проверки знаний и умений (по виду дидактических задач).

Форма организации деятельности учащихся на занятии: фронтальная (проведение лекции со всем составом учащихся), групповая (проведение занятия в группах), индивидуально-групповая (индивидуальные консультации).

Занятия проводятся 2 раза в неделю по 2 академических часа (1-ый час — теория, работа с конспектом и у доски, перерыв 10 минут (возможна физкультминутка), 2-ой час — опрос по теме, практика, выполнение упражнений по теме).

Планируемые результаты.

Предметные

Учащиеся :

- приобретут глубокие знания языка программирования C++ при использовании различных сред разработки
- освоят технологию объектно-ориентированного программирования в одном из наиболее полных её реализаций
- получат представление об основах объектно-ориентированного анализа и проектирования, о паттернах проектирования (design patterns)

Метапредметные

- разовьют навыки формализации и реализации решения поставленных задач по созданию работающих программ для ЭВМ
- разовьют навыки постановки задач по созданию работающих программ для ПЭВМ
- разовьют способности к логическому мышлению с помощью задач программирования
- разовьют способности к долговременной упорной работе над проектами малого и среднего (по меркам реальных проектов) размера

Личностные

Учащиеся:

- приобретут практические навыки по разработке и доведению до готовности программного продукта с представлением его на суд жюри итоговой конференции сектора информатики
- сформируют ценностные отношения к современным компьютерным технологиям и постоянному повышению собственного образовательного уровня.
- освоят методы поиска нужной для реализации собственной разработки информации и навыки общения с программистами в сети Интернет
- приобретут коммуникативные навыки при подготовке доклада для конференций и при совместной работе (по желанию)

УЧЕБНЫЙ ПЛАН

Название раздела, темы	Количество ак. часов			Формы контроля
	Теория	Практика	Всего	
Тема 0 Знакомство с нормативными документами СПбГДТЮ для учащихся, правилами работы в сетях и историей СПбГДТЮ и отдела техники	2	4	6	Беседа устный опрос
Тема 1 Вводное занятие по первоначальному ознакомлению с компьютером, составом программного обеспечения и средой разработки	2	4	6	устный опрос
Тема 2: Углубленное повторение основных понятий программирования и изучение их реализации в языке C++	18	20	38	предъявление примеров с теории
Тема 3: объектно-ориентированное программирование в C++	28	34	62	устный опрос предъявление примеров с теории
Тема 4: Обобщенное программирование: шаблоны, STL	8	6	14	устный опрос предъявление примеров с теории
Тема 5: Введение в объектно-ориентированный анализ и проектирование	6	---	6	устный опрос
Тема 6. Практические консультационные занятия по освоению выбранной среды визуальной разработки приложений (CBuilder v.6 или VisualStudio 2005-2013), выполнению заданий повышенной сложности и по подготовке компьютерных программ учащихся к “Ежегодной итоговой конференции Компьютерного центра” и/или городской конференции по программированию	---	60	60	Проверка самостоятельной работы в выбранной среде разработки по электронным ресурсам
Проверочные и консультационные занятия (осенние каникулы)	2	4	6	Нет
Проверочные и консультационные занятия (зимние каникулы)	2	4	6	Нет
Проверочные и консультационные занятия (весенние каникулы)	2	4	6	Нет
Тема 7. Заключительное занятие: итоги конференции и учебного года	2	4	6	предъявление собственной разработки
Итого занятий	36	72	108	
Итого часов	72	144	216	

РАБОЧАЯ ПРОГРАММА

к дополнительной общеобразовательной общеразвивающей программе
«Основы профессионального программирования на языке С++»

Особенности обучения В результате обучения учащиеся выполняют большие самостоятельные проекты, рассчитанные на длительное время работы. Созданные программы должны быть представлены для оценки либо на городской конференции школьников по программированию, либо на итоговой конференции компьютерного центра отдела техники в конце учебного года.

В результате выполнения учебных проектов учащиеся научатся работать над постановкой и алгоритмизацией задач, планировать свое время на выполнение проекта, творчески представлять итоги своей деятельности.

Задачи

- обучающие:

- знакомство с реализацией языка программирования С++ при использовании различных сред разработки, от устаревшей среды начального уровня BorlandC++ до относительно современных сред фирм Borland (С Builder v.6+) и/или Microsoft (Visual Studio 2005-2013).
- знакомство с концепцией объектно-ориентированного анализа и проектирования, паттернами проектирования
- доведение до профессионального уровня навыков формализации и реализации решения поставленных задач по созданию работающих программных продуктов для ЭВМ
- дальнейшее развитие навыков постановки задач по созданию работающих приложений для ПЭВМ

- развивающие:

- развитие навыков формализации и реализации решения поставленных задач по созданию работающих программ для ЭВМ
- развитие навыков постановки задач по созданию работающих программ для ПЭВМ
- развитие способностей к логическому мышлению с помощью задач программирования
- развитие способности к долговременной упорной работе над проектами малого и среднего (по меркам реальных проектов) размера

- воспитательные:

- воспитание целеустремленности, силы воли и настойчивости при овладении абстрактными математическими и алгоритмическими конструкциями как при участии педагога, так и самостоятельно
- формирование навыков алгоритмизации деятельности как средства достижения результатов не только в области программирования
- понимание неизбежности постоянного овладения вновь появляющимися технологиями разработки программного обеспечения как условия самореализации в этой сфере деятельности
- формирование коммуникативных навыков при представлении своих разработок на мероприятиях
- формирование навыков сетевой коммуникативной культуры в сфере профессионального общения программистов через Интернет.

Планируемые результаты.

Предметные

Учащиеся :

- приобретут глубокие знания языка программирования С++ при использовании различных сред разработки
- освоят технологию объектно-ориентированного программирования в одном из наиболее

полных её реализаций

- получают представление об основах объектно-ориентированного анализа и проектирования, о паттернах проектирования (design patterns)

Метапредметные:

-- разовьют навыки формализации и реализации решения поставленных задач по созданию работающих программ для ЭВМ

-- разовьют навыки постановки задач по созданию работающих программ для ПЭВМ

-- разовьют способности к логическому мышлению с помощью задач программирования

-- разовьют способности к долговременной упорной работе над проектами малого и среднего (по меркам реальных проектов) размера

В результате обучения учащиеся выполняют большие самостоятельные проекты, рассчитанные на длительное время работы. Созданные программы должны быть представлены для оценки либо на городской конференции школьников по программированию, либо на итоговой конференции Компьютерного центра отдела техники в конце учебного года.

Личностные

Учащиеся:

- приобретут практические навыки по разработке и доведению до готовности программного продукта с представлением одного на суд жюри итоговой конференции сектора информатики

- сформируют ценностные отношения к современным компьютерным технологиям и постоянному повышению собственного образовательного уровня.

- освоят методы поиска нужной для реализации собственной разработки информации и навыки общения с программистами в сети Интернет

- приобретут коммуникативные навыки при подготовке доклада для конференций и при совместной работе (по желанию)

Содержание программы:

Сокращения: т теория п практика ПК практика-консультация

Тема предварительная (т1, п1, ПК1)

Знакомство с нормативными документами СПбГДТЮ для учащихся, правилами работы в сетях и историей СПбГДТЮ и отдела техники

Т1: Знакомство с нормативными документами СПбГДТЮ для учащихся, правилами работы в сетях и историей СПбГДТЮ и отдела техники

- знакомство с правилами для учащихся СПбГДТЮ, историей, традициями и достижениями СПбГДТЮ и отдела техники

П1, ПК1 : - инструктаж по правилам техники безопасности, пожарной безопасности, правилам дорожного движения, знакомство с правилами работы в комп. классах отдела техники (в т.ч. с правилами работы в локальной сети ОТ и нормами пользования сетью Интернет).

Тема 1 Вводное занятие по первоначальному ознакомлению с компьютером, составом программного обеспечения и средой разработки (т2, п2, ПК2)

Т2: - система авторизации доступа и раздачи учебных материалов в комп. классе

П2, ПК2:- порядок запуска среды разработки, хранения, переноса и работы для собственных проектов

Тема 2: Углубленное повторение основных понятий программирования и изучение их реализации в языке С++ (т3-11 (9 теор. занятий) п3-12: (10 практ. занятий))

т2.1 Объявления, определения, область видимости и время жизни объектов С++ программ

п2.1 Примеры: [1] #2.1 Упражнения : #2.7: 1,2,8. Самостоятельно: #2.7: 5,9.

т2.2 Операции объявления ссылки и указателя, массива, функции, структуры.

п2.2 Примеры: [1] #2.3.3-2.3.7, 2.3.10 Упражнения : #2.7: 3,4,10,13.

