МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«ИВАНОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Шуйский филиал ИвГУ

Кафедра математики, информатики и методики обучения

ОДОБРЕНО:

Руководитель ОП

_____ С.В. Белов
(подпись)

«21» февраля 2023 г.

Рабочая программа дисциплины

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

Уровень высшего образования:	бакалавриат
Квалификация выпускника:	бакалавр
Направление подготовки:	44.03.01 Педагогическое образование
Направленность (профиль) образовательной программы:	Информационные технологии и робототехника

Шуя

1. Цели освоения дисциплины

содействие профессиональному становлению педагога и развитию его компетенции в области проектной деятельности по информационным технология и робототехники через освоение навыков алгоритмизации, формирование у студентов умения разрабатывать алгоритмы и реализовывать их на компьютере, алгоритмически подходить к решению информационных задач.

2. Место дисциплины в структуре ОП

Дисциплина «Основы алгоритмизации и программирования» является обязательной для изучения; относится к обязательной части блока дисциплин Б1, входит в предметно-методический модуль.

Успешное освоение данной дисциплины будет способствовать готовности обучающихся к освоению дисциплин: «Информационные технологии в образовании»; «Образовательная робототехника»; «Программирование»; «Теория и методика обучения информационным технологиям»; «Базы данных»; «Системное программирование»; «Прикладное программирование»; «Основы искусственного интеллекта»; «Логическое программирование»; «Теория и методика обучения робототехнике»; успешному прохождению следующих практик: «Учебная практика, технологическая (проектно-технологическая) практика, по программированию (Предметно-методический модуль)», «Учебная практика, технологическая (проектно-технологическая) практика, по информационным технологиям и робототехнике (Предметно-методический модуль)», подготовке к сдаче и сдаче государственного экзамена, выполнению и защите выпускной квалификационной работы.

Студент, приступающий к изучению данной дисциплины, должен на уровне среднего общего образования:

Знать:

– основы современных технологий сбора, обработки и представления информации;

Уметь:

– получать, обрабатывать и анализировать информацию из различных источников;

– создавать и редактировать простейшие графические изображения;

Владеть:

– навыками сопоставления разных источников информации с целью выявления их противоречий и поиска достоверных суждений;

– технологиями анализа и синтеза информации на основе системного подхода.

3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

3.1. Компетенции, формированию которых способствует данная дисциплина

При реализации дисциплины формируются следующие компетенции в соответствии с ФГОС ВО по данному направлению подготовки:

а) универсальные компетенции (УК) и индикаторы их достижения:

Наименование категории (группы) универсальных компетенций	Код и наименование универсальной компетенции выпускника	Код и наименование индикатора достижения универсальной компетенции
Системное и критическое мышление	УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения	УК-1.1. Демонстрирует знание особенностей системного и критического мышления и готовность к нему. УК-1.2. Применяет логические формы и

	поставленных задач	процедуры, способен к рефлексии по поводу собственной и чужой мыслительной деятельности. УК-1.3. Анализирует источник информации с точки зрения временных и пространственных условий его возникновения. УК-1.4. Анализирует ранее сложившиеся в науке оценки информации. УК-1.5. Сопоставляет разные источники информации с целью выявления их противоречий и поиска достоверных суждений. УК-1.6. Аргументировано формирует собственное суждение и оценку информации, принимает обоснованное решение. УК-1.7. Определяет практические последствия предложенного решения задачи.
Разработка и реализация проектов	УК-2. Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1. Определяет совокупность взаимосвязанных задач, обеспечивающих достижение поставленной цели, исходя из действующих правовых норм. УК-2.2. Определяет ресурсное обеспечение для достижения поставленной цели. УК-2.3. Оценивает вероятные риски и ограничения в решении поставленных задач. УК-2.4. Определяет ожидаемые результаты решения поставленных задач.

б) общепрофессиональные компетенции (ОПК) и индикаторы их достижения:

Наименование категории (группы) общепрофессиональных компетенций	Код и наименование общепрофессиональной компетенции выпускника	Код и наименование индикатора достижения общепрофессиональной компетенции
Научные основы педагогической деятельности	ОПК-8. Способен осуществлять педагогическую деятельность на основе специальных научных знаний	ОПК-8.1. Применяет методы анализа педагогической ситуации, профессиональной рефлексии на основе специальных научных знаний. ОПК-8.2. Проектирует и осуществляет учебно-воспитательный процесс с опорой на знания основных закономерностей возрастного развития когнитивной и личностной сфер обучающихся, научно-обоснованных закономерностей организации образовательного процесса, научных основ преподаваемых предметов, их истории и места в мировой культуре и науке. ОПК-8.3. Осуществляет мониторинг и отбор программ профессионального и личностного развития.

3.2. Планируемые результаты обучения по дисциплине, соотнесенные с индикаторами достижения формируемых компетенций

В результате освоения дисциплины обучающийся должен:

Знать:

– логические формы и процедуры (УК-1 (индикаторы – 1, 2, 3, 4, 5, 6, 7);

– особенности проектного мышления (УК-2 (индикаторы – 1, 2, 3, 4);

– основные этапы проектирования, их последовательность и взаимосвязь (УК-2 (индикаторы – 1, 2, 3, 4);

– научные основы преподаваемого предмета «Информационные технологии», его историю и место в мировой культуре и науке (ОПК-8 (индикаторы – 1, 2, 3);

– этапы развития предметной области алгоритмизации и программирования (ОПК-8 (индикаторы – 1, 2, 3);

– содержательные, методологические и мировоззренческие связи предметной области со смежными научными областями (ОПК-8 (индикаторы – 1, 2, 3);

Уметь:

– аргументированно формировать собственное суждение и оценку информации (УК-1 (индикаторы – 1, 2, 3, 4, 5, 6, 7);

– осуществлять корректный подбор методов анализа, проводить обработку данных исследования и правильную интерпретацию результатов (УК-1 (индикаторы – 1, 2, 3, 4, 5, 6, 7);

– выделять в поставленной цели основные смысловые и структурные компоненты (УК-2 (индикаторы – 1, 2, 3, 4);

– формулировать задачи на основе этапов получения промежуточных результатов (УК-2 (индикаторы – 1, 2, 3, 4);

– оценивать уровень и качество каждого ресурса, обеспечивающего выполнение определенной задачи (УК-2 (индикаторы – 1, 2, 3, 4);

– руководствоваться в профессиональной деятельности научными основами преподаваемого предмета «Информационные технологии» (ОПК-8 (индикаторы – 1, 2, 3);

– соотносить основные этапы развития предметной области с ее актуальными задачами, методами и концептуальными подходами, тенденциями и перспективами ее современного развития (ОПК-8 (индикаторы – 1, 2, 3);

Владеть:

– способами решения конкретных задач проекта на уровне заявленного качества и за установленное время (УК-2 (индикаторы – 1, 2, 3, 4);

– навыками публичного представления результатов решения конкретной задачи проекта (УК-2 (индикаторы – 1, 2, 3, 4);

– научными основами преподаваемого предмета «Информационные технологии» (ОПК-8 (индикаторы – 1, 2, 3);

– теоретическими и практическими знаниями для постановки и решения исследовательских задач в предметной области и в области образования (ОПК-8 (индикаторы – 1, 2, 3).

