

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Алтайский государственный университет»

Утверждено:
на заседании Ученого совета
ФГБОУ ВО «Алтайский
государственный университет»
протокол №1 от 29.09.2020

Рабочая программа дополнительного образования
***«Прикладная математика и программирование:
Основы программирования на языке C++»***
в рамках направления
«Малая академия»
на 2020-2021 учебный год

Составитель:
Кравченко Г.В., к.п.н.,
доцент кафедры ДУ ИМИТ

Барнаул, 2020

Пояснительная записка

Актуальность: Информационно-коммуникационные технологии проникли во все сферы человеческой деятельности: медицину, проектирование зданий, машин, образование и т.п. Для каждой из этих областей разрабатываются соответствующие программы. Следовательно, сегодня является востребованной такая профессия, как программист: навыки программирования пользуются высоким спросом, должность программиста хорошо оплачивается. Даже за пределами ИТ-мира знание хотя бы одного языка программирования – это серьезный плюс в резюме.

Однако в последние годы у большинства школьников и студентов наблюдается значительное снижение интереса к математике и физике, но без достаточных знаний по этим дисциплинам невозможно стать высококлассным программистом. Следует заметить, что кружки по программированию пользуются у школьников намного большей популярностью, чем по математике или физике (если этот кружок не готовит к ЕГЭ). Количество желающих научиться программированию в разы больше, чем количество любителей решать задачи математических или физических олимпиад. В этих условиях занятия по программированию можно и нужно использовать в качестве стимула для изучения математики и физики в целом или отдельных их разделов.

Отличительная особенность данной образовательной программы:

Обучение программированию рекомендуется начинать класса с 7-8, когда у большинства школьников уже на начальном уровне сформировалось алгоритмическое и логическое мышление, развивается абстрактное мышление.

Одним из первых языков программирования, с которых можно начинать обучение, является высокоуровневый язык объектно-ориентированного программирования С, на котором строятся такие популярные языки, как: С++, С#, Java, JavaScript и Python. В предложенном курсе обучение

программированию ведется с помощью языка C++, который широко используется для разработки программного обеспечения и является одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ (Firefox, Winamp и продукты фирмы Adobe), драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр.

Основная направленность предложенного курса заключается не в рассмотрении информатики как простого продолжения математических дисциплин, отработке чисто компьютерных программистских или пользовательских навыков, а в обучении универсальным и эффективным методам работы с информацией в любой области знания и технологии, в том числе и при изучении любого школьного предмета. При этом компьютер рассматривается не как самоцель обучения, а лишь как средство усиления способностей человека к обработке информации, а также как партнер в процессе информационного обмена.

Курс рассчитан на учащихся 10-11 классов и основывается на той их особенности, что предварительная подготовка учащихся по информатике часто крайне разнородна. По этой причине преподавание программирования ведется «с нуля», в предположении, что учащийся не обладает какими-либо специальными знаниями в области программирования. Кроме того, контингент учащихся часто разнороден по своим устремлениям – если среди учащихся не было произведено специального отбора, то среди них будут не только потенциальные математики и программисты, но и компьютерные графики, и биологи, и гуманитарии. По этой причине, чтобы знания курса оказались востребованы учащимися, а также для создания комфортных условий для обучения школьников, используются следующие принципы (И.Р. Дединский):

1. Во главу угла ставится задача, ее решение и, главное, путь от задачи к решению, а не программирование (кодирование) как таковое.

2. Для записи алгоритма на языке программирования выбирается минимальное подмножество средств языка, чтобы не акцентировать внимания на кодировании и для более легкого перехода на другие языки программирования.

3. Самостоятельность решения является ключевым условием, которое необходимо доказать при сдаче работы.

4. Понимание учащимся тех средств, с помощью которых он решил задачу, ставится выше уровня самих средств решения.

5. Аккуратность и надежность решения ставятся выше «программистских трюков», иногда позволяющих в отдельных случаях добиться несколько лучших результатов.

6. Задачи ставятся в нескольких вариантах различной сложности (от базового до творческого), при сдаче работы засчитывается решение на любом уровне (но удовлетворяющее принципам 2-4). Однако уровень сложности фиксируется, в частности, для выяснения уровня профессионализма ученика (и его повышения).

7. Главным методологическим принципом является системный подход.

Важнейшей частью курса является формирование системы профессиональных ценностей (предпочтений) учащегося. В конечном счете, это формирование и есть основная инвариантная методологическая задача курса, так как все остальное – технология и будет неотвратно изменяться с течением времени. Курс излагается с опорой на развитие самостоятельного мышления учащихся, самообразование.

В курсе изучаются основы алгоритмизации и современного процедурного языка программирования C++. Упор здесь делается на технологию компьютеризации задачи – приспособление компьютера под решение задачи из некомпьютерной области. Изучение тем проводится в виде выполнения ряда проектов и практических (лабораторных) работ. В процессе разработки проекта обучающиеся коллективно обсуждают идеи решения поставленной задачи, осуществляют концептуальную проработку и

написание кода программы. В процессе обучения производится акцент на навыки устной и письменной коммуникации и командной работы. Для выполнения учащимися выбираются задания, допускающие решения в эстетически привлекательной форме, имеющие игровой компонент.

Формы занятий – лекции и практические (лабораторные) работы.

Формы контроля – опрос, сдача практических работ, защита проектов.

В качестве инструментария используется среда Microsoft Visual Studio 2019 или Code::Blocks и Bloodshed Dev-Cpp (компилятор gcc). В качестве библиотеки программной поддержки – бесплатная графическая библиотека TX Library с открытым исходным кодом (<http://ded32.ru>).