т2.3 Связь массивов и указателей. Использование ссылок при передаче параметров встроженных типов. Символы и строковые константы.

п2.3 Примеры: [1] #2.7: 6,7,14.

т2.4 Техника работы со строками, оканчивающимися нулем

п2.4 Примеры: [1] #2.4.3-2.4.5 Упражнения : #2.7:11. Самостоятельно: #2.7:12.

т2.5 Выражения, операции и операторы (условный, выбора, циклов) в C++.

п2.5, п.2.6 Упражнения : [1] #3.5:1,2,5,6,7,9. Самостоятельно: #3.5: 3,4,10

т2.6 "Настольный калькулятор": грамматика и лексический анализатор

п2.7 Примеры: [1] #3.1.1-3.1.3 ("настольный калькулятор")

т2.7 "настольный калькулятор", синтаксический анализатор и таблица имен)

п.2.8 Примеры: [1] #3.1.4-3.1.5 ("настольный калькулятор") Самостоятельно: #3.5: 11-15

т2.8 Функции: возврат значения и векторные параметры.

п2.9 Примеры: [1] #4.6.1-4.6.3 Упражнения : #4.8:1,3,8 Самостоятельно: #4.8:2,4

т2.9 Функции: перегрузка имен и параметры по умолчанию. Указатели на функции. Макросы.

п2.10 Примеры: [1] #4.6.6-4.6.9 Упражнения : #4.8:6,16 Самостоятельно: #4.8: 9,14,15,19

Тема 3: объектно-ориентированное программирование в C++ (т12-25 (14 теор. занятий) п13-29 (17 практ. занятий))

т3.1 Классы C++ : первое знакомство.

п3.1 Примеры: [1] #5.1; 5.2.1-5.2.3

т3.2 Классы C++ : инициализация и очистка объектов класса.

п3.2 Примеры: [1] #5.2.4-5.2.6

т3.3 Классы C++ : интерфейсы и реализация (таблица имен, множество во целых).

п3.3 Примеры: [1] #5.3; [2] #9.1с Упражнения : [1] #5.6:1,3 Самостоятельно: #5.6:2,4

т3.4 Второстепенные свойства класса.(friend functions, static members, структуры (struct) и объединения (union) как классы.)

п3.4 Примеры: [1] #5.4.1-5.4.6 Самостоятельно: #5.6:9

т3.5 Специальные функции-члены: влияние на семантику класса.

п3.5 Примеры: [1] #5.5.1-5.5.5; [2] #12.1,2 Упражнения : [1] #5.6:12 Самост.: [1] #5.6:8,9

т3.6 Производные классы : основные сведения.

п3.6 Примеры: [1] #6.1, 6.2 Упражнения : [1] #6.8:1 Самостоятельно: [1] #6.8:6

т3.7 Иерархия графических фигур для VGI-графики VC++3.1

п3.7, п3.8 Упражнения : [1] #6.8:2,3,4 приложение 2 (классы Ball, Ball2)

т3.8 Управление доступом к членам классов в C++.

п3.9 Примеры: [1] #6.6; [2] #11.1-11.7, 11.3с

т3.9 Перегрузка операций: основные сведения.

п3.10 Примеры: [1] #7.1, 7.2.1-7.3.2; [2] #13.4.1-2

т3.10 Операции преобразования, присваивания и инициализации

п3.11 Примеры: [1] #7.3.3-7.6; [2] #13.4.3 Упр.: [1] #7.14:1,8,9 Самост.: [1] #7.14:10,11

т3.11 Перегрузка операций [], (), ->, ++, -- .

п3.12 Примеры: [1] #7.7-7.12; [2] #13.4.4-13.4.7

т3.12 Пример: класс, реализующий строку с отложенным копированием и набором операций с такими строками.

п3.13, п3.14 Упражнения : [1] #7.14:14,15,17 Самостоятельно: [1] #7.14:12,13,16,18-20

т3.13 Потоки ввода/вывода в C++ как иерархия классов.

п3.15, п3.16 Примеры: [1] #10.1-10.7 Упражнения : [1] #10.7:1,2 Самост.: [1] #10.7:3,5,6,13

т3.14 Обработка исключительных ситуаций (exception)

п3.17 примеры из системы помощи CBuilder/VisualStudio, Примеры: [1] #9.1,2 [2] #15.1-6

Тема 4: Обобщенное программирование: шаблоны, STL (т26-29 (4 теор. занятия) п30-32 (3 практ. занятия))

т4.1 Шаблоны функций и классов: основные сведения. Пример: шаблон класса Vector и функции sort.

п4.1 Примеры: [1] #8.4.1,8.5,8.6 [2] #14.4

т4.2 Пример: шаблон класса ассоциативного массива Map и итератора для него Mapiter.

п4.2 Упражнения : [1] #8.9:4

т4.3 STL (станд. библиотека шаблонов): основные концепции (последовательность, итератор, категории итераторов, свойства итераторов)

т4.4 STL: стандартные контейнеры и стандартные алгоритмы

п4.3: примеры из системы помощи CBuilder/VisualStudio

Тема 5: Введение в объектно-ориентированный анализ и проектирование (т30-32 (3 теор. занятия))

т5.1 Основы объектно-ориентированного анализа и проектирования: объектная модель (абстрагирование, инкапсуляция, модульность, иерархия)

т5.2 Структура классов ("is a") и структура объектов ("part of"). Нотация Буча. Отношения между классами и между объектами.

т5.3 Понятие о паттернах проектирования.

Тема 6. Практические консультационные занятия по освоению выбранной среды визуальной разработки приложений (CBuilder v.6+ или VisualStudio 2005-2013), выполнению заданий повышенной сложности и по подготовке компьютерных программ учащихся к "Ежегодной итоговой конференции Компьютерного центра" и/или городской конференции по программированию (пк2-31 вариативная часть, соответствует выбору и индивидуальным возможностям учащегося)

Примерный список заданий для учащихся, использующих CBuilder (www.firststeps.ru)

Маленький текстовый редактор.

Создание дочерней формы.

Хранилище объектов.

Апгрейдем MDI проект.

Прием файлов Drag'n'Drop.

Консоль [спецификаторы printf, функции getch, getchar].

Строковые классы [AnsiString, char*, WideString].

Списочные классы [TList, оператор dynamic_cast].

Класс списка строк TStringList и его связь с INI-файлами [TStringList, TIniFile].

Реестр Windows. Обзор.

TRegistry.

Извлечение иконки по расширению файла и ее загрузка из DLL или EXE.

Обзор WinAPI.

Функции Windows [MoveWindow, GetWindowRect].

Прокрутка области ввода [SendMessage, PostMessage, SCROLL].

WinSight.

Таблицы откликов [MESSAGE_MAP, MESSAGE_HANDLER].

Компоненты TPopupMenu, TCheckBox, TRadioButton.

Компоненты TGroupBox, TRadioGroup, TPanel.

Компоненты TListBox, TComboBox.

Компонент TActionList.

Компонент TFrame.

Дополнительные компоненты.

Компонент TBitBtn.

Компонент TSpeedButton.

Канва.

Методы и свойства канвы.

TPen.
TBrush.
Методы рисования. Линии, полигоны.
Методы рисования. Текст.
Методы и свойства канвы. То, что осталось.
Выведение внеэкранных образов.
Компонент TImage.
TShape.
Про BROWSEINFO и <shlobj.h>.
Про отладку.
TBevel.
Область прокрутки TScrollBar. Свойства Parent и Owner.
Компонент TCheckBoxList.
Разделитель TSplitter
Компонент TStaticText
Контрольная панель TControlBar. Принципы Drag-Dock.
Компонент TControlBar. Продолжение.
Файловый ввод-вывод.
Файловый ввод-вывод. Продолжение.
Класс TFileStream. Понятие потока VCL.
Пример использования TFileStream.

Тема 7. Заключительное занятие: итоги конференции и учебного года (т,п, пк 36)

Без темы: Проверочные и консультационные занятия – 3т, 3пр, 3пр(к)
(во время осенних, зимних, весенних каникул) (т,п,пк 33-35)

ОЦЕНОЧНЫЕ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

В процессе обучения используются следующие оценочные материалы:

- Анкета для учащихся – 1 раз в год
- Карта самооценки учащегося - 1 раз в год
- Карта диагностики результатов обучения при промежуточном контроле – один раз в год
- Карта оценки итоговой работы – 1 раз в год

Виды и формы контроля:

Входной контроль осуществляется путем анкетирования с последующим анализом анкет.

Итоговый контроль включает в себя итоги по второму полугодию, а также оценку итоговой работы (проекта)

В конце учебного года заполняется протокол итоговой конференции, в котором выставляются баллы за итоговую работу.