4. Объём и содержание дисциплины

Очная форма обучения

Объём дисциплины составляет 3 зачётных единиц (81 час.)

Контактная работа 37,83 час.

Самостоятельная работа 43,17 час.

Виды учебных занятий и формы промежуточной аттестации	1 сем.	
	Контактная работа с преподавателем	Самостоятельная работа обучающегося
Лекции	12	18
Практические занятия	24	
Экзамен	0,33	25,17
Групповые консультации перед экзаменом	1,5	
Всего	37,83	43,17

4.1. Содержание дисциплины по разделам (темам), соотнесенное с видами и объемом учебных занятий

Очная форма обучения

№ п/п	Раздел (тема) дисциплины	Семестр	Виды учебных занятий, включая самостоятельную работу обучающихся и объем (в часах)				Формы текущего контроля успеваемости Формы промежуточной аттестации
			Занятия лекционного типа	Занятия семинарского типа		Самостоятельная работа обучающегося	
				Лаб.	Пр.		
1.	Введение. Основные понятия алгоритмизации	1	1,5	-	1,5	1,5	практические работы
2.	Основы программирования в среде Scratch	1	1,5	-	4,5	3	практические работы
3.	Язык и среда программирования Кумир	1	3	-	6	4,5	практические работы
4.	Основы языка программирования на языке Python	1	3	-	6	4,5	практические работы
5.	Структуры данных в языке программирования Python	1	3	-	6	4,5	практические работы
ИТОГО			12	-	24	18	Экзамен

4.2. Развернутое описание содержания учебного материала по разделам (темам)

Введение. Основные понятия алгоритмизации. Понятие алгоритма. Свойства алгоритмов. Формы записей алгоритмов. Общие принципы построения алгоритмов. Основные алгоритмические конструкции: линейные, разветвляющиеся (полное или неполное ветвление), циклические (детерминированный цикл, цикл с предусловием, цикл с постусловием). Основные базовые типы данных и их характеристика. Основные этапы решения задач.

Основы программирования в среде Scratch. Установка среды программирования Scratch. Знакомство с официальным сайтов в сети Интернет. Написание мини-игры в среде Scratch. Основные понятия среды: спрайты, скрипты, блок, сцена, сообщение, клон. Блоки

для программирования движения спрайта и изменения внешнего вида. Понятие сенсора в среде программирования Scratch. Основные инструменты встроенного графического редактора программной среды Scratch. Создание переменных, списков и подпрограмм.

Язык и среда программирования Кумир. Установка среды программирования Кумир. Алфавит языка. Типы данных языка программирования Кумир. Реализация линейных, разветвляющихся и циклических алгоритмов в Кумир. Решение алгоритмических задач. Знакомство с исполнителями среды программирования Кумир. Написание программ для исполнителя «Робот». Создание и обработка массивов. Обработка строк.

Основы языка программирования на Python. Эволюция языков программирования и языка программирования Python. Общая информация о языке. Алфавит языка. Знакомство с официальным сайтом в сети Интернет. Интерпретатор языка программирования и его установка. Создание и запуск программы во встроенной среде разработки. Математические операции языка Python. Основы алгебры логики. Логические операции с высказываниями: конъюнкция, дизъюнкция, инверсия. Простой условный оператор. Сокращенный условный оператор. Оператор цикла for. Оператор цикла while.

Структуры данных в языке программирования Python. Понятие списка в Python. Хранение списка в памяти. Операции для работы со списками. Ввод и вывод списков. Обработка списков. Понятие словаря в Python. Хранение словаря в памяти. Операции для работы со словарями. Основные операции работы со строками в языке Python. Обработка строк.

5. Образовательные технологии, в том числе перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

В программе данного курса предусмотрено использование в учебном процессе активных форм проведения занятий и организации самостоятельной работы, рейтинговой технологии оценивания результатов обучения студентов. В учебном процессе используются как активные, так и интерактивные формы проведения занятий: технология учебной дискуссии, мозговой штурм.

Аудиторные занятия проводятся в интерактивной форме с использованием мультимедийного обеспечения (ноутбук, проектор) и технологии проблемного обучения.

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине: технологии смешанного обучения, мультимедиа технологии, облачные технологии, технологии визуализации (инфографика, презентации).

6. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся по дисциплине осуществляется в следующих формах:

- установка и настройка программного обеспечения для работы с языком программирования;
- составление алгоритма для решения задач в словесном виде и в виде блок-схем;
- написание и тестирования программ на языке Python;
- изучение и анализ предложенных электронных образовательных ресурсов, содержащих информацию по алгоритмизации и программированию;
- повторение материала с использованием конспектов.

Полностью весь методический материал по обеспечению самостоятельной работы студентов приводится в приложении 1 к настоящей программе. Доступ к методическим материалам осуществляется через сайт Шуйского филиала ИвГУ, в том числе через систему Moodle.

7. Характеристика оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине

Текущая аттестация осуществляется с использованием накопительного рейтингового оценивания.

Для текущего контроля успеваемости применяются:

- Практические работы на компьютере: 16 работ по 3 балла каждая (48 баллов);
- Практические задания для самостоятельного выполнения: 16 заданий по 2 балла (32 балла);
- Опорный конспект по темам, предложенным преподавателем (10 баллов);

Помимо этого учитывается:

- посещаемость (отсутствие пропусков занятий) – 10 баллов.

Критерии оценки практических работ:

- 3 балла за практическую работу выставляется, если работа выполнена полностью правильно и самостоятельно, а студент способен уверенно отвечать на вопросы преподавателя о проделанной работе;
- 2 балла выставляется, если работа выполнена в целом корректна, но имеются неточности в ответах на вопросе преподавателя или некоторые действия выполнены неоптимально;
- 1 балл выставляется, когда работа не может быть завершена студентом самостоятельно, без помощи преподавателя.
- 0 баллов выставляется, если работа не была выполнена.

Критерии оценки практических заданий для самостоятельного выполнения:

- 2 балла за практическую работу выставляется, если работа выполнена полностью правильно и самостоятельно, а студент способен ответить на вопросы преподавателя о проделанной работе;
- 1 балл выставляется, если работа выполнена в целом корректна, но имеются неточности в ответах на вопросе преподавателя;
- 0 баллов выставляется, если работа не была выполнена.

Критерии оценки опорного конспекта по темам: Написание конспекта по каждой из пяти тем оценивается от 0 до 2 баллов: 2 балла – в конспекте отражены основные положения темы, имеются схемы, выделения особо значимой информации; 1 балл - в конспекте отражены основные положения темы, но отсутствуют признаки попытки переработки информации со стороны студенты; 0 баллов – основные положения темы не были отражены в конспекте.