Результатом прохождения курса учащимися должны стать понимание основных принципов программирования и владение основными алгоритмическими конструкциями. Кроме того, определенная часть учащихся должна быть сориентирована на углубленное изучение программирования – в качестве специализации. Остальная часть не должна воспринять программирование как непреодолимо сложную дисциплину, что позволит им продолжить обучение в этой области и успешно решать задания с развернутым ответом из ЕГЭ по информатике, традиционно считающиеся «не для всех».

Адресат программы: школьники 10-11 классов.

Сроки реализации программы: 1 год. Режим занятий – 1 раз в неделю по 2 академических часа, наполняемость в группе – 13 учащихся.

Цель программы – формирование и развитие у обучающихся интеллектуальных и практических компетенций (освоение спектра Hard- и Soft-компетенций) в области основ программирования через кейс-технологии.

Задачи программы:

- обучающие:

- обучение основным принципам программирования и владение основными алгоритмическими конструкциями языка

программирования C++;

- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования C++;

- интеграция программ школьного курса;

- взаимное усиление информатики и других школьных предметов по принципу: технология работы с информацией – из информатики, конкретные примеры и задачи – из смежных предметов;

- формировать умение создавать проекты, в т.ч. математические, для решения поставленных задач;

- *развивающие:*

- формировать 4К-компетенции (критическое мышление, креативное мышление, коммуникация, кооперация);

- создать условия для развития коммуникативных навыков через разнообразные виды речевой деятельности (монологическая, диалогическая речь);

- создать условия для развития самостоятельности в работе и творческой активности, умения работать в группе, умения правильно обобщать данные и делать выводы;

- сформировать умение формулировать, аргументировать и отстаивать своё мнение;

- создать условия для развития таких аналитических способностей учащихся, как умение анализировать, сопоставлять, сравнивать;

- содействовать развитию умений осуществлять рефлексивную деятельность;

- содействовать формированию самостоятельной познавательной деятельности учащихся;

- способствовать развитию памяти, внимания, алгоритмического мышления;

- способствовать формированию интереса к знаниям, в т.ч. к информатике и программированию;

- *воспитательные:*

- стремиться воспитать чувство ответственности за порученное дело, исполнительности, аккуратности, добросовестности и дисциплинированности при выполнении работы;
- способствовать формированию положительной мотивации к трудовой деятельности;
- способствовать развитию умения отстаивать свою точку зрения;
- содействовать воспитанию культуры общения, потребности в самовоспитании;
- формировать чувство коллективизма и взаимопомощи; способствовать развитию культуры взаимоотношений при работе в парах, группах, коллективе;
- продолжить формирование информационной культуры учащихся.

Планируемые результаты освоения программы

Продуктовый(е) результат(ы):

- понять, как строить алгоритмы;
- понять, как создавать программы на языке C++;
- научиться писать простейшие компьютерные программы на языке C++;
- научиться создавать свои программы, уметь их модернизировать;
- научиться создавать проект по информатике в виде компьютерной программы, оформлять его и публично защищать.

Личностные результаты:

- критическое отношение к информации и избирательность её восприятия;
- осмысление мотивов своих действий при выполнении заданий;
- развитие любознательности, сообразительности при выполнении разнообразных заданий проблемного и эвристического характера;
- развитие внимательности, настойчивости, целеустремлённости, умения преодолевать трудности;

- развитие самостоятельности суждений, независимости и нестандартности мышления;

- освоение социальных норм, правил поведения, ролей и форм социальной жизни в группах и сообществах;

- формирование коммуникативной компетентности в общении и сотрудничестве с другими обучающимися.

Метапредметные результаты:

Регулятивные универсальные учебные действия:

- умение принимать и сохранять учебную задачу;
- умение планировать последовательность шагов алгоритма для достижения цели;

- умение ставить цель (создание творческой работы), планировать достижение этой цели;

- умение осуществлять итоговый и пошаговый контроль по результату;
- способность адекватно воспринимать оценку наставника и других обучающихся;

- умение различать способ и результат действия;
- умение вносить коррективы в действия в случае расхождения результата решения задачи на основе её оценки и учёта характера сделанных ошибок;

- умение в сотрудничестве ставить новые учебные задачи;
- способность проявлять познавательную инициативу в учебном сотрудничестве;

- умение осваивать способы решения проблем творческого характера в жизненных ситуациях;

- умение оценивать получающийся творческий продукт и соотносить его с изначальным замыслом, выполнять по необходимости коррекции либо продукта, либо замысла.

Познавательные универсальные учебные действия:

- умение осуществлять поиск информации в хранилищах

информационных образовательных ресурсов;

- умение использовать средства информационных и коммуникационных технологий для решения коммуникативных, познавательных и творческих задач;

- умение ориентироваться в разнообразии способов решения задач;

- умение осуществлять анализ объектов с выделением существенных и несущественных признаков;

- умение проводить сравнение, классификацию по заданным критериям;

- умение строить логические рассуждения в форме связи простых суждений об объекте;

- умение устанавливать аналогии, причинно-следственные связи;

- умение моделировать, преобразовывать объект из чувственной формы в модель, где выделены существенные характеристики объекта (пространственно-графическая или знаково-символическая);

- умение синтезировать, составлять целое из частей, в том числе самостоятельно достраивать с восполнением недостающих компонентов.