Форма протокола представлена в приложении.

Текущий контроль за учащимися осуществляется на каждом практическом занятии.

Знания учащихся оцениваются по зачетной системе по каждой учебной теме. Оценка зависит от количества и качества выполненных учащимся программ. Работа оценивается следующим образом:

Зачет: учащийся полностью освоил тему, выполнил все предложенные программы с минимальной помощью преподавателя, или учащийся освоил тему, выполнил предложенные программы с подсказками преподавателя, или : учащийся частично освоил тему, выполнил необходимый минимум программ с большим количеством подсказок преподавателя.

Незачет: учащийся посещал занятия, но тему не освоил, или учащийся не присутствовал на занятиях по теме.

Промежуточный контроль проводится по окончании первого полугодия.

Годовым итогом являются: выполненные задания по примерам с лекций; выполненные уроки по освоению визуального программирования в выбранной среде разработки; программа, представленная на итоговой конференции в конце учебного года. Итоги конференции фиксируются способом, указанным в положении об итоговой конференции компьютерного центра отдела техники.

При оценивании качества представленной программы используются следующие критерии:

- алгоритмическая сложность задачи;
- соблюдение стандартных требований к качеству программного кода;
- применение технологии объектно-ориентированного программирования;
- эффективность выбора использованных при решении стандартных компонентов выбранной среды разработки C++
- сложность и обоснованность примененного математического аппарата;
- общность использованных методов решения, возможность масштабирования решенной задачи;
- оригинальность задачи, самостоятельность постановки задачи;
- возможность практического применения разработанной программы.

В процессе реализации программы применяются современные образовательные технологии:

1. Программа предусматривает выполнение учащимися индивидуальных учебных проектов. Проектная технология является современным методом обучения,

позволяющим комплексно развивать активность обучающихся в приобретении необходимых знаний, способность творческого выбора целей, умения планировать свою работу, координировать усилия с другими участниками проекта, представлять результаты деятельности в целом. Выполнение проектов имеет ценное психологическое значение в аспекте самооценки обучающихся и осмысления ими своей деятельности и ее результатов, а также воспитательное значение, т.к. способствует развитию устойчивой самоотдачи, трудолюбия и ответственного отношения к работе в коллективе, коммуникативности, инициативности и самостоятельности.

Использование технологии проектного обучения позволяет сформировать проектное мышление, будет отработан алгоритм проектной деятельности в области компьютерных технологий.

Дидактический материал

Дидактический материал составляют: конспект лекций на базе [1-3], [5], [17], примеры программ (фрагменты кода) к практическим занятиям, базирующиеся в основном на [1,3], а также пошаговые уроки для овладения средами визуальной разработки, доступные в большом количестве в сети Интернет (например, www.firststeps.ru).

Примеры конспектов лекций вместе с заданиями на практическое занятие приведены в приложении 2.

Приложение №1

- Информационная карта освоения дополнительной образовательной программы.

Информационная карта освоения дополнительной образовательной программы.

Группа _____

Программа **Основы профессионального программирования на языке C++**

Педагог Гузенко П.Ю.

	Фамилия / тема	T1	T2	T3	T4	T5	T6	T7
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								

Знания учащихся оцениваются по зачетной системе по каждой учебной теме. Оценка зависит от количества и качества выполненных учащимся программ. Работа оценивается следующим образом:

Зачет: учащийся полностью освоил тему, выполнил все предложенные программы с минимальной помощью преподавателя, или учащийся освоил тему, выполнил

предложенные программы с подсказками преподавателя, или : учащийся частично освоил тему, выполнил необходимый минимум программ с большим количеством подсказок преподавателя.

Незачет: учащийся посещал занятия, но тему не освоил, или учащийся не присутствовал на занятиях по теме.

Тема 7 не оценивается, т.к. результатом здесь является балл, полученный на итоговой конференции или (при участии) диплом городского конкурса по программированию..

Протокол итоговой конференции

_____ 201__ года

№	Фамилия	класс	группа	название проекта	Балл
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					

Педагогическое жюри:

- 1
- 2
- 3

Работы оцениваются по 20-балльной системе

20 – достойная работа, получен диплом городского конкурса;
18 – работа хорошая, участвовала в городском конкурсе
8-14- работа сделана хорошо на уровне учреждения

1-7 – работа слабая, учащийся освоил не все темы.

Образец анкеты учащегося.

Оцените по шкале «Отлично» - «Удовлетворительно» - «Неудовлетворительно» следующие свои показатели по результатам обучения.

- 1-Уровень своих творческих способностей
- 2-Уровень своего внимания, памяти, воображения
- 3-Приобретенные навыки работы в коллективе.
4. -Осознание ценности освоения новейших компьютерных технологий для себя.
5. - Уровень своей самостоятельности, ответственности, независимости мнения

При ответах руководствоваться следующими определениями:

Отлично – в результате обучения мои умения и навыки существенно возросли, я способен и далее их наращивать. Удовлетворительно – я ощущаю рост моих умений и навыков, но он не кажется мне устойчивым, в некоторых направлениях я продвинулся незначительно. Неудовлетворительно – обучение ничего существенного мне не дало.

Приложение 2. Примеры конспектов теоретических занятий вместе с заданием на практику.

//Пр-р объектного типа (класса) -мячик (Ball) дата: 09.01.2006 Занятие 3.7

```
#include <graphics.h> // Упражнение:1) пусть по экрану бегают
#include <stdlib.h>    // ЭЛЛИПСЫ (аккуратнее с отражением от
#include <stdio.h>    // гориз. / верт. стенок
#include <conio.h>    // 2) пусть во время движения эллипс из
#include <alloc.h>    // горизонтального плавно становится
                    // вертикальным
                    // 3) попробуйте сделать нечто подобное в C Builder / Visual
class BALL {                                               // Studio
void *pb; int Xb, Yb, R; // это должно быть понятно...
int Vx, Vy; // {0,+1} задают ТОЛЬКО направление
int dx0,dy0,dx,dy ; // dx0, dy0 задают абс.величину
                    //перемещения за шаг
char ini, visi; // признаки инициализированности и видимости
public:
int GetX() {return Xb;} int GetY() {return Yb;}
BALL(int ix,int iy,int ir,int idx0,int idy0,int ivx,int ivy);
void Show(); void Hide();
void Move(int Nx, int Ny);
void Step(); // один шаг с текущими параметрами движения
}; // конец ОБЪЯВЛЕНИЯ класса BALL
```

```
// = ===== Ball realization
// все ф-ции-члены должны выполняться в граф. режиме
BALL::BALL(int ix, int iy, int ir, int idx0, int idy0,
int ivx, int ivy)
{ int tc,i; cleardevice(); tc = getbkcolor();
setcolor(LIGHTRED);
for(i=1; i<=5; i++) circle(ir, ir, (ir/5)*i);
i = imagesize(0,0,2*ir, 2*ir);
pb = malloc(i); getimage(0,0,2*ir, 2*ir, pb); setcolor(tc);
cleardevice(); ini=1; visi=0;
dx0=idx0; dy0=idy0; Vx=ivx; Vy=ivy; R=ir; Xb=ix; Yb=iy;
```

```

} // Ball::BALL

void BALL::Show(){
  if (!visi){ putimage(Xb, Yb, pb, XOR_PUT); visi=1;}
} // BALL::Show()

void BALL::Hide(){
  if (visi) { putimage(Xb, Yb, pb, XOR_PUT); visi=0;}
} // BALL::Hide()

void BALL::Move(int Nx, int Ny){
  char vi; vi=visi; Hide(); Xb=Nx; Yb=Ny; if (vi) Show();
} // Ball::Move(...)

void BALL::Step() { // шаг шара с отражениями от всех стенок
  char vi; vi=visi; if (vi) Hide();
  Xb+=Vx*dx0; Yb+=Vy*dy0;
  if ( (Xb<dx0) || ( (Xb+2*R)>(getmaxx()-dx0) ) ) Vx=-Vx;
  if ( (Yb<dy0) || ( (Yb+2*R)>(getmaxy()-dy0) ) ) Vy=-Vy;
  if (vi) Show();
} // BALL::Step()

int main(){ int gd, gm, gr, i; char ch; gd=VGA; gm=VGAHI;
initgraph(&gd, &gm, "e:\\borlandc\\bgi");
BALL a(100, 100, 25, 3, 1, -1, -1),
      b(150, 150, 40, 2, 2, -1, -1),
      c(200, 200, 20, 2, 2, 1, -1),
      d(250, 250, 15, 1, 3, 1, 1),
      e(350, 350, 10, 1, 3, 1, 1); // 5 объектов класса BALL

setcolor(LIGHTGREEN); rectangle(0,0, getmaxx(), getmaxy());
// нарисовать фоновую картинку -- сеточку
for(i=1;i<=10;i++){ line(64*i,0,64*i,getmaxy());line(0,48*i,getmaxx(),48*i);}

a.Show(); b.Show(); c.Show();// показать все шарики
d.Show(); e.Show();

// каждый шарик делает СВОЙ шаг:
do{ a.Step(); b.Step(); c.Step(); d.Step(); e.Step();
  if ( kbhit() ) ch=getch(); }
while( ch!=27);

getch(); closegraph(); } //main