Шкала количества баллов, выставляемых за посещение занятий:

Процент посещения занятий от их общего количества	**Количество баллов, выставляемое за посещаемость**
85%-100%	10 баллов
80%-84%	9 баллов
75%-79%	8 баллов
70%-74%	7 баллов

65%-69%	6 баллов
60%-64%	5 баллов
55%-59%	4 балла
50%-54%	3 балла
45%-49%	2 балла
40%-44%	1 балл
0%-39%	0 баллов

Оценка результатов освоения дисциплины при рейтинговой системе оценивания формируется по 100-балльной шкале из результатов текущего контроля успеваемости и результатов, полученных на экзамене. Экзамен проводится в форме выполнения практической работы (максимально 10 баллов) и интегрального тестового задания, включающего вопросы по всем темам курса (максимально 20 баллов).

Оценка на экзамене выставляется по следующей шкале:

Шкала оценки результатов освоения дисциплины «Основы алгоритмизации и программирования»
при рейтинговой системе оценивания

Повышенный уровень	Базовый уровень	Пороговый уровень	Ниже порогового уровня
отлично	хорошо	удовлетворительно	неудовлетворительно
85-100 баллов	71-84 балла	61-70 баллов	60 и менее баллов

8. Учебно-методическое и информационное обеспечение дисциплины

Учебная и учебно-методическая литература:

– основная литература:

1. Анеликова Л. А. Программирование на алгоритмическом языке КуМир : учебное пособие / Л. А. Анеликова, О. Б. Гусева. – Москва : СОЛОН-ПРЕСС, 2012. – 48 с. – («Элективный курс. Профильное обучение»). – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=226976. – ISBN 978-5-91359-098-5. – Текст : электронный.
2. Карякин, М. И. Технологии программирования и компьютерный практикум на языке Python : учебное пособие / М. И. Карякин, К. А. Ватульян, Р. М. Мнухин ; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2022. – 244 с. : ил., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=698687. – Библиогр. в кн. – ISBN 978-5-9275-4108-9. – Текст : электронный.
3. Лубашева, Т. В. Основы алгоритмизации и программирования : учебное пособие / Т. В. Лубашева, Б. А. Железко. – Минск : РИПО, 2016. – 378 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=463632. – Библиогр. в кн. – ISBN 978-985-503-625-9. – Текст : электронный.

– дополнительная литература:

1. Зюзьков, В. М. Программирование : учебное пособие / В. М. Зюзьков ; Томский Государственный университет систем управления и радиоэлектроники (ТУСУР). – Томск : Эль Контент, 2013. – 186 с. : ил. – Режим доступа: по подписке. – URL:

https://biblioclub.ru/index.php?page=book&id=480616. – Библиогр. в кн. – ISBN 978-5-4332-0141-5. – Текст : электронный.
2. Информатика : учебное пособие / Е. Н. Гусева, И. Ю. Ефимова, Р. И. Коробков [и др.]. – 5-е изд., стер. – Москва : ФЛИНТА, 2021. – 260 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=83542. – Библиогр. в кн. – ISBN 978-5-9765-1194-1. – Текст : электронный.
3. Колокольникова, А. И. Практикум по информатике: основы алгоритмизации и программирования / А. И. Колокольникова. – Москва ; Берлин : Директ-Медиа, 2019. – 424 с. : ил., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=560695. – Библиогр. в кн. – ISBN 978-5-4499-0097-5. – DOI 10.23681/560695. – Текст : электронный.
4. Лебеденко, Л. Ф. Основы программирования на C++ : учебное пособие : [16+] / Л. Ф. Лебеденко, О. И. Моренкова. – 2-е изд., перераб. и доп. – Новосибирск : Сибирский государственный университет телекоммуникаций и информатики, 2021. – 200 с. : ил., табл., схем. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=694769. – Библиогр. в кн. – Текст : электронный.
5. Незнанов, А. А. Программирование и алгоритмизация : учебник / А. А. Незнанов ; науч. ред. В. П. Кутепов. - Москва : Академия, 2010. - 304 с.
6. Хахаев, И. А. Практикум по алгоритмизации и программированию на Python: курс : учебное пособие : [16+] / И. А. Хахаев. – 2-е изд., исправ. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 179 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=429256. – Библиогр. в кн. – Текст : электронный.
7. Шелудько, В. М. Основы программирования на языке высокого уровня Python : учебное пособие : [16+] / В. М. Шелудько. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2017. – 147 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=500056. – Библиогр. в кн. – ISBN 978-5-9275-2649-9. – Текст : электронный.

Интернет-ресурсы:

1. Самоучитель Python 3 для начинающих. Режим доступа: https://pythonworld.ru/
2. Официальный сайт языка программирования Python. Режим доступа: https://www.python.org/
3. Официальный сайт среды программирования Scratch. Режим доступа: https://scratch.mit.edu/
4. Страница системы программирования Кумир на сайте ФГУ "Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук". Режим доступа: https://www.niisi.ru/kumir/
5. Единая коллекция цифровых образовательных ресурсов http://school-collection.edu.ru/
6. Образовательные сайты и порталы. http://school2100.com/
7. Видеоматериалы для учителей. https://video.1sept.ru/subject/programmirovanie/archive
8. Профессиональный сайт учителей «Методисты». http://nsportal.ru/

Современные профессиональные базы данных и информационно-справочные системы:
ЭБС «Университетская библиотека онлайн»;
Коллекция Шуйского филиала ИвГУ на национальном цифровом ресурсе «Руконт».
Консультант Плюс http://www.consultant.ru/

Методическое хранилище Ивановского института развития образования].- http://www.iroio.ru/;

Программное обеспечение:

В процессе изучения дисциплины используются офисные пакеты Microsoft Office, Open Office, среда разработки Кумир версии 2.1.0 (rc7), интерпретатор языка Python 3.5 (или выше) и среды разработки IDLE и PyScripter, программа для просмотра и чтения файлов PDF Adobe Acrobat Reader DC, программа для воспроизведения флэш-анимации в браузерах Adobe Flash Player, вэб-обозреватели Google Chrome, Opera, Антивирус Касперского, 1С университет.

9. Материально-техническое обеспечение дисциплины

1. Учебная аудитория для проведения занятий лекционного типа, укомплектованная специализированной мебелью и техническими средствами обучения, служащими для предоставления учебной информации большой аудитории.
2. Учебная аудитория для проведения занятий семинарского типа, укомплектованная специализированной мебелью и техническими средствами обучения.
3. Учебная аудитория для проведения групповых и индивидуальных консультаций, укомплектованная специализированной мебелью и техническими средствами обучения.
4. Учебная аудитория для проведения текущего контроля и промежуточной аттестации, укомплектованная специализированной мебелью и техническими средствами обучения.
5. Помещение для самостоятельной работы, оснащенное специализированной мебелью, компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.
6. Учебно-наглядные пособия для занятий лекционного типа, обеспечивающие тематические иллюстрации: презентации по программированию и алгоритмизации.