Коммуникативные универсальные учебные действия:

- умение аргументировать свою точку зрения на выбор оснований и критериев при выделении признаков, сравнении и классификации объектов;

- умение выслушивать собеседника и вести диалог;

- способность признавать возможность существования различных точек зрения и право каждого иметь свою;

- умение планировать учебное сотрудничество с наставником и другими обучающимися: определять цели, функции участников, способы взаимодействия;

- умение осуществлять постановку вопросов: инициативное сотрудничество в поиске и сборе информации;

- умение разрешать конфликты: выявление, идентификация проблемы, поиск и оценка альтернативных способов разрешения конфликта, принятие решения и его реализация;

- умение с достаточной полнотой и точностью выражать свои мысли в соответствии с задачами и условиями коммуникации;
- владение монологической и диалогической формами речи.

Предметные результаты

В результате освоения программы, обучающиеся должны

знать:

- правила безопасности и охраны труда при работе в компьютерном классе;
- основные алгоритмические конструкции;
- возможности компьютера, границы и способы его применения;
- технологию «компьютеризации» задачи;
- основные принципы и методы программирования;
- принципы реализации базовых алгоритмических конструкций и типов данных средствами языка C++;

уметь:

- применять на практике полученные знания по алгоритмизации и программированию;
- пользоваться языком блок-схем;
- разрабатывать и тестировать программы на языке C++, в т.ч. простые игры;
- работать в среде программирования Microsoft Visual Studio 2019/Code::Blocks, системедoxygen;
- оценивать условия применимости технологии;
- описывать с помощью текста, рисунков, графического изображения технологию создания проекта;
- выявлять и формулировать проблему, требующую технологического решения;
- модифицировать имеющиеся программные продукты в соответствии с ситуацией/заказом/потребностью/задачей деятельности;

- проводить оценку и испытание полученного программного продукта;
- использовать компьютер как помощь в решении задач по некомпьютерным предметам;
- подбирать оптимальный алгоритм решения задачи, модифицировать его для своих нужд;
- использовать язык программирования C++ для записи алгоритмов и в ходе работы над компьютерными проектами;
- успешно работать в групповых проектах;
- представлять свой проект;

владеть:

- научной терминологией, ключевыми понятиями, методами и приёмами в области алгоритмизации и программирования;
- основными объектно-ориентированными конструкциями языка C++;
- навыками написания программ на языке C++;
- навыками использования полученных алгоритмических и программистских знаний для решения различных задач;
- навыками работы в группе;
- навыками оформления проекта и его публичного представления.

Содержание программы

Программа предполагает постепенное расширение знаний и их углубление, а также приобретение умений в области алгоритмизации и программирования на языке C++.

Занятия предполагают развитие личности:

- развитие интеллектуального потенциала обучающегося (использование личностно-ориентированного подхода; вовлечение обучающихся в учебно-исследовательскую деятельность; развитие креативного, абстрактного, логического и самостоятельного мышления; формирование умений применять нестандартные решения; повышение у обучающихся познавательного интереса к информатике и программированию; развитие заинтересованности

обучающихся в выполнении заданий поискового характера, в выполнении ими творческих самостоятельных работ);

- развитие практических умений и навыков (умение составлять и осуществлять программу своей деятельности; умение работать в среде программирования Microsoft Visual Studio 2019 / Code::Blocks и системе doxygen; умение создавать свои собственные информационные продукты на языке C++ для решения задач из разных областей).

Учебно-воспитательный процесс направлен на формирование и развитие у обучающихся таких важных социально значимых качеств, как готовность к нравственному самоопределению, стремление к сохранению и приумножению технических, культурных и исторических ценностей. Становление личности через творческое самовыражение.

Программа дополнительного образования ориентирована на систематизацию знаний и умений по курсу информатики и ИКТ в области алгоритмизации и программирования.

Объём программы составляет 72 часа.

Содержание курса представлено в составе 4 кейсов.

Учебный план

№	Название кейса	Количество часов		
		всего	теория	практика
1.	Кейс 1. Мультфильм	22	8	14
2.	Кейс 2. Бесконечный лабиринт	14	4	10
3.	Кейс 3. Математика в программировании	16	6	10
4.	Кейс 4. Кто быстрее?	20	6	14
Итого:		72	24	48

Учебно-тематический план

№	Тема занятия	Сроки	Формы аттестации/ контроля
Кейс 1. Мультфильм (22 часа)			
1.	Введение. Алгоритмы. Перевод алгоритма в программу. Понятие о языке программирования C++	октябрь	опрос
2.	Решение задач на составление алгоритмов и их запись в различных формах. Основные возможности графической библиотеки	октябрь	опрос

3.	Набор и отладка первой программы. Линейные алгоритмы и их программирование	октябрь	практическая работа
4.	Смысловое разделение частей алгоритма. Концепция процедурного программирования. Функции без параметров	октябрь	опрос
5.	Понятие стиля программирования. Функции с параметрами	ноябрь	опрос, практическая работа
6.	Отладка программы с переменными и параметрами	ноябрь	практическая работа
7.	Повторяющиеся действия в алгоритмах. Цикл while	ноябрь	практическая работа
8.	Создание библиотеки функций. Оформление библиотеки и написание документации с помощью системы doxygen	ноябрь	практическая работа
9.	Создание библиотеки героев с документацией и обмен библиотеками между учащимися	декабрь	практическая работа
10.	Создание короткого мультфильма с использованием чужих героев. Написание рецензии на используемую библиотеку	декабрь	практическая работа
11.	Подготовка к публичному выступлению. Демонстрация результатов работы	декабрь	защита проекта
Кейс 2. Бесконечный лабиринт (14 часов)			
1.	Загрузка и использование изображений. Условные операторы и их использование	декабрь	практическая работа
2.	Задача о функции, изменяющей значения своих параметров (на примере функции движения объекта-шарика)	январь	практическая работа
3.	Передача указателей на переменные и работа с ними	январь	практическая работа
4.	Структуры: объявление, инициализация и использование	январь	опрос, практическая работа
5.	Создание необходимых для игры объектов	январь	практическая работа
6.	Написание документации к игре и выпуск версии программы	февраль	практическая работа
7.	Подготовка к публичному выступлению. Демонстрация результатов работы	февраль	защита проекта
Кейс 3. Математика в программировании (16 часов)			
1.	Построение графиков элементарных функций	февраль	практическая работа
2.	Графики функций в C++	февраль	практическая работа
3.	Возврат значений в функции. Функция как «черный ящик»	март	опрос, практическая работа
4.	Построение графиков с использованием функций возврата значений и пересчета координат	март	практическая работа
5.	Кнопочный интерфейс. Реализация кнопок с помощью структур	март	практическая работа
6.	Графики неравенств (тесты на принадлежность точки заданной области)	март	опрос, практическая работа
7.	Построение алгоритма «от определения». Вложенные циклы	апрель	практическая работа