```

/* Некоторые проблемы: шарики прекрасно бегают по экрану, не реагируя друг на друга. Но если надо, чтобы шарики как-то взаимодействовали, то друг с другом – сделать несложно; если же взаимодействующие объекты разных классов, то при переборе пар в полиморфной структуре данных надо уметь определять фактический тип того, на что смотрит указатель, но в Borland C++3.x нет функции, аналогичной typeid() Borland Pascal'a. Это весьма общая проблема, решенная в стандарте C++'98 введением средств RTTI */

/*Пр-р наследования классов: из класса Ball, объекты которого движутся по прямой с отражением от стенок, сделать производный класс Ball2, объекты которого движутся по кривой семейства РОЗ. дата: 09.01.2006*/

```
#include <graphics.h> #include <stdlib.h> #include <stdio.h>
```

```
#include <conio.h> #include <alloc.h> #include <math.h>
```

```
#include <dos.h> #define Pi 3.1415926535897932385
```

```
class BALL{// Xb,Yb – верхн.лев.точка описанного вокруг шарика
```

```
protected: int Xb, Yb, R; // квадрата, указывается в putimage
```

```
private: void *pb; int Vx, Vy; // {0,+1} задают направление
```

```
int dx0,dy0,dx,dy;// dx0, dy0 абс.величина перемещения за шаг
```

```
char ini, visi; // признаки инициализированности и видимости
```

```
public:
```

```
BALL(int ix,int iy,int ir,int idx0,int idy0,int ivx,int ivy);
```

```
~BALL(); virtual void Show(); virtual void Hide();
```

```
int GetX(){return Xb;} int GetY(){return Yb;} 
```

```
int isvisi(){return visi;} void Move(int Nx, int Ny);
```

```
virtual void Step(); }; // один шаг с тек. пар-рами
```

```
// Ball realization все ф-ции-члены должны выполняться в граф. режиме
```

```
BALL::BALL(int ix, int iy, int ir, int idx0, int idy0, int ivx, int ivy) { int tc,i; cleardevice(); tc =
```

```
getbkcolor(); setcolor(LIGHTRED); for(i=1;i<=5;i++) circle(ir,ir,(ir/5)*i);
```

```
i=imagesize(0,0,2*ir,2*ir); pb=malloc(i); getimage(0,0,2*ir, 2*ir, pb); setcolor(tc); cleardevice();
```

```
ini=1; visi=0;
```

```
dx0=idx0; dy0=idy0; Vx=ivx; Vy=ivy; R=ir; Xb=ix; Yb=iy; }
```

```
BALL::~BALL() {Hide(); free(pb);}
```

```
void BALL::Show(){if(!visi){putimage(Xb,Yb,pb,XOR_PUT);visi=1;}}
```

```
void BALL::Hide(){if(visi){putimage(Xb,Yb,pb,XOR_PUT);visi=0;}}
```

```
void BALL::Move(int Nx, int Ny){ char vi; vi=visi; Hide(); Xb=Nx; Yb=Ny; if (vi) Show(); } //
```

```
Ball::Move(...)
```

```
void BALL::Step() { // шаг шара с отражениями от всех стенок
```

```
char vi; vi=visi; if (vi) Hide(); Xb+=Vx*dx0; Yb+=Vy*dy0;
```

```
if ( (Xb<dx0) || ( (Xb+2*R)>(getmaxx()-dx0) ) ) Vx=-Vx;
```

```
if ( (Yb<dy0) || ( (Yb+2*R)>(getmaxy()-dy0) ) ) Vy=-Vy;
```

```
if (vi) Show(); } //Delay(0);
```

```
class BALL2: public BALL { double Ax,Ay,Bx,By, /* полярные углы:*/ qmin,qmax,q,dq, k ; int
```

```
N; /* r = sin kq, N точек на кривой от qmin до qmax */ int Xorg,Yorg, Rorg; /*центр и радиус
```

```
описанного вокруг РОЗЫ круга в граф.СК; строим отображение:
```

```
[-1,+1]x[-1,+1] --> [Xorg-Rorg;Xorg+Rorg]x[Yorg-Rorg;Yorg+Rorg]
```

```
double f();int Xg();int Yg();/*(f(q),q)->граф.(X,Y)*/ public:
```

```
BALL2(int ir,int iXorg,int iYorg,int iRorg,double ik,int iN);
```

```
~BALL2(); // см.комментарий в конце программы. Стираются ли BALL2 ?
```

```
virtual void Step(); }; // один шаг с тек. пар-рами по кривой
```

```
//Ball2 realization все ф-ции-члены должны выполняться в граф. режиме
```

```
BALL2::BALL2(int ir, int iXorg, int iYorg, int iRorg,
```

```
double ik,int iN):BALL(0,0,ir,0,0,0,0),Xorg(iXorg),Yorg(iYorg),
```

```
Rorg(iRorg),k(ik),N(iN) {Ax=Rorg;Bx=(Xorg-Rorg)+Rorg;Ay=-Rorg; By = (Yorg-Rorg)+Rorg;
```

```
qmin=0.0; qmax=3*2*Pi; q=qmin; dq=(qmax-qmin)/N; Xb=Xg(/*f(q),q*/)-ir;
```

```
Yb=Yg(/*f(q),q*/)-ir;}
```

```
double BALL2::f(){ return sin (k*q); }
```

```
int BALL2::Xg() { return (int)(Ax*f()*cos(q)+Bx +0.5); }
```

```
int BALL2::Yg() { return (int)(Ay*f()*sin(q)+By +0.5); }
```

```
void BALL2::Step() { // шаг шара по своей кривой из q в q+dq
```

```

char vi;vi=visvi(); if(vi)Hide(); q+=dq; Xb=Xg(/*f(q),q*/)-R; Yb=Yg(/*f(q), q*/)-R; if(vi)/* Show();
так со следом:*/{Show(); putpixel(Xb+R,Yb+R,LIGHTCYAN);} delay(0); if(q>qmax)q-=qmax; }
typedef BALL* pBALL; typedef BALL2* pBALL2;
int main(){int gd=VGA,gm=VGAHI,i;char ch;initgraph(&gd,&gm,"");
pBALL bls[10]; int indB=7;//индекс последнего имеющегося BALL'a
bls[1]= new BALL(100, 100, 25, 3, 1, -1, -1);
bls[3]= new BALL(300, 300, 15, 2, 2, -1, +1);
bls[5]= new BALL(200, 200, 20, 2, 2, 1, +1);
bls[7]= new BALL(400, 400, 10, 4, 1, 1, 3);
bls[0]= new BALL2(10, 100, 100, 50, 5, 2400);
bls[2]= new BALL2(20, 400, 200, 150, 8/3, 2400);
bls[4]= new BALL2(15, 250, 350, 150, 9, 7200);
bls[6]= new BALL2(5, 200, 150, 100, 8, 7200);
setcolor(LIGHTGREEN); rectangle(0,0, getmaxx(), getmaxy());
/* нарисовать фоновую картинку - сеточку*/ for(i=0;i<=10;i++){
line(64*i,0,64*i,getmaxy());line(0,48*i,getmaxx(),48*i); }
for(i=0;i<=indB;i++) bls[i]->Show(); // показать всех
do{ for(i=0;i<=indB;i++) bls[i]->Step(); //каждый шарик делает
/*СВОЙ шар*/ if(kbhit())ch=getch();} while( ch!=27); getch(); closegraph(); /* ?!УПР.:
for(i=0;i<=indB;i++) delete bls[i]; delete[] bls;*/ } //main

```

/* Деструкторы не наследуются. Если базовый класс или член имеют деструктор, а сам производный класс - нет, то создается стандартный деструктор, который вызывает деструкторы базовых классов и членов производного класса. Такие созданные деструкторы имеют спецификацию public.