Авторы рабочей программы дисциплины: А.В. Ермошин, доцент кафедры математики, информатики и методики обучения, кандидат технических наук, доцент; В.А. Смирнов, преподаватель (аспирант) кафедры математики, информатики и методики обучения.

Программа рассмотрена на заседании кафедры математики, информатики и методики обучения

«16» февраля 2023 г., протокол № 7

Программа обновлена
протокол заседания кафедры № _____ от «_____» _____20 __ г.
Согласовано:
Руководитель ОП _____ С.В. Белов
(подпись)

Программа обновлена
протокол заседания кафедры № _____ от «_____» _____20 __ г.
Согласовано:
Руководитель ОП _____ С.В. Белов
(подпись)

Программа обновлена
протокол заседания кафедры № _____ от «_____» _____20 __ г.
Согласовано:
Руководитель ОП _____ С.В. Белов
(подпись)

Приложение 1.

Методические указания для обучающихся по освоению дисциплины

Сопровождение самостоятельной работы студентов по курсу «Основы алгоритмизации и программирования» может быть организовано в следующих формах:

– согласование индивидуальных заданий (виды и темы заданий, сроки представления результатов) самостоятельной работы студента в пределах часов, отведенных на самостоятельную работу;

– консультации (индивидуальные и групповые), в том числе с применением дистанционных технологий взаимодействия;

– промежуточный контроль хода выполнения заданий, строящихся на основе различных способов самостоятельной деятельности в открытой информационной среде.

Преподаватель определяет содержание самостоятельной работы, график её выполнения; создает сетевую информационную и коммуникационную среду для организации самостоятельной работы.

Одной из важнейших особенностей заданий для самостоятельной работы по курсу «Основы алгоритмизации и программирования» является их ориентация на формирование способности оценивать приоритеты и ограничения при выборе средств информационных технологий для решения управленческих задач, а также на мотивирование самообразовательной деятельности с использованием ИКТ.

К типовым заданиям для самостоятельной работы студентов относятся:

– Конспектирование и реферирование литературных источников по тематике учебного курса.

– Решение задач, связанных с установкой программного обеспечения для написания программ, создание программ на языке Python для решения прикладных задач;

– Изучение и анализ предложенных электронных образовательных ресурсов, содержащих информацию по основам алгоритмизации и программирования;

– Подготовка к тестированию на занятиях по каждой теме курса;

– Повторение материала с использованием конспектов.

Перечень вопросов для самостоятельного изучения

1. Определение алгоритма. Свойства алгоритма. Формы записи алгоритмов. Примеры.
2. Запись алгоритмов блок-схемами. Основные элементы блок-схем.
3. Алгоритмы с ветвлением. Пример алгоритма.
4. Алгоритм цикла с предусловием. Пример алгоритма.
5. Алгоритм цикла с постусловием. Пример алгоритма.
6. Алгоритм цикла с управляющей переменной. Пример алгоритма.
7. Логические и арифметические операции в среде программирования Scratch
8. Дополнения «Музыка», «Перо», «Текст в речь», «Перевести» среды программирования Scratch
9. Среда программирования Snap! в обучении школьников на курсах дополнительного образования
10. Среда программирования Alice в обучении школьников на курсах дополнительного образования
11. Проведение занятий по программированию с использованием учебного исполнителя «Черепашка» в среде Кумир

12. Проведение занятий по программированию с использованием учебного исполнителя «Чертежник» в среде Кумир
13. Функции для работы со строками среды программирования Кумир
14. Команды для работы с файлами среды программирования Кумир
15. Массивы. Линейный и бинарный поиск в массиве.
16. Сортировка массивов. Сортировка выбором. Сортировка методом "пузырька". Сортировка вставками. Сортировка подсчетом. Сортировка слиянием. Быстрая сортировка Хоара.
17. Переменные. Типы данных в Python.
18. Структура программы. Команда присваивания в Python.
19. Ввод-вывод данных. Формат выводимых данных в Python.
20. Алгоритм линейной структуры в Python.
21. Структура if, классификация в Python. Примеры.
22. Алгоритмы циклической итерационной структуры. Оператор цикла while в Python. Примеры использования.
23. Строки как последовательности символов в Python. Примеры использования.
24. Алгоритмы циклической итерационной структуры. Оператор цикла обхода for в Python. Примеры использования.
25. Списки – изменяемые последовательности в Python. Примеры использования.

Задания для организации практических занятий:

Практическая работа №1 «Основы алгоритмизации»

Цель работы: усвоить понятия: алгоритм как фундаментальное понятие информатики, способы описания, основные типы алгоритмов, освоить принципы решения задач с использованием основных алгоритмических конструкций.

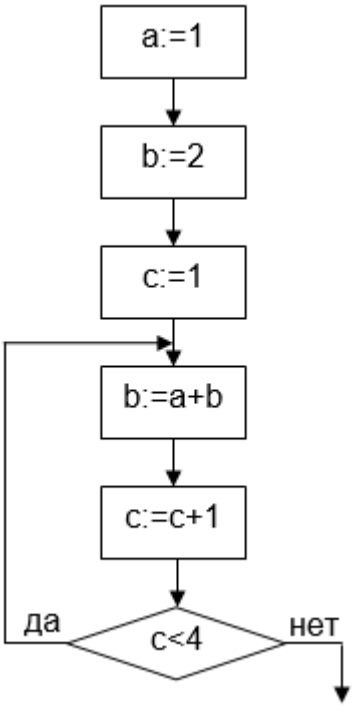
Общие сведения: Этапы решения задачи на ЭВМ. Работа по решению любой задачи с использованием компьютера делится на следующие этапы:

- Постановка задачи.
- Формализация задачи.
- Построение алгоритма.
- Составление программы на языке программирования.
- Отладка и тестирование программы.
- Проведение расчетов и анализ полученных результатов.

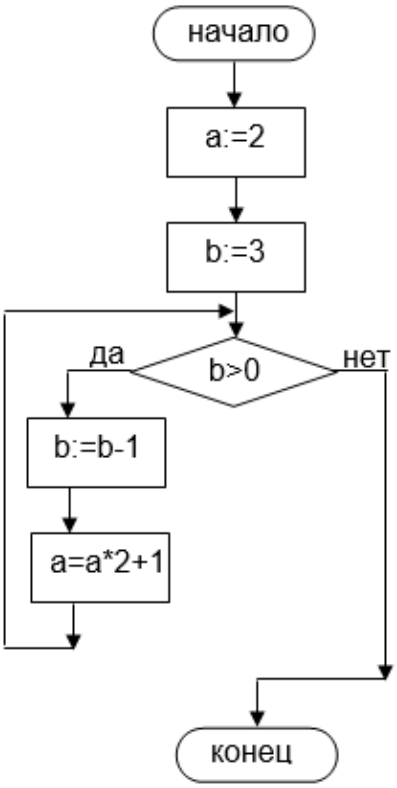
Блок-схема алгоритма — графическое изображение алгоритма в виде связанных между собой с помощью стрелок (линий перехода) и блоков — графических символов, каждый из которых соответствует одному шагу алгоритма. Внутри блока дается описание соответствующего действия.