8.	Подготовка к публичному выступлению. Демонстрация результатов работы	апрель	защита проекта
Кейс 4. Кто быстрее? (20 часов)			
1.	Хранение и обработка данных. Массивы. Алгоритмы работы с массивами (ввод, распечатка)	апрель	опрос
2.	Реализация простейших функций для работы с массивами (заполнение массива, сумма элементов, последовательный поиск элемента)	апрель	практическая работа
3.	Сортировка в массивах. Основные виды сортировки. Алгоритм сортировки выбором	май	опрос
4.	Реализация сортировки выбором с юнит-тестированием и сбором статистики эффективности (количество обменов и сравнений). Построение графиков эффективности	май	практическая работа
5.	Алгоритм сортировки пузырьком. Шейкерная сортировка	май	опрос
6.	Реализация сортировки «пузырьком» с юнит-тестированием и сбором статистики эффективности. Построение графиков эффективности	май	практическая работа
7.	Другие алгоритмы сортировки (простые вставки, бинарные вставки, подсчитывающая, слиянием, Шелла, быстрая). Сравнение эффективности алгоритмов	июнь	практическая работа
8.	Сравнение сортировок. Реализация алгоритмов с юнит-тестированием и сбором статистики эффективности	июнь	практическая работа
9.	Разработка программы с интерфейсом для построения и сравнения графиков эффективности разных алгоритмов сортировки. Написание документации	июнь	практическая работа
10.	Подготовка к публичному выступлению. Демонстрация результатов работы	июнь	защита проекта
	Итого:	72	

Тематическое содержание программы

Кейс 1. Мультфильм

Кейс позволяет обучающимся познакомиться с концепцией процедурного программирования; поработать на языке C++ с использованием циклических алгоритмов; научиться создавать библиотеки функций и писать документацию к программе; создать уникальный мультфильм.

Описание проблемной ситуации или феномена:

Представьте, что ваш любимый герой решил поменять свой имидж. Помогите ему в этом, изобразив своего героя в новом имидже с помощью

языка программирования C++. Наделите своего героя некоторыми движениями. Создайте мультфильм, используя свои и чужие персонажи.

Категория кейса: вводный.

Место кейса в структуре модуля: базовый.

Количество учебных часов/занятий, на которые рассчитан кейс: 22 часа / 11 занятий (может варьироваться в зависимости от уровня подготовки, условий и т.д.).

Этап работы	Цель	Описание	Планируемый результат продуктивный	Планируемый результат образовательный
Введение	Обоснование актуальности работы над задачей кейса	Описание способов вовлечения в проблему, погружения в ситуацию	Формулировка цели и задач деятельности; распределение ролей	Присвоение задачи кейса
Подготовительный	Создание плана-графика	Декомпозиция задач. Этапность реализации	Список задач по SMART; периоды и контрольные точки по реализации кейса	Получение навыка тайм-менеджмента; умение декомпозировать задачи
Реализационный	Генерация путей решения проблемы	Проведение анализа проблемной ситуации, генерации и обсуждение методов ее решения и возможности достижения конечного результата	Создан алгоритм для решения проблемной ситуации	Умение генерировать идеи, слушать и слышать собеседника, аргументированно отстаивать свою точку зрения; развитие критического мышления, коммуникативных навыков, навыков командной работы
	Создание героев мультфильма	Написание программной части для отображения героя мультфильма	Написана программа, рисующая героя мультфильма	Развитие алгоритмического мышления; навыки работы с языком программирования C++
	Создание программы-мультфильма	Наделение героя некоторыми движениями. Создание собственных библиотек. Создание мультфильма с помощью созданных библиотек	Написана программа, в которой герой наделен движениями; созданы собственные библиотеки	Развитие алгоритмического мышления; навыки работы с языком программирования C++
	Тестирование и доработка	Тестирование работы программы. Выявление багов и их исправление	Готовый программный продукт – мультфильм	Умение программировать на языке C++; умение проводить тестирование созданных про-

				дуктов
	Подготовка к публичному выступлению для защиты результатов. Демонстрация мультфильма в группе и защита результатов	Подготовка речи и презентации для публичной демонстрации результатов работы в кейсе. Публичная презентация результатов работы	Создана презентация для публичной демонстрации результатов	Умение анализировать результаты работы; грамотное представление результатов своей деятельности; базовые навыки ораторства, публичных выступлений, аргументирование точки зрения
Экспертный	Коммуникация с экспертным сообществом	Обсуждение результатов работы над задачей кейса. Ответы на вопросы. Рефлексия результатов. Постановка последующих целей	Получена экспертная оценка, разработан план-график дальнейшей реализации	Навыки публичных выступлений, аргументирование точки зрения

Метод работы с кейсом: продвижение по шагам жизненного цикла проекта.