Тело деструктора выполняется прежде деструкторов для объектов, являющихся членами. Деструкторы для нестатических объектов, являющихся членами, выполняются прежде, чем деструкторы для базовых классов. Деструкторы для неvirtуальных базовых классов выполняются прежде, чем деструкторы для virtуальных базовых классов. Деструкторы для неvirtуальных базовых классов выполняются в порядке, обратном их описанию в производном классе. Деструкторы virtуальных базовых классов выполняются в порядке, обратном появлению их при обходе снизу и слева-направо ациклического направленного графа базовых классов. Здесь "слева-направо" означает порядок появления имен базовых классов, который был при описании их в производном классе.

Деструкторы для элементов массива вызываются в порядке, обратном вызовам при их построении. Деструктор может быть virtуальным.

В деструкторе можно вызывать функцию-член.

Объект класса с деструктором не может быть членом объединения. */

Занятие 3.11 (фрагмент: перегрузка операции ())

Вызов функции, т.е. конструкцию **выражение(список-выражений)**, можно рассматривать как бинарную операцию, в которой **выражение** является левым операндом, а **список-выражений** -правым. Операцию вызова можно перегружать как и другие операции. В функции **operator()** список фактических параметров вычисляется и проверяется по типам согласно обычным правилам передачи параметров. Перегрузка операции вызова имеет смысл прежде всего для типов, с которыми возможна только одна операция, а также для тех типов, одна из операций над которыми имеет настолько важное значение, что все остальные в большинстве случаев можно не учитывать.

```

// класс, реализующий ассоциативный массив и итератор для него
#include <iostream.h> #include <stdlib.h> #include <string.h>

```

```

class assoc { friend class assoc_iterator;
// вложенная структура для пар (имя, значение)
struct pair { char* name; int val; };
pair* vec; int max; int free;

```

```

/* В объекте класса assoc хранится вектор из структур pair размером max. В перем. free
хранится индекс первого свободного элемента вектора. Чтобы предотвратить копирование
объектов assoc, конструктор копирования и операция присваивания описаны как private. */
assoc(const assoc&); // предотвращает копирование
assoc& operator=(const assoc&); // предотвращает копирование
public: assoc(int); int& operator[](const char*); };

```

```

class assoc_iterator { const assoc* cs; // ассоц.массив
int i; // текущий индекс
public: assoc_iterator(const assoc& s) { cs = &s; i = 0; }
pair* operator() { return (i<cs->free)? &cs->vec[i++]:0; };

```

// Конструктор выглядит так:

```

assoc::assoc(int s) { max = (s<16) ? s : 16;
free = 0; vec = new pair[max]; }

```

```

int& assoc::operator[](const char* p)

```

/* работает с множеством пар (структур pair): проводит поиск p, возвращает ссылку на целое значение из найденной пары, создает новую пару, если p не найдено */

```

{ register pair* pp;
for (pp=&vec[free-1]; vec<=pp; pp-- )
if (strcmp(p,pp->name) == 0) return pp->val;
if (free == max) { //переполнение: вектор увеличивается
pair* nvec = new pair[max*2];
for (int i=0; i<max; i++) nvec[i] = vec[i];
delete vec; vec = nvec; max = 2*max; } // увеличили
pp = &vec[free++]; pp->name = new char[strlen(p)+1];
strcpy(pp->name,p); pp->val = 0; // начальное значение = 0
return pp->val; }

```

// подсчет числа вхождений во входной поток каждого слова

```

void main()

```

```

{ const MAX = 256; // больше длины самого длинного слова

```

/* Массив assoc объекта assoc_iterator нужно инициализировать, и при каждом обращении к нему с помощью операторной функции () будет возвращаться указатель на новую пару (структура pair) из этого массива. При достижении конца массива возвращается 0: */

```

char buf[MAX]; assoc vec(512); // длина vec == 16
while (cin>>buf) vec[buf]++;
assoc_iterator next(vec); pair* p;
while ( p = next() ) // вызов p=next.operator() !
cout << p->name << ": " << p->val << "\n"; } // main

```

/* Итератор подобного вида имеет преимущество перед набором

функций, решающим ту же задачу: итератор может иметь собственные закрытые данные, в которых можно хранить информацию о ходе итерации. Обычно важно и то, что можно одновременно запустить сразу несколько итераторов одного типа. */

Операцию косвенного обращения к члену -> можно определить как унарную постфиксную операцию. Это значит, если есть класс

```

class Ptr { /* что-то еще */ X* operator->(); };

```

объекты класса Ptr могут использоваться для доступа к членам класса X также, как для этой цели используются указатели:

```

void f(Ptr p) { p->m = 7; /* (p.operator->())->m = 7 */ }

```

Преобразование объекта p в указатель p.operator->() никак не зависит от члена m, на который он указывает. Именно по этой причине operator->() является унарной постфиксной операцией.

Однако, мы не вводим новых синтаксических обозначений, так что имя члена по-прежнему должно идти после -> :

```
void g(Ptr p) { X* q1 = p->; // синтаксическая ошибка
  X* q2 = p.operator->(); // нормально
}
```

Перегрузка операции -> прежде всего используется для создания "хитрых указателей", т.е. объектов, которые помимо использования как указатели позволяют проводить некоторые операции при каждом обращении к указываемому объекту с их помощью. Например, можно определить класс ResPtr для организации доступа к объектам класса Res, хранимым на диске. Параметром конструктора ResPtr является имя, которое будет использоваться для поиска объекта на диске. При обращении к объекту с помощью функции ResPtr::operator->() он переписывается в основную память, а в конце работы деструктор ResPtr записывает измененный объект обратно на диск.

(теория 06.04.10)

Приложение 3. Элементарные задачи на знание основных понятий программирования (входной контроль)

Факт их успешного решения ни о чем не говорит, а вот неспособность решить -- говорит об очень многом...

В принципе, достаточно четкого понимания, как решать, и уверенности в возможности успешного решения.

1. Написать программу сложения двух целых чисел, введенных с клавиатуры.
Модифицировать ее так, чтобы она складывала два ВЕЩЕСТВЕННЫХ числа.

2. Написать программу, вычисляющую значение выражения
 $x^2 + xy(y-x) + \cos xy$

$$1 + \frac{(x+y)(y-x)}{x^2 + y^2}$$

при $x=1.5$ и 3

$y=2.5$ и 4 (всего 4 пары значений).

Вывод программы оформить с форматом 4:1 для x и y и 10:4 для ответа:

$x= 1.5$ $y= 2.5$ result= _____. ____

$x= 3.0$ $y= 2.5$ result= _____. ____

$x= 1.5$ $y= 4.0$ result= _____. ____

$x= 3.0$ $y= 4.0$ result= _____. ____

3. Написать программу, ведущую простой диалог:

(подчеркиванием обозначен ввод с клавиатуры, имя во второй строке вывода ДОЛЖНО быть тем же, что введено в первой, а число прожитых лет ДОЛЖНО зависеть от введенного ранее года рождения).

Здравствуйте! Как Вас зовут? >Вася

~~~~~

Добрый день, Вася! В каком году Вы родились? >1985

~~~~~

В 2001 году Вам будет 16 лет. До свидания!

4. Модифицировать программу п.3 так, чтобы первый вопрос был красного цвета ввод пользователя – зеленого второй вопрос – желтого третья строка – белого экран очищался в начале работы

1я строка диалога выводилась в 10й строке экрана

2я в 15й

3я в 20й.

5. Модифицировать последнюю программу (В ДРУГОМ ФАЙЛЕ) следующим образом: (подчеркиванием обозначен ввод с клавиатуры, имя во второй строке вывода ДОЛЖНО быть тем же, что введено в первой). Программа должна определить число полных лет, месяцев и дней от введенной даты до 1.1.2016.

Здравствуй! Как Вас зовут? >Вася

~~~~~

Добрый день, Вася!

Введите день, месяц и год Вашего рождения? >5 10 2000

~~~~~

К 1 января 2016 года Вы прожили __ лет, _ месяц и __ дня. До свидания!

6. Написать программу, вычисляющую и печатающую таблицу квадратов чисел от 2 до 20.

Примерный вид:

x x^2

=====

2 4

3 9

.....

20 400

Обязательно использование оператора цикла for. Подсказка: достаточно одной целой переменной и одного цикла.

7. Написать программу, вычисляющую и печатающую таблицу умножения:

1 | 1 2 3 4 5 6 7 8 9

=====

1 | 1 2 3 4 5 6 7 8 9

2 | 2 4 6 8 10 12 14 16 18

.....

9 | 9 18 27 36 45 54 63 72 81

Обязательно использование двух вложенных операторов цикла for. Подсказка: достаточно двух целых переменных и двух циклов.

9. Написать программу, вычисляющую и выводящую таблицу 5-ти степеней для вещественных чисел от 0.1 до 1.9 с шагом 0.2.

Примерный вывод программы:

x x^2 x^3 x^4 x^5

=====

0.1 0.01 0.001 0.0001 0.00001

0.3 0.09 0.027 0.0081 0.00243

.....