Задачи:

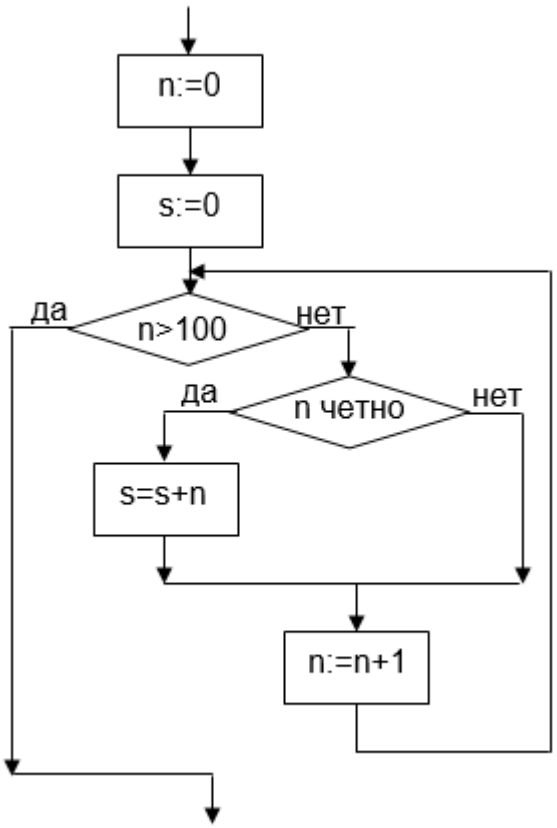
1. Составить алгоритм решения задачи с помощью блок-схем, используя конструкцию линейного алгоритма.
    a. Определить площадь трапеции по введенным значениям оснований (a и b) и высоты (h)
    b. Вычислить площадь треугольника со сторонами a, b, c по формуле Герона.
2. Определите значение переменной b после выполнения фрагмента алгоритма.



3. Определите значение переменной a после выполнения алгоритма.



4. Определите значение переменной s после выполнения фрагмента алгоритма.



5. Составить алгоритм решения задачи с помощью блок-схем, используя конструкцию алгоритма с ветвлением:
    a. Определить максимальное четное число из двух введенных;
    b. Определить, можно ли из отрезков с длинами x, y и z построить треугольник.
    c. Определить, в каком квадранте находится точка с координатами x и y.
6. Составить алгоритм решения задачи с помощью блок-схем, используя конструкцию циклического алгоритма.
    a. Найти сумму чисел, кратных трем, в диапазоне от 0 до 50.
    b. Найти сумму первых десяти чисел, кратных пяти.
    c. Определить количество целых чисел, кратных 3 (от 3 и далее), дающих в сумме число, превышающее 200.
7. Составить алгоритм решения практической задачи с помощью блок-схем:
    a. Вы положили в банк 1500 рублей. Определить, сколько денег будет на Вашем вкладе через 1 год, если каждый месяц вклад увеличивается на 0.76 % от суммы предыдущего месяца.
    b. Решив заняться легкой атлетикой, Вы пробежали в первый день 2 км. Сколько километров Вы пробежите за 2 недели, если каждый день Вы увеличиваете дистанцию на 10% от предыдущего дня?

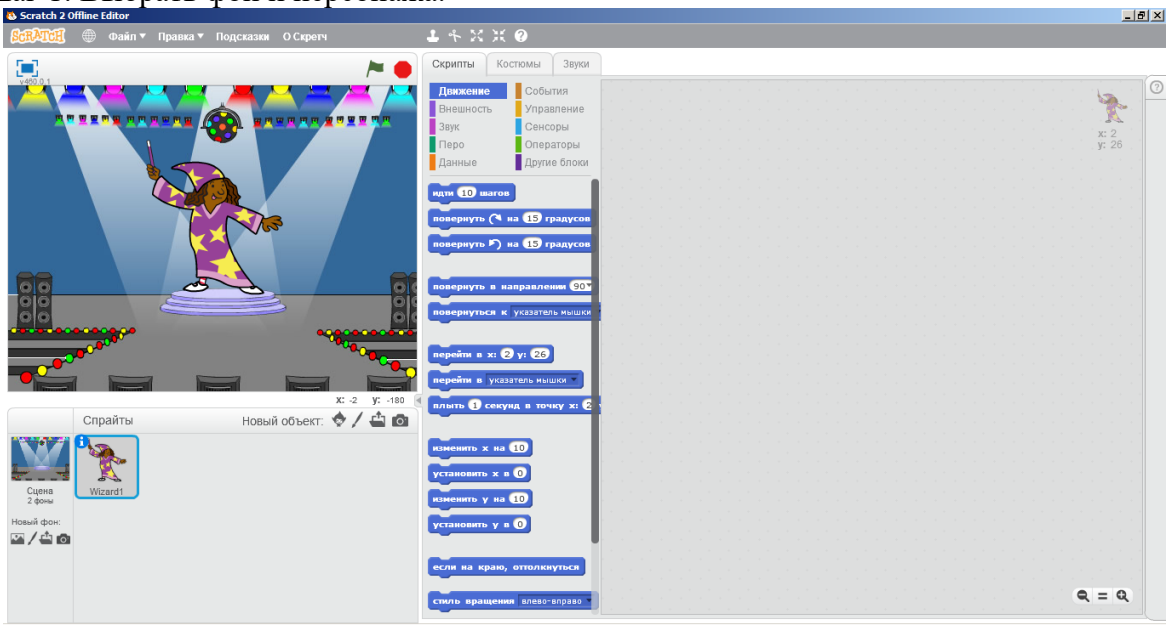
Практическая работа №2 «Простые проекты в среде Scratch»

Часть 1 «Сообщения»

Факт передачи сообщения является событием, на которое могут реагировать спрайты. При помощи этого механизма можно организовать запуск скриптов одного спрайта из другого.

Передача сообщений происходит невидимо для пользователя, но на каждое из сообщений может среагировать любой спрайт.