Минимально необходимый уровень входных компетенций:

- работа с компьютером на уровне пользователя;
- знание математики на уровне 8 класса.

Предполагаемые образовательные результаты учащихся:

Артефакты: готовый продукт – мультфильм.

Формируемые навыки (softskills):

- развитие аналитического и алгоритмического мышления;
- понятие алгоритма, последовательного выполнения действий;
- понятие программы, подпрограммы, цикла, условия;
- умение аргументировать свою точку зрения и отстаивать ее.

Формируемые навыки (hardskills):

- умение создавать программы в C++;
- умение создавать презентации.

Процедуры и формы выявления образовательного результата:

Представление результатов образовательной деятельности пройдет в форме публичной презентации решений кейса командами и последующих ответов выступающих на вопросы наставника и других команд.

Педагогический сценарий (руководство для наставника):

Кейс представляет собой разработку приложения для решения проблемной ситуации. Кейс включает в себя:

- введение в проблему при помощи беседы с обучающимися (приведение жизненных примеров);
- групповое обсуждение проблемы, поиск путей решения, введение в группу мысли о решении проблемы через разработку приложения;
- введение в программирование вC++;
- составление технического задания на разработку приложения;
- проведение тестирования разработанного приложения и его доработка;
- подготовка к публичной презентации и защите проекта; защита проекта с демонстрацией приложения;
- подведение итогов, рефлексия.

Кейс 2. Бесконечный лабиринт

Кейс позволяет обучающимся поработать на языке C++ с условными операторами, функциями, указателями, структурами; создать уникальный дизайн будущего проекта.

Описание проблемной ситуации или феномена:

Известно, что передвигаться можно только по проложенным дорогам (автомобильные, железнодорожные, авиационные и т.п.). Проведите объект (автомобиль, поезд, самолет и т.п.) из пункта отправления в пункт назначения, написав для этого программу на языке C++.

Категория кейса: вводный.

Место кейса в структуре модуля: базовый. Рекомендуется изучать после кейса «Мультфильм».

Количество учебных часов/занятий, на которые рассчитан кейс: 14 часов / 7 занятий (может варьироваться в зависимости от уровня подготовки, условий и т.д.).

Этап работы	Цель	Описание	Планируемый результат	Планируемый результат
-------------	------	----------	-----------------------	-----------------------

			продуктовый	образовательный
Введение	Обоснование актуальности работы над задачей кейса	Описание способов вовлечения в проблему, погружения в ситуацию	Формулировка цели и задач деятельности; распределение ролей	Присвоение задачи кейса
Подготовительный	Создание плана-графика	Декомпозиция задач. Этапность реализации	Список задач по SMART; периоды и контрольные точки по реализации кейса	Получение навыка тайм-менеджмента; умение декомпозировать задачи
Реализационный	Генерация путей решения проблемы	Проведение анализа проблемной ситуации, генерации и обсуждение методов ее решения и возможности достижения конечного результата	Создан основной алгоритм	умение генерировать идеи, слушать и слышать собеседника, аргументированно отстаивать свою точку зрения; развитие критического мышления, коммуникативных навыков, навыков командной работы.
	Изучение дизайна и механики игры	Знакомство с игровыми механиками игры, изучение ограничений и правил	Модернизирован алгоритм: проверка – не победил ли игрок; обработка ошибочных предположений; проверка – не проиграл ли игрок	работа с переменными величинами; умение программировать на языке C++
	Создание простой игры	Написание документации к игре и выпуск версии программы	Написана документация к игре, выпуск версии программы	Развитие алгоритмического мышления; навыки работы с языком программирования C++
	Тестирование и доработка	Тестирование созданной игры. Проверка на граничные условия. Выявление багов и их исправление	Создание готовой игры, позволяющей передвигаться по дороге	Умение программировать на языке C++; умение проводить тестирование созданных продуктов
	Подготовка к публичному выступлению для защиты результатов. Демонстрация игры в группе и защита результатов	Подготовка речи и презентации для публичной демонстрации результатов работы в кейсе. Публичная презентация результатов работы	Создана презентация для публичной демонстрации результатов	Умение анализировать результаты работы; грамотное представление результатов своей деятельности; базовые навыки ораторства, публичных выступлений, аргументирование точки зрения
Экспертный	Коммуникация с экспертным сообществом	Обсуждение результатов работы над задачей кейса. Ответы на вопросы. Рефлексия результа-	Получена экспертная оценка, разработан план-график дальней-	Навыки публичных выступлений, аргументирование точки зрения

		тов. Постановка последующих целей	шей реализации	
--	--	-----------------------------------	----------------	--

Метод работы с кейсом: продвижение по шагам жизненного цикла игры.

Минимально необходимый уровень входных компетенций: рекомендуется к выполнению после изучения основ языка программирования C++, основных алгоритмических понятий и конструкций.

Предполагаемые образовательные результаты учащихся:

Артефакты: готовая игра.

Формируемые навыки (softskills):

- развитие аналитического и алгоритмического мышления;
- умение аргументировать свою точку зрения и отстаивать ее.

Формируемые навыки (hardskills):

- понятие алгоритма, последовательного выполнения действий;
- понятие программы, подпрограммы, цикла, условия;
- умение создавать программы и игры в C++;
- умение создавать презентации.

Процедуры и формы выявления образовательного результата:

Представление результатов образовательной деятельности пройдет в форме публичной презентации решений кейса командами и последующих ответов выступающих на вопросы наставника и других команд.