1.7.....

1.9 3.61 6.859 13.0321 24.76099

Обязательно использование оператора цикла while.

10. Написать программу ввода возраста человека в переменную age типа unsigned int с контролем диапазона вводимых значений от 1 до 120. Необходимо предусмотреть обработку трех ошибочных действий пользователя:

1) введенное число меньше 1; 2) введенное число больше 120

3) введенные символы не образуют числа.

Обязательно использование оператора цикла do ... while() .

11. Написать, набрать и запустить программу игры "Угадай число" (имя файла:ugad1.cpp).

12. Модифицировать программу п.12 так, чтобы попытки угадывания нумеровались:

Попытка #1: введите Ваше число >25

Загаданное число больше Вашего

Попытка #2: введите Ваше число >56

и т.д. Имя файла: ugad2.cpp

13. Модифицировать программу п.13 так, чтобы попытки угадывания не только нумеровались, но и выводилась информация о превышении достаточного числа попыток. Это число рассчитывается по верхней границе загадываемых чисел по формуле: $N_d = (\ln(N)/\ln(2))+1$; Имя файла: ugad3.cpp Т.е. если счетчик попыток превысил значение N_d , надо вывести сообщение

вроде "Загаданное число можно угадать за ## попыток!" (##-значение N_d)

14. Написать, запустить и отладить:

-- программу управления движением символа #1 (НЕ по инерции, БЕЗ отражений, с обработкой ЧЕТЫРЕХ клавиш со стрелками). Имя файла: face1.cpp

-- модифицированную программу п.14.1 так, чтобы в двух местах экрана были прямоугольники 3x4, при попадании в который символ переносился бы в случайное место экрана (кроме другого прямоугольника). Кроме того, символ при движении должен случайно менять свой цвет. Имя файла: face1a.cpp

15. Написать, запустить и отладить программу управления движением символа #2 (по инерции, с отражениями, с обработкой ВОСЬМИ клавиш со стрелками). Имя файла: face2.cpp

16. Модифицировать программу п.15 так, чтобы в двух местах экрана были прямоугольники 3x4, при попадании в который символ переносился бы в случайное место экрана (кроме другого прямоугольника). Кроме того, символ при движении должен случайно менять свой цвет. Имя файла: face2a.cpp

17. Написать программу управления движением символа по инерции с обработкой 8ми клавиш управления курсором и отражениями не только от границ экрана, но и от несимметрично расположенных на экране ШЕСТИ

прямоугольников размера 3x4. Имя файла: face2b.cpp

18. Написать, запустить и отладить программу, рисующую 3 ряда "рожиц" по 6 штук в ряду с помощью графических примитивов и якорной точки. "Рожица" состоит из овала лица, улыбающегося рта, носа, глаз со зрачками, двух ушей и шевелюры. Элементы "рожицы" должны быть разного цвета. Размеры подобрать так, чтобы равномерно заполнить экран. Подсказка: достаточно двух вложенных арифметических циклов. Имя файла: faces1gp.cpp

19. Написать, запустить и отладить программу, рисующую 3 ряда "рожиц" (описанных в п.18) по 6 штук в ряду с помощью getImage/putimage. Имя файла: faces1gi.cpp

20. Сравнить тексты программ п.18 и п.19.

21. Модифицировать программу п.18 так, чтобы все "рожицы" переливались. Имя файла: faces1pp.cpp

22. Модифицировать программу п.19 так, чтобы все "рожицы" переливались. Имя файла: faces1pi.cpp

23. Написать, запустить и отладить программу управления движением "рожицы" (п.18) НЕ по инерции, БЕЗ отражений, с обработкой ЧЕТЫРЕХ клавиш со стрелками). Имя файла: grface1.cpp

24. Модифицировать программу п.23 так, чтобы в ТРЕХ местах экрана были прямоугольники 30x40, при попадании в который "рожица" переносилась бы в случайное место экрана (кроме других прямоугольников). Кроме того, "рожица" при движении должна случайно менять свой цвет. Имя файла: grface1a.cpp

25. Написать, запустить и отладить две программы управления движением "рожицы" ПО инерции, С отражениями, с обработкой ВОСЬМИ клавиш БЕЗ выхода за пределы экрана: - с графическими примитивами -- имя файла grface2p.cpp - с помощью GetImage/PutImage - имя файла grface2i.cpp

26. Написать, набрать и отладить программу, в которой:

а) описывается тип массива из 9 целых чисел (имя: int9) и тип массива из 6 вещественных чисел (имя: real6)

Подсказка: есть такое слово -- typedef

б) объявляются по одной переменной обоих типов (т.е. 2 массива)

в) массив целых чисел вводится с клавиатуры, затем выводится на экран, после чего

определяются его максимальный и минимальный элементы.

г) массив вещественных чисел заполняется случайными числами в диапазоне от -40.0 до +50.0, выводится на экран, после чего определяются его максимальный и минимальный элементы.

27. Написать, набрать и отладить программу, в которой описывается тип двумерного вещественного массива 5×4 (имя: `matrix`), объявляются две переменных этого типа, заполняются случайными числами в диапазонах от -80.0 до -50.0 для первой и от -70.0 до +30.0 для второй, и выводятся на печать вместе с максимальным и минимальным значениями для каждой из них.

28. Написать программу, рисующую график функции $y=f(x)$ на отрезке $[x_{\min}; x_{\max}]$ в декартовой СК. Поиск макс. и мин. знач. ф-ции - перебором по сетке значений x . Желательно нарисовать оси координат и засечки на осях с цифрами около засечек.

ОПИСЬ УЧЕБНО-МЕТОДИЧЕСКОГО КОМПЛЕКСА
к дополнительной общеобразовательной общеразвивающей программе
«Основы профессионального программирования на языке С++»

Направленность	Техническая			
Продолжительность освоения	1 год			
Возраст детей	13 — 16 лет			
Нормативное обеспечение	<p>общеобразовательная программа Рабочая программа план мероприятий отдела техники Инструкции по технике безопасности Нормативная документация:</p> <ul style="list-style-type: none"> • Федеральный закон Российской Федерации №273-ФЗ "Об образовании в Российской Федерации" от 29.12.2012 • Концепция развития дополнительного образования детей в Российской Федерации <i>Распоряжение Правительства РФ от 04.09.2014 №1726-р</i> • Стратегия развития системы образования Санкт-Петербурга на 2011–2020 гг. «Петербургская Школа 2020» // <i>Совет по образовательной политике Комитета по образованию Правительства Санкт-Петербурга, 2010</i> • Стратегия развития воспитания в Российской Федерации на период до 2025 года // <i>Распоряжение Правительства РФ от 29.05.2015 №996-р</i> • Об утверждении СанПиН 2.4.4.3172-14 "Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательной организации дополнительного образования детей" // <i>Постановление Главного санитарного врача РФ от 04.07.2014 №41</i> • Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам // <i>Приказ Министерства образования и науки РФ от 29.08.2013 г. №1008</i> • Приказ Минздравсоцразвития РФ от 26.08.2010 N 761н (ред. от 31.05.2011) "Об утверждении Единого квалификационного справочника должностей руководителей, специалистов и служащих, раздел "Квалификационные характеристики должностей работников образования" : п/п Педагог дополнительного образования (включая старшего) • Приказ Минтруда России от 08.09.2015 N 613н "Об утверждении профессионального стандарта "Педагог дополнительного образования детей и взрослых" 			
Разделы / темы дополнительной общеобразовательной программы	Учебно-методические пособия для педагогов	Учебно-методические пособия для детей	Диагностические и контрольные материалы	Средства обучения
0. Вводное занятие. Знакомство с курсом. Инструктаж по технике	Правила внутреннего распорядка, инструкции по	-	Опрос на знание основных понятий программирования	-