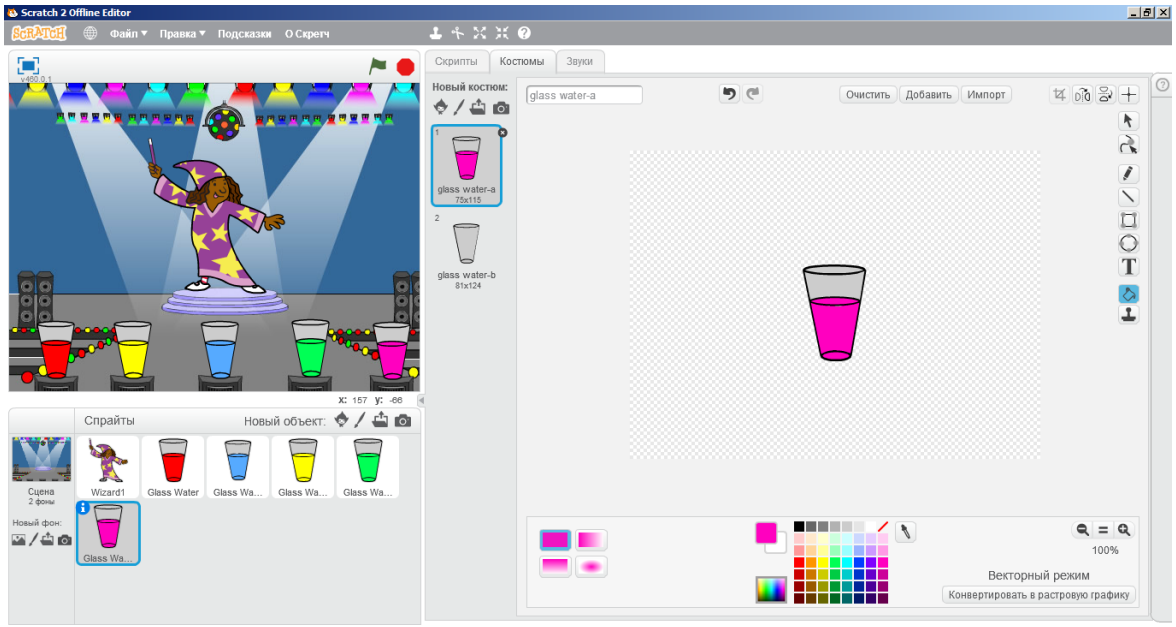
Шаг 1: Выбрать фон и персонажа.



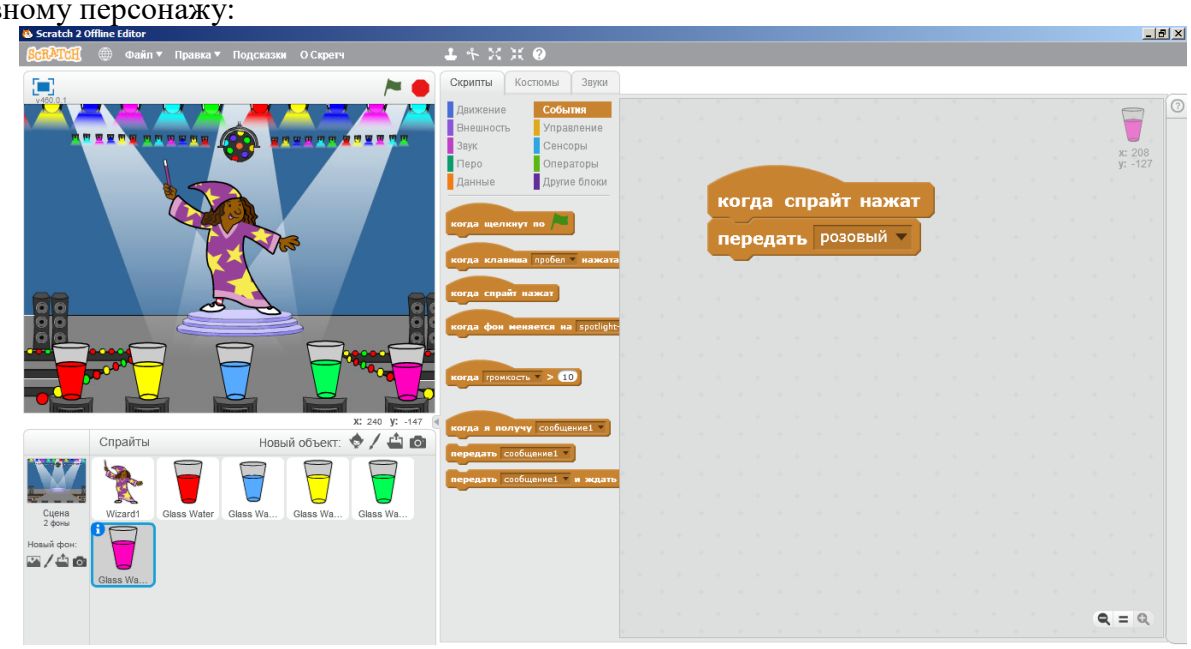
Шаг 2: Выбираем предметы, который будет добавлять эффекты персонажу. Рекомендуем использовать спрайт