Педагогический сценарий (руководство для наставника):

Кейс представляет собой разработку приложения для решения проблемной ситуации. Кейс включает в себя:

- введение в проблему при помощи беседы с обучающимися (приведение жизненных примеров);
- групповое обсуждение проблемы, поиск путей решения, введение в группу мысли о решении проблемы через разработку игрового приложения;
- введение в программирование игровых приложений в C++;

- составление технического задания на разработку игрового приложения;
- создание уровней, программирование механики игры;
- проведение тестирования разработанного приложения и его доработка;
- подготовка к публичной презентации и защите проекта; защита проекта с демонстрацией игры;
- подведение итогов, рефлексия.

Кейс 3. Математика в программировании

Кейс позволяет обучающимся поработать на языке C++ с функциями и структурами; научиться пересчитывать координаты точек; строить графики функций; создать уникальный дизайн будущего проекта.

Описание проблемной ситуации или феномена:

Реализуйте проект «Построение графиков функций» с использованием кнопочного интерфейса на языке программирования C++.

Категория кейса: вводный.

Место кейса в структуре модуля: базовый. Рекомендуется изучать после кейса «Бесконечный лабиринт».

Количество учебных часов/занятий, на которые рассчитан кейс: 16 часов / 8 занятий (может варьироваться в зависимости от уровня подготовки, условий и т.д.).

Этап работы	Цель	Описание	Планируемый результат продуктивный	Планируемый результат образовательный
Введение	Обоснование актуальности работы над задачей кейса	Описание способов вовлечения в проблему, погружения в ситуацию	Формулировка цели и задач деятельности; распределение ролей	Присвоение задачи кейса
Подготовительный	Создание плана-графика	Декомпозиция задач. Этапность реализации	Список задач по SMART; периоды и контрольные точки по реализации кейса	Получение навыка тайм-менеджмента; умение декомпозировать задачи
Реализационный	Генерация путей решения проблемы	Проведение анализа проблемной ситуации, генерации и обсуждение	Создан алгоритм построения графиков линейных,	Умение генерировать идеи, слушать и слышать собесед-

		методов ее решения и возможности достижения конечного результата	квадратичных, степенных, показательных, логарифмических, экспоненциальных функций	ника, аргументированно отстаивать свою точку зрения; развитие критического мышления, коммуникативных навыков, навыков командной работы.
	Создание программы для построения графиков функций	Написание программы для построения графиков. Разбор элементов графика, управление цветом, опциями и свойствами элементов	Написана программа, строящая графики некоторых функций	Развитие алгоритмического мышления; умение работать на языке программирования С++
	Разработка графического интерфейса	Создание кнопок, с помощью которых разрабатываем графический интерфейс	Реализован графический интерфейс для приложения	Развитие алгоритмического мышления; навыки работы с языком программирования С++
	Тестирование и доработка	Тестирование работы программы. Выявление багов и их исправление	Готовое приложение для построения графиков некоторых функций	Умение программировать на языке С++; умение проводить тестирование созданных продуктов
	Подготовка к публичному выступлению для защиты результатов. Демонстрация программы в группе и защита результатов	Подготовка речи и презентации для публичной демонстрации результатов работы в кейсе. Публичная презентация результатов работы.	Создана презентация для публичной демонстрации результатов работы программы	Умение анализировать результаты работы; грамотное представление результатов своей деятельности; базовые навыки ораторства, публичных выступлений, аргументирование точки зрения
Экспертный	Коммуникация с экспертным сообществом	Обсуждение результатов работы над задачей кейса. Ответы на вопросы. Рефлексия результатов. Постановка последующих целей	Получена экспертная оценка, разработан план-график дальнейшей реализации	Навыки публичных выступлений, аргументирование точки зрения

Метод работы с кейсом: продвижение по шагам жизненного цикла проекта.

Минимально необходимый уровень входных компетенций: рекомендуется к выполнению после изучения основ языка программирования С++, основных алгоритмических понятий и конструкций; знание математики на уровне 9 класса.

Предполагаемые образовательные результаты учащихся:

Артефакты: готовый продукт – программа с кнопочным интерфейсом, строящая графики некоторых функций.

Формируемые навыки (softskills):

- развитие аналитического и алгоритмического мышления;
- понятие алгоритма, последовательного выполнения действий;
- понятие программы, подпрограммы, цикла, условия;
- умение аргументировать свою точку зрения и отстаивать ее;
- умение создавать презентации.

Формируемые навыки (hardskills):

- умение создавать программы вC++;
- умение создавать презентации.

Процедуры и формы выявления образовательного результата:

Представление результатов образовательной деятельности пройдет в форме публичной презентации решений кейса командами и последующих ответов выступающих на вопросы наставника и других команд.

Педагогический сценарий (руководство для наставника):

Кейс представляет собой разработку приложения для решения проблемной ситуации. Кейс включает в себя:

- введение в проблему при помощи беседы с обучающимися (приведение жизненных примеров);
- групповое обсуждение проблемы, поиск путей решения, введение в группу мысли о решении проблемы через разработку приложения;
- составление технического задания на разработку приложения;
- проведение тестирования разработанного приложения и его доработка;
- подготовка к публичной презентации и защите проекта; защита проекта с демонстрацией приложения;
- подведение итогов, рефлексия.

Кейс 4. Кто быстрее?

Кейс позволяет обучающимся поработать на языке C++ с массивами; изучить алгоритмы сортировки (выбором, пузырьком, простые вставки, бинарные вставки, подсчитывающая, слиянием, Шелла, быстрая); научиться юнит-тестированию; сравнить эффективность алгоритмов; создать уникальный дизайн будущего проекта.

Описание проблемной ситуации или феномена:

На пульт дежурного поступили данные о пробках на дорогах в городе Барнауле. По этим данным определите самый быстрый путь передвижения по городу, используя разные алгоритмы сортировки массивов. Докажите оптимальность построенного маршрута, разработав программу с графическим интерфейсом для построения и сравнения графиков эффективности разных алгоритмов сортировки.