безопасности и правилам дорожного движения.	ТБ.			
1. Вводное занятие по первоначальному ознакомлению с компьютером, составом программного обеспечения и средой разработки		Конспекты в рабочих тетрадях.	Опрос, контрольные задания.	Доска, персональный компьютер преподавателя, проектор, экран.
2: Углубленное повторение основных понятий программирования и изучение их реализации в языке C++	Конспекты лекций на базе [1]-[3]	Конспекты лекций, конспекты в рабочих тетрадях.	Опрос, примеры с теории с доп. заданиями	Доска, персональный компьютер преподавателя, проектор, экран.
3. объектно-ориентированное программирование в C++	Конспекты лекций на базе [1]-[3]	Фотографии конспектов лекций в лок. сети , конспекты в рабочих тетрадях	Опрос, примеры с теории с доп. заданиями	Доска, персональные компьютеры, выбранная среда программирования на C++, проектор, экран.
4. Обобщенное программирование: шаблоны, STL	Конспекты лекций на базе [1], [3]	Фотографии конспектов лекций в лок. сети , конспекты в рабочих тетрадях	Опрос, примеры с теории с доп. заданиями	Доска, персональные компьютеры, выбранная среда программирования на C++, проектор, экран.
5: Введение в объектно-ориентированный анализ и проектирование	Конспекты лекций на базе [3], [5], [17]	Конспекты лекций, конспекты в рабочих тетрадях.	Опрос	Доска, проектор, экран.
6. Практические консультационные занятия по освоению выбранной среды визуальной разработки приложений (CBuilder v.6+ или VisualStudio 2005-2013), выполнению заданий повышенной сложности и по подготовке компьютерных программ учащихся к "Ежегодной итоговой конференции Компьютерного центра" и/или городской конференции по программированию	[7],[8],[10],[11],[18] [6],[23],[24] (список для педагогов) www.programmersclub.ru, www.firststeps.com, www.progclub.ru	[3],[4],[5],[6],[9],[10] (список для учащихся) www.programmersclub.ru, www.firststeps.com, www.progclub.ru,	Опрос, предъявление выполненных заданий по изучению разработки приложений в выбранной среде разработки. Предъявление собственного проекта	персональные компьютеры, выбранная среда программирования на C++, проектор, экран..
7. Проведение итоговой конференции.	-	-	-	Доска, персональный компьютер преподавателя, проектор, экран.

СПИСОК ЛИТЕРАТУРЫ, ИСПОЛЬЗОВАННОЙ ПЕДАГОГОМ

1. Страуструп Б. Язык программирования C++(вторая редакция) Киев:"Диасофт", 1993 ч.1 - 264 стр., ч.2 - 296 стр.
2. Эллис М., Строуструп Б. Справочное руководство по языку программирования C++ с комментариями. М.:МИР, 1992 - 445 стр.
3. Страуструп Б. Язык программирования C++(специальное издание). Бином, Невский Диалект, 2006 - 1104 стр.;
4. Эккель Б. Философия C++: ч.1 Введение в стандартный C++. Питер, 2004 г. 572 стр., ч.2 Практическое программирование. Питер, 2004 г. 608 с.
5. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений. 2 изд. Бином, 1998. –560с., 3 изд. Вильямс, 2010. – 720с.
6. Visual C++ .NET для "чайников". Хаймен М., Арнсон Б. Вильямс, 2002. – 288стр.
7. В. Ермолаев, Т. Сорока C++ Builder: Книга рецептов . 2006 г. 208 стр
8. А. Я. Архангельский Приемы программирования в C++ Builder 6 и 2006 (+ CD-ROM), Бином-Пресс, 2010г. – 992с.
9. Казаков А.А., Дорошенко Е.М., Емельянова Т.Ю., Блохина Н.В., Гузенко П.Ю. Система поиска, отбора и профессионально-ориентированного обучения талантливых учащихся в области программирования. СПб:изд-во СПбГДТЮ, 1993.
10. Ст.С. Скиена, М.А. Ревилла Олимпиадные задачи по программированию. Руководство по подготовке к соревнованиям. КУДИЦ-Образ, 2005 г. - 416 стр.
11. Ф.В. Меньшиков. Олимпиадные задачи по программированию. СПб.:Питер,2006г.-314с.
12. Э. Салливан Создание команды разработчиков программного обеспечения. М.:Издательско-торговый дом «Русская Редакция» ,2002.-368 стр.
13. Александреску А. Современное проектирование на C++. Вильямс, 2002. 336с.
14. Романовский И.В. Дискретный анализ (любое издание)
15. СанПиН 2.2.2/2.4.1340-03 и 2.4.4.3172-14
16. М.И. Степанова. Как обеспечить безопасное общение детей с компьютером. //Вестник образования России, Ноябрь 21'2003, с. 44-59
17. Э.Гамма, Р.Хелм, Р.Джонсон, Дж.Влиссидес. Приемы объектно-ориентированного проектирования. Паттерны проектирования.-СПб:Питер,2001.- 368 с., 2007г. – 366с.
18. Калверт Ч., Рейсдорф К. Borland C++ Builder 5. Энциклопедия программиста ДиаСофт, 2001 939стр.
19. Программы для общеобразовательных учреждений: Информатика. 2-11 классы. – 2-е изд.М.:БИНОМ. Лаборатория знаний., 2005. – 380с.
20. Зубрилова Н.А. Дополнительное образование и воспитание детей в Санкт-Петербурге на рубеже XX-XXI вв. СПб: ГОУ «СПб ГДТЮ», 2003г. – 218с.
21. Косов Б.В. Личность: теория, диагностика и развитие. М.: Академический проект, 2000г.- 240с.
22. Горский В.А. Технология разработки авторской программы дополнительного образования детей. // Дополнительное образование – 2003 №6, с.16-17.
23. Зиборов В. В. MS Visual C++ 2010 в среде .NET. Библиотека программиста. - СПб.: Питер, 2012. - 320 с.: ил.
24. Б. Пахомов C/C++ и MS Visual C++ 2012 для начинающих. БХВ-Петербург 2013, 509с.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ ДЛЯ ОБУЧАЮЩИХСЯ

1. Страуструп Б. Язык программирования C++ (вторая редакция) Киев:"Диасофт", 1993 ч.1 - 264 стр., ч.2 - 296 стр.; (специальное издание) Бином, Невский Диалект, 2008 – 1104с.
2. Эккель Б. Философия C++: Введение в стандартный C++. Питер, 2004 г. 572 стр.
3. Visual C++ .NET для "чайников". Хаймен М., Арнсон Б. Вильямс, 2002. – 288стр.

4. В. Ермолаев, Т. Сорока С++ Builder: Книга рецептов . 2006 г. 208 стр
5. А. Я. Архангельский Приемы программирования в С++ Builder 6 и 2006 (+ CD-ROM), Бином-Пресс, 2010г. – 992с.
6. А. Я. Архангельский Компоненты С++Builder. Справочное и методическое пособие. Бином-Пресс, 2008г. – 960с.
7. Э.Гамма, Р.Хелм, Р.Джонсон, Дж.Влиссидес. Приемы объектно-ориентированного проектирования. Паттерны проектирования.-СПб:Питер,2001.- 368 с.; 2007г. – 366с.
8. Страуструп Б. Программирование: принципы и практика использования С++.: Пер. с англ. – М.:ООО “И.Д.Вильямс”, 2011. – 1248с.
9. Зиборов В. В. MS Visual С++ 2010 в среде .NET. Библиотека программиста. - СПб.: Питер, 2012. - 320 с.: ил.
10. Б. Пахомов С/С++ и MS Visual С++ 2012 для начинающих. БХВ-Петербург 2013, 509с.

ЭЛЕКТРОННЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕСУРСЫ

1. <http://www.programmersclub.ru>
2. <http://www.firststeps.com>
3. <http://www.progclub.ru>
4. <http://хоax.net/cpp/> (на англ. языке)
5. <http://kychka-pc.ru/category/sfml>

Приложение 4 . Обоснование тезиса: Наличие воспитательных задач не является основанием для исполнения педагогом дополнительного образования воспитательной трудовой функции, так как воспитательная трудовая функция по этой должности отсутствует как в Едином квалификационном справочнике должностей руководителей, специалистов и служащих, раздел "Квалификационные характеристики должностей работников образования" : Педагог дополнительного образования (включая старшего), так и в приказе Минтруда России от 08.09.2015 N 613н "Об утверждении профессионального стандарта "Педагог дополнительного образования детей и взрослых" .

Цитаты из нормативных актов, устанавливающих отсутствие трудовой функции воспитательной работы по должности педагог доп. образования.