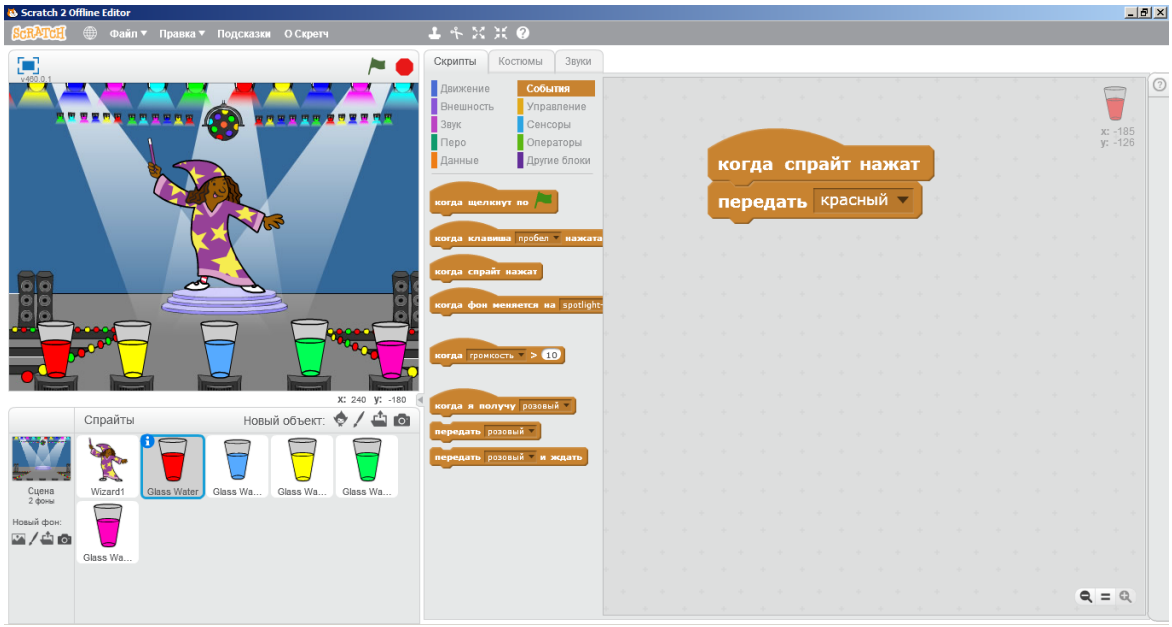


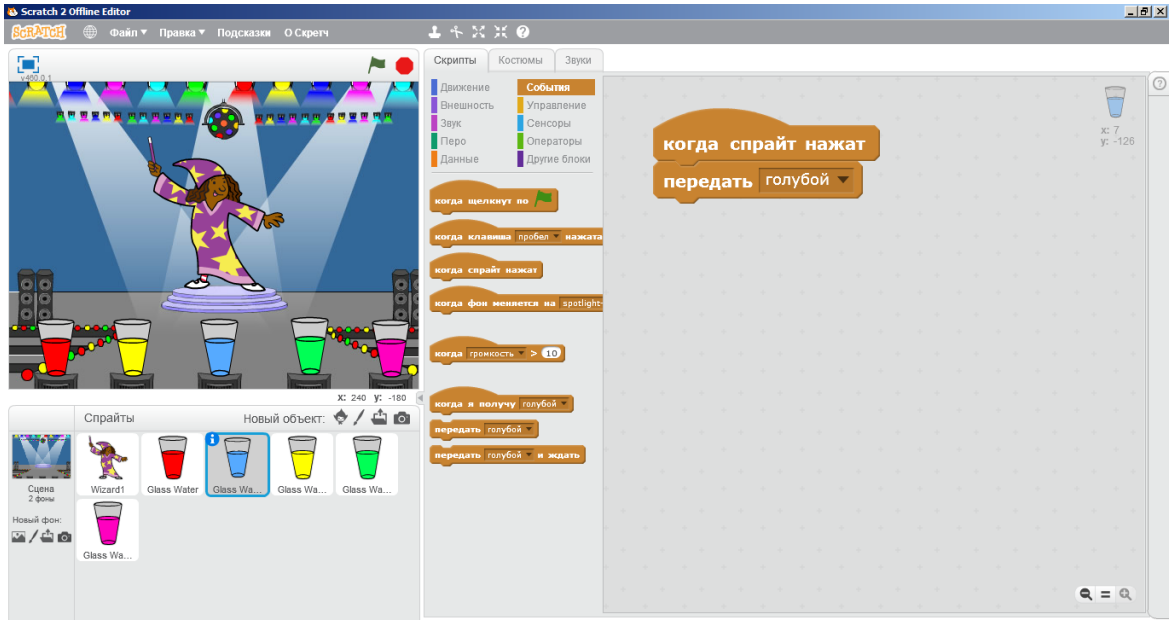
раскрашенный в разные цвета.



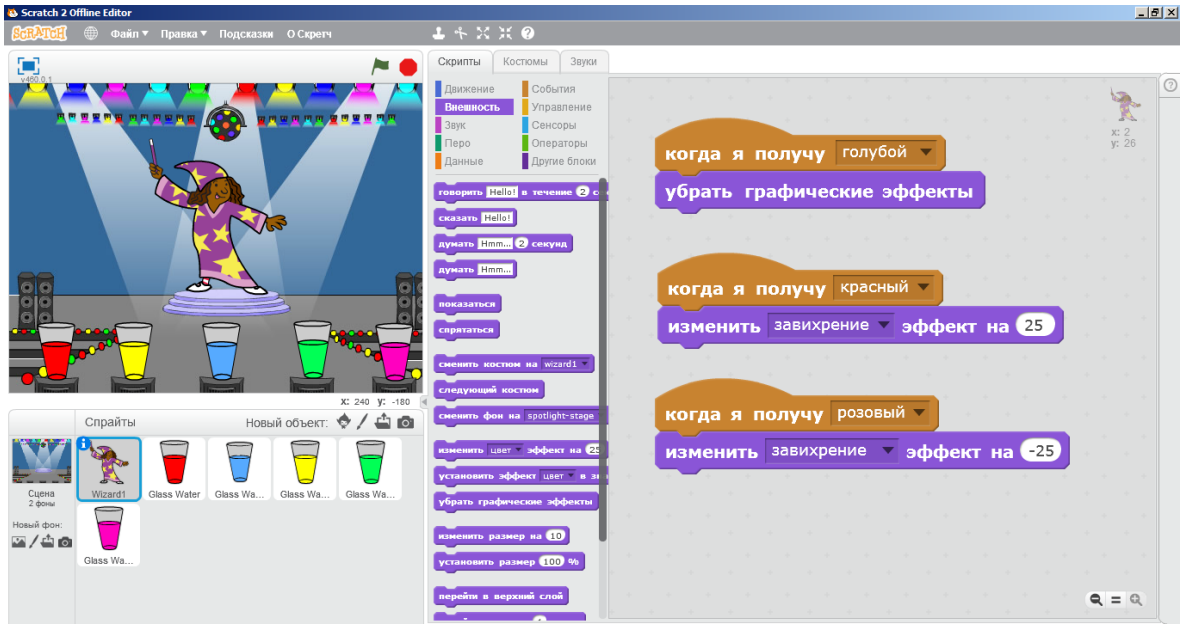
Каждому «стакану» назначаем соответствующий скрипт о передаче сообщения нашему главному персонажу:




Scratch 2 Offline Editor
Файл
Правка
Подсказки
О Скретч
Скрипты
Костюмы
Звуки
Движение
Внешность
Звук
Перо
Данные
События
Управление
Сенсоры
Операторы
Другие блоки
когда щелкнут по
когда клавиша пробел нажата
когда спрайт нажат
когда фон меняется на spotlight
когда громкость > 10
когда я получу розовый
передать розовый
передать розовый и ждать
когда спрайт нажат
передать красный
Спрайты
Новый объект:
Wizard1
Glass Water
Glass Wa...
Сцена
2 фоны
Новый фон:


Scratch 2 Offline Editor
Файл
Правка
Подсказки
О Скретч
Скрипты
Костюмы
Звуки
Движение
Внешность
Звук
Перо
Данные
События
Управление
Сенсоры
Операторы
Другие блоки
когда щелкнут по
когда клавиша пробел нажата
когда спрайт нажат
когда фон меняется на spotlight
когда громкость > 10
когда я получу голубой
передать голубой
передать голубой и ждать
когда спрайт нажат
передать голубой
Спрайты
Новый объект:
Wizard1
Glass Water
Glass Wa...
Сцена
2 фоны
Новый фон:

Шаг 3: В персонаже добавляем обработку сообщений:



Шаг 4: Проверить работу скриптов. Красный стакан увеличивает эффект завихрения, а розовый – уменьшает. Нажатие на голубой стакан возвращает спрайт в первоначальное состояние.

Задание:
1) Сменить эффект «завихрение» на любой другой.
2) Самостоятельно добавить эффекты по двум оставшимся стаканам.

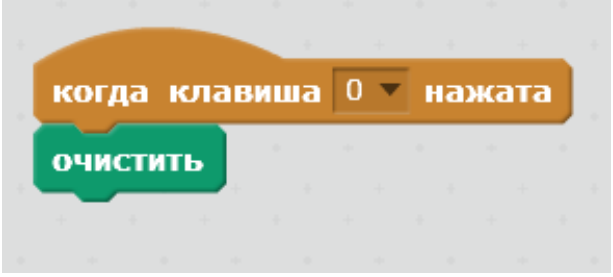
Сохранить проект под именем «ФамилияИО_Сообщения».