Категория кейса: вводный.

Место кейса в структуре модуля: базовый. Рекомендуется изучать после кейса «Математика в программировании».

Количество учебных часов/занятий, на которые рассчитан кейс: 20 часов / 10 занятий (может варьироваться в зависимости от уровня подготовки, условий и т.д.).

Этап работы	Цель	Описание	Планируемый результат продуктивный	Планируемый результат образовательный
Введение	Обоснование актуальности работы над задачей кейса	Описание способов вовлечения в проблему, погружения в ситуацию	Формулировка цели и задач деятельности; распределение ролей	Присвоение задачи кейса
Подготовительный	Создание плана-графика	Декомпозиция задач. Этапность реализации	список задач по SMART; периоды и контрольные точки по реализации кейса	Получение навыка тайм-менеджмента; умение декомпозировать задачи
Реализационный	Генерация путей решения проблемы	Проведение анализа проблемной ситуации, генерации и обсуждение методов ее решения и возможности достижения конечного результата	Создан алгоритм решения проблемной ситуации	умение генерировать идеи, слушать и слышать собеседника, аргументированно отстаивать свою точку зрения; развитие критического мышления, коммуникативных навыков, навыков

				командной работы
	Создание программы для сортировки массива	Реализация алгоритмов сортировки (выбором, пузырьком, простые вставки, бинарные вставки, подсчитывающая, слиянием, Шелла, быстрая) с юнит-тестированием и сбором статистики эффективности. Сравнение эффективности алгоритмов	Написана программа, сортирующая элементы массива разными алгоритмами	Развитие алгоритмического мышления; умение работать на языке программирования С++
	Разработка графического интерфейса	Создание кнопок, с помощью которых разрабатываем графический интерфейс	Реализован графический интерфейс для приложения. Построение графиков эффективности	Развитие алгоритмического мышления; навыки работы с языком программирования С++
	Тестирование и доработка	Тестирование работы программы. Выявление багов и их исправление	Разработана программа с графическим интерфейсом для построения и сравнения графиков эффективности разных алгоритмов сортировки	Умение программировать на языке С++; умение проводить тестирование созданных продуктов
	Подготовка к публичному выступлению для защиты результатов. Демонстрация программы в группе и защита результатов	Подготовка речи и презентации для публичной демонстрации результатов работы в кейсе. Публичная презентация результатов работы	Создана презентация для публичной демонстрации результатов работы программы	Умение анализировать результаты работы; грамотное представление результатов своей деятельности; базовые навыки ораторства, публичных выступлений, аргументирование точки зрения
Экспертный	Коммуникация с экспертным сообществом	Обсуждение результатов работы над задачей кейса. Ответы на вопросы. Рефлексия результатов. Постановка последующих целей	Получена экспертная оценка, разработан план-график дальнейшей реализации	Навыки публичных выступлений, аргументирование точки зрения

Метод работы с кейсом: продвижение по шагам жизненного цикла проекта.

Минимально необходимый уровень входных компетенций: рекомендуется к выполнению после изучения основ языка программирования С++, основных алгоритмических понятий и конструкций; знание математики на уровне 9 класса.

Предполагаемые образовательные результаты учащихся:

Артефакты: готовый продукт – программа с кнопочным интерфейсом, строящая графики некоторых функций.

Формируемые навыки (softskills):

- развитие аналитического и алгоритмического мышления;
- понятие алгоритма, последовательного выполнения действий;
- понятие программы, подпрограммы, цикла, условия;
- умение аргументировать свою точку зрения и отстаивать ее;
- умение создавать презентации.

Формируемые навыки (hardskills):

- умение создавать программы вC++;
- умение создавать презентации.

Процедуры и формы выявления образовательного результата:

Представление результатов образовательной деятельности пройдет в форме публичной презентации решений кейса командами и последующих ответов выступающих на вопросы наставника и других команд.

Педагогический сценарий (руководство для наставника):

Кейс представляет собой разработку приложения для решения проблемной ситуации. Кейс включает в себя:

- введение в проблему при помощи беседы с обучающимися (приведение жизненных примеров);
- групповое обсуждение проблемы, поиск путей решения, введение в группу мысли о решении проблемы через разработку приложения;
- составление технического задания на разработку приложения;
- проведение тестирования разработанного приложения и его доработка;
- подготовка к публичной презентации и защите проекта; защита проекта с демонстрацией приложения;
- подведение итогов, рефлексия.

Формы подведения итогов реализации программы

Подведение итогов реализуется в рамках презентации и защиты результатов выполнения кейсов, представленных в программе.

Формы демонстрации результатов обучения

Представление результатов образовательной деятельности пройдет в форме публичной презентации решений кейсов командами и последующих ответов выступающих на вопросы наставника и других команд.