Приказ Минздравсоцразвития РФ от 26.08.2010 N 761н (ред. от 31.05.2011) "Об утверждении Единого квалификационного справочника должностей руководителей, специалистов и служащих, раздел "Квалификационные характеристики должностей работников образования"
Педагог дополнительного образования (включая старшего)

Должностные обязанности. Осуществляет **дополнительное образование обучающихся**, воспитанников в соответствии со своей образовательной программой, развивает их разнообразную творческую деятельность. Комплекует состав обучающихся, воспитанников кружка, секции, студии, клубного и другого детского объединения и принимает меры по сохранению контингента обучающихся, воспитанников в течение срока обучения. Обеспечивает педагогически обоснованный выбор форм, средств и методов работы (обучения) исходя из психофизиологической и педагогической целесообразности, используя современные образовательные технологии, включая информационные, а также цифровые образовательные ресурсы. Проводит учебные занятия, опираясь на достижения в области методической, педагогической и психологической наук, возрастной психологии и школьной гигиены, а также современных информационных технологий. Обеспечивает соблюдение прав и свобод обучающихся, воспитанников. Участвует в разработке и реализации образовательных программ. Составляет планы и программы занятий, обеспечивает их выполнение. Выявляет творческие способности обучающихся, воспитанников, способствует их развитию, формированию устойчивых профессиональных интересов и склонностей. Организует разные виды деятельности обучающихся, воспитанников, ориентируясь на их личности, осуществляет развитие мотивации их познавательных интересов, способностей. Организует самостоятельную деятельность обучающихся, воспитанников, в том числе исследовательскую, включает в учебный процесс проблемное обучение, осуществляет связь обучения с практикой, обсуждает с обучающимися, воспитанниками актуальные события современности. Обеспечивает и анализирует достижения обучающихся, воспитанников. Оценивает эффективность обучения, учитывая овладение умениями, развитие опыта творческой деятельности, познавательного интереса, используя компьютерные технологии, в т.ч. текстовые редакторы и электронные таблицы в своей деятельности. Оказывает особую поддержку одаренным и талантливым обучающимся, воспитанникам, а также обучающимся, воспитанникам, имеющим отклонения в развитии. Организует участие обучающихся, воспитанников в массовых мероприятиях. Участвует в работе педагогических, методических советов, объединений, других формах методической работы, в работе по проведению родительских собраний, оздоровительных, воспитательных и других мероприятий, предусмотренных образовательной программой, в организации и проведении методической и консультативной помощи родителям или лицам, их заменяющим, а также педагогическим работникам в пределах своей компетенции. Обеспечивает охрану жизни и здоровья обучающихся, воспитанников во время образовательного процесса. Обеспечивает при проведении занятий соблюдение правил охраны труда и пожарной безопасности. При выполнении обязанностей старшего педагога дополнительного образования наряду с выполнением обязанностей, предусмотренных по должности педагога дополнительного образования, осуществляет координацию деятельности педагогов дополнительного

образования, других педагогических работников в проектировании развивающей образовательной среды образовательного учреждения. Оказывает методическую помощь педагогам дополнительного образования, способствует обобщению передового их педагогического опыта и повышению квалификации, развитию их творческих инициатив

НЕОБХОДИМО СРАВНИТЬ С должностью, по которой воспитательная трудовая функция предусмотрена :

Учитель

Должностные обязанности. Осуществляет обучение **и воспитание обучающихся** с учетом их психолого-физиологических особенностей и специфики преподаваемого предмета, способствует формированию общей культуры личности, социализации, осознанного выбора и освоения образовательных программ, используя разнообразные формы, приемы, методы и средства обучения, в том числе по индивидуальным учебным планам, ускоренным курсам в рамках федеральных государственных образовательных стандартов, современные образовательные технологии, включая информационные, а также цифровые образовательные ресурсы. Обоснованно выбирает программы и учебно-методическое обеспечение, включая цифровые образовательные ресурсы. Проводит учебные занятия, опираясь на достижения в области педагогической и психологической наук, возрастной психологии и школьной гигиены, а также современных информационных технологий и методик обучения. Планирует и осуществляет учебный процесс в соответствии с образовательной программой образовательного учреждения, разрабатывает рабочую программу по предмету, курсу на основе примерных основных общеобразовательных программ и обеспечивает ее выполнение, организуя и поддерживая разнообразные виды деятельности обучающихся, ориентируясь на личность обучающегося, развитие его мотивации, познавательных интересов, способностей, организует самостоятельную деятельность обучающихся, в том числе исследовательскую, реализует проблемное обучение, осуществляет связь обучения по предмету (курсу, программе) с практикой, обсуждает с обучающимися актуальные события современности. Обеспечивает достижение и подтверждение обучающимися уровней образования (образовательных цензов). Оценивает эффективность и результаты обучения обучающихся по предмету (курсу, программе), учитывая освоение знаний, овладение умениями, развитие опыта творческой деятельности, познавательного интереса обучающихся, используя компьютерные технологии, в т.ч. текстовые редакторы и электронные таблицы в своей деятельности. Соблюдает права и свободы обучающихся, поддерживает учебную дисциплину, режим посещения занятий, уважая человеческое достоинство, честь и репутацию обучающихся. Осуществляет контрольно-оценочную деятельность в образовательном процессе с использованием современных способов оценивания в условиях информационно-коммуникационных технологий (ведение электронных форм документации, в том числе электронного журнала и дневников обучающихся). Вносит предложения по совершенствованию образовательного процесса в образовательном учреждении. Участвует в деятельности педагогического и иных советов образовательного учреждения, а также в деятельности методических объединений и других формах методической работы. Обеспечивает охрану жизни и здоровья обучающихся во время образовательного процесса. Осуществляет связь с родителями (лицами, их заменяющими). Выполняет правила по охране труда и пожарной безопасности.

Приказ Минтруда России от 08 .09.2015 N 613н "Об утверждении профессионального стандарта "Педагог дополнительного образования детей и взрослых", столбец «Трудовые функции»

Организация деятельности учащихся, направленной на освоение дополнительной общеобразовательной программы А/01.6

Организация досуговой деятельности учащихся в процессе реализации дополнительной общеобразовательной программы А/02.6

Обеспечение взаимодействия с родителями (законными представителями) учащихся, осваивающих дополнительную общеобразовательную программу, при решении задач обучения и воспитания <4> А/03.6

Педагогический контроль и оценка освоения дополнительной общеобразовательной программы А/04.6

Разработка программно-методического обеспечения реализации дополнительной общеобразовательной программы А/05.6

Трудовые действия по трудовой функции А/03.6

Планирование взаимодействия с родителями (законными представителями) учащихся

Проведение родительских собраний, индивидуальных и групповых встреч (консультаций) с родителями (законными представителями) учащихся

Организация совместной деятельности детей и взрослых при проведении занятий и досуговых мероприятий

Обеспечение в рамках своих полномочий соблюдения прав ребенка и выполнения взрослыми установленных обязанностей

НЕОБХОДИМО СРАВНИТЬ С должностью, по которой воспитательная трудовая функция предусмотрена :

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)" столбец «Трудовые функции»

Общепедагогическая функция. Обучение А/01.6

Воспитательная деятельность А/02.6

Развивающая деятельность А/03.6

Трудовые действия по трудовой функции А/02.6

Регулирование поведения обучающихся для обеспечения безопасной образовательной среды

Реализация современных, в том числе интерактивных, форм и методов воспитательной работы, используя их как на занятии, так и во внеурочной деятельности

Постановка воспитательных целей, способствующих развитию обучающихся, независимо от их способностей и характера

Определение и принятие четких правил поведения обучающимися в соответствии с уставом образовательной организации и правилами внутреннего распорядка образовательной организации

Проектирование и реализация воспитательных программ

Реализация воспитательных возможностей различных видов деятельности ребенка (учебной, игровой, трудовой, спортивной, художественной и т.д.)

Проектирование ситуаций и событий, развивающих эмоционально-ценностную сферу ребенка (культуру переживаний и ценностные ориентации ребенка)

Помощь и поддержка в организации деятельности ученических органов самоуправления

Создание, поддержание уклада, атмосферы и традиций жизни образовательной организации

Развитие у обучающихся познавательной активности, самостоятельности, инициативы, творческих способностей, формирование гражданской позиции, способности к труду и жизни в условиях современного мира, формирование у обучающихся культуры здорового и безопасного образа жизни

Формирование толерантности и навыков поведения в изменяющейся поликультурной среде

Использование конструктивных воспитательных усилий родителей (законных представителей) обучающихся, помощь семье в решении вопросов воспитания ребенка

===== дополнение 23 авг и далее ПЕЧАТАТЬ НЕ НАДО

Сноска насчет трудовой функции перенесена в прил. 4 .а в тексте сокращена до двух строк

Содержание переведено в текст. По желанию добавлен один из вариантов для вариативной части – практик-консультаций. + еще 2 стр.

30/08/2017 правка файла гузенко_CPP_2017_23AUG17.doc

Электронные ресурсы продублированы в списке источников (добавлен еще один) В УМК ресурсы также имелись.

Проверены часы учебного плана – по графе ВСЕГО – 216 ч. Доп тем в этой программе не было. На каникулах – пров. И конс. Занятия по пройденному материалу

Эксперт обещала что-то пометить карандашом то ли в восп задачах, то ли в результатах – если эту часть можно исправить без меня – правьте .

Красную фразу про содержание сверять с уч планом уберите, если прочтете все это и оно устроит

Результат – файл гузенко_CPP_2017_30AUG17.doc