Методическое обеспечение программы

Приемы и методы организации образовательного процесса с отражением условий его реализации:

- структуризация учебного материала: визуализация мыслительного процесса; построение структурно-логических схем; построение таблиц; отбор ключевых понятий;

- применение дидактических средств: наглядности, литературных источников, дидактических материалов, таблиц, схем и др.;

- организационные: привлечение внимания; постановка задач; объяснение затруднений; оценивание; организация самостоятельной работы;

- активизация учебно-познавательной деятельности: проблемный вопрос; переадресация вопроса; занимательное задание; организация обсуждения; организация игровых моментов; создание ситуаций успеха;

- метод картирования мышления помогает учащемуся справиться с информационным потоком, управлять им и структурировать его. С помощью картирования можно: объединить информацию; отобразить взаимосвязи; визуализировать мысли. Ментальная карта наглядно отражает ассоциативные связи, возникающие у обучающегося. Используя этот метод, обучаемый избавляется от страха забыть или потерять какие-то сведения, учиться ориентироваться в большом потоке информации. С помощью картирования создаются интеллект-карты, помогающие при выполнении всех

организационных и структурирующих задач. Этот метод требует точности и ясности мышления и позволяет справиться со сложными задачами;

- итерационный метод обучения можно применить как при изложении лекционного материала, так и в процессе выполнения лабораторных работ по информатике. Тем более что специфика заданий, предназначенных для выполнения на лабораторном практикуме, вполне соответствует поступательному итерационному процессу, который выражается в построении ряда алгоритмов и программ решения задачи, причем каждый следующий алгоритм является уточнением или расширением предыдущего;

- метод системной динамики заключается в создании в сознании обучающегося интуитивных картин поведения объектов или систем реального мира. Применение данного метода способствуют формированию ментальных схем, отражающих алгоритмические действия;

- метод динамической визуализации информации и знаний включает в себя: комплекс учебной информации; визуальные способы ее предъявления; визуально-технические средства передачи информации; набор психологических приемов использования и развития визуального мышления в процессе обучения;

- метод поэтапного формирования умственных действий способствует формированию у обучающихся знаний по программированию и навыков алгоритмического мышления;

- проектно-исследовательский метод заключается в достижении дидактической цели через детальную разработку реальной проблемы, которая должна завершиться определенным практическим результатом, оформленным в виде проекта с использованием информационных технологий;

- использование ИКТ.

Характеристика учебно-методического комплекса и технического оснащения:

Рабочее место обучающегося: ноутбук: производительность процессора (по тесту PassMark– CPU BenchMark <http://www.cpubenchmark.net/>): не менее 2000 единиц; объём оперативной памяти: не менее 4 Гб; объём накопителя SSD/eMMC: не менее 128 Гб (или соответствующий по характеристикам персональный компьютер с монитором, клавиатурой и колонками); мышь.

Рабочее место наставника: ноутбук: процессор IntelCore i5-4590/AMD FX 8350 – аналогичная или более новая модель, графический процессор NVIDIA GeForce GTX 970, AMD Radeon R9 290 – аналогичная или более новая модель, объём оперативной памяти: не менее 4 Гб, видеовыход HDMI 1.4, DisplayPort 1.2 или более новая модель (или соответствующий по характеристикам персональный компьютер с монитором, клавиатурой и колонками)и выходом в Интернет.

Программное обеспечение:

- офисное программное обеспечение;
- графический редактор;
- среда программирования MicrosoftVisualStudio 2019/ Code::Blocks и BloodshedDev-Cpp (компилятор gcc);
- графическая библиотека TX Library с открытым исходным кодом (<http://ded32.ru>);
- системadoxygen.

Оборудование:

- интерактивная доска;
- презентационное оборудование с возможностью подключения к компьютеру;
- МФУ;
- маркерная доска с соответствующим набором письменных принадлежностей.

Список литературы

Список рекомендованной литературы для преподавателя:

Основная:

1. Ашарина И.В. Основы программирования на языках С и С++: курс лекций. – М.: Горячая линия – Телеком, 2015.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++; Пер. с англ. – М.: Бинум, 2017.
3. Павловская Т.А. С/С++. Программирование на языке высокого уровня. – СПб.: Питер, 2003.
4. Подбельский В.В. Язык С++: учебное пособие. – М.: Финансы и статистика, 2009.

Дополнительная:

1. Вайнер Р., Пинсон Л. С++ изнутри; Пер. с англ. – К.: НПИФ «ДиаСофт», 2000.
2. Дагене В.А., Григас Г.К., Аугутис К.Ф. 100 задач по программированию. – М., Просвещение, 1993.
3. Романов Е.Л. Язык Си++ в задачах, вопросах и ответах. Серия: Учебники НГТУ. – Новосибирск, НГТУ, 2003.
4. Тихомирова Л.Ф., Басов А.В.. Развитие логического мышления детей. – М., 1997.
5. Шень А.Х. Программирование: теоремы и задачи. – М.: МЦНМО, 2017.
6. Шилдт Г. С++ Базовый курс; Пер. с англ. – М.: ООО «ИД Вильямс», 2010.

Список рекомендованной литературы для обучающихся:

Основная:

1. Березин Б.И., Березин С.Б. Начальный курс С и С++. М.: Диалог-МИФИ, 2012.
2. Доусон М. Изучаем С++ через программирование игр. – СПб.: Питер, 2019.
3. Керниган Б., Ритчи Д. Язык программирования С; Пер. с англ. – М.:

ООО «ИД Вильямс», 2020.

4. Прата С. Язык программирования C++. Лекции и упражнения; Пер. с англ. – М.: ООО «ИД Вильямс», 2019.
5. Страуструп Б. Язык программирования C++. Специальное издание. – М.: Бином, 2017.

Дополнительная:

1. Динман М. C++. Освой на примерах. – СПб.: БХВ-Петербург, 2006.
2. Лаптев В. C++. Экспресс-курс. – СПб.: БХВ-Петербург, 2004.
3. Либерти Д. Освой самостоятельно C++ за 21 день. – М.: ООО «ИД Вильямс», 2003.
4. Нейбауэр А. Моя первая программа на C/C++. – СПб.: Питер, 2002.
5. Оверленд Б. C++ без страха. – М.: Триумф, 2005.
6. Уэйт У., Мартин Дж., Прата Л. Язык Си для начинающих. – М.: Мир, 1988.