Часть 2 «Перо»

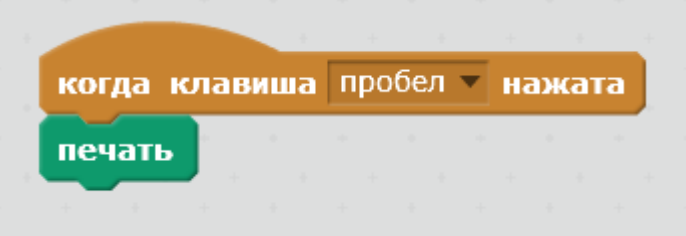
Перо необходимо для рисования линий на сцене. Все нарисованные линии могут быть удалены при выполнении соответствующей команды («очистить»).

Перо может находиться в двух состояниях: опущено (когда при перемещении спрайта на сцене отрисовывается проделанный им путь), поднято (когда рисования не происходит).

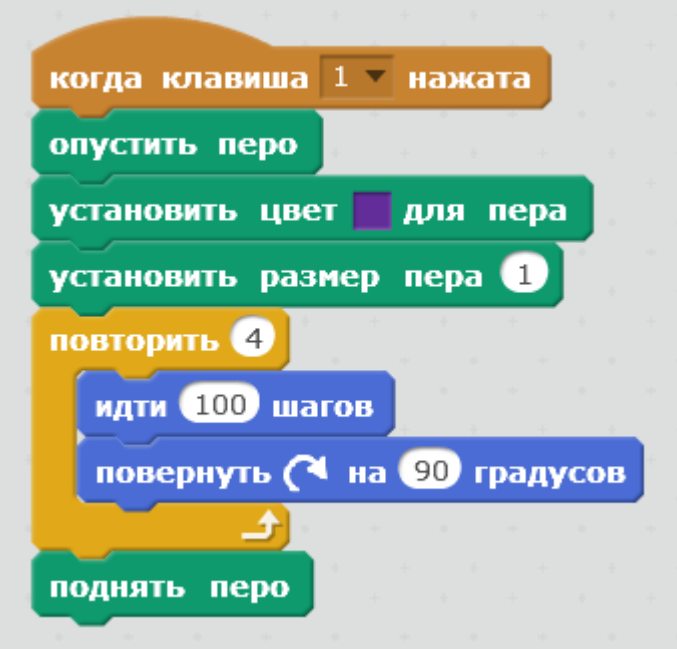
В новом проекте запрограммировать у кота следующие действия:
1) Очистка нарисованного на сцене по нажатию на клавишу 0 на клавиатуре.



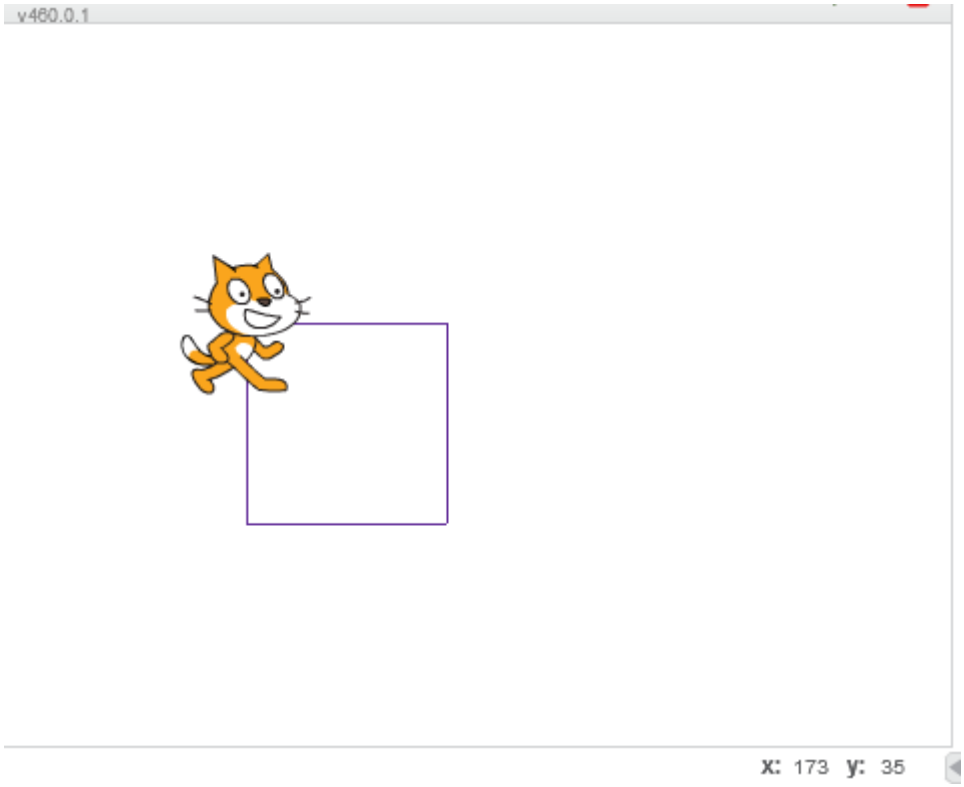
2) Оттиска спрайта по нажатию на клавишу «пробел» на клавиатуре.



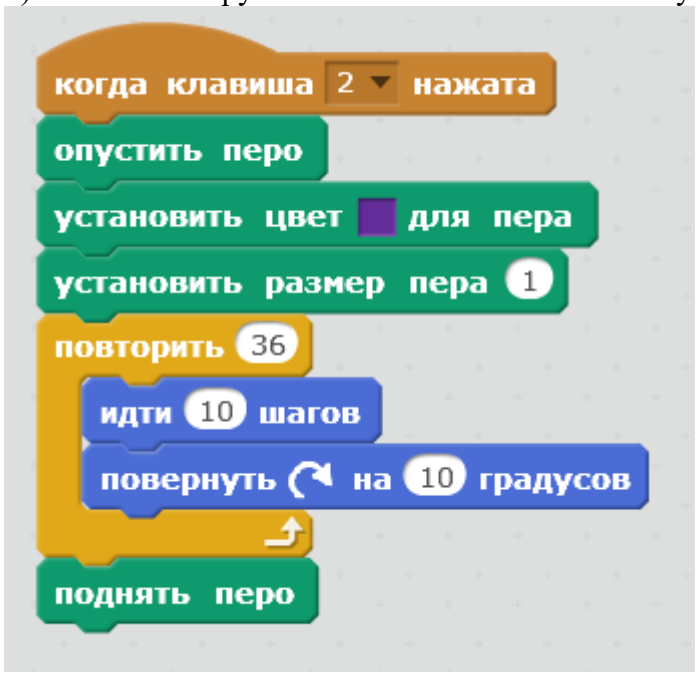
3) Рисование квадрата по нажатию на клавишу 1 на клавиатуре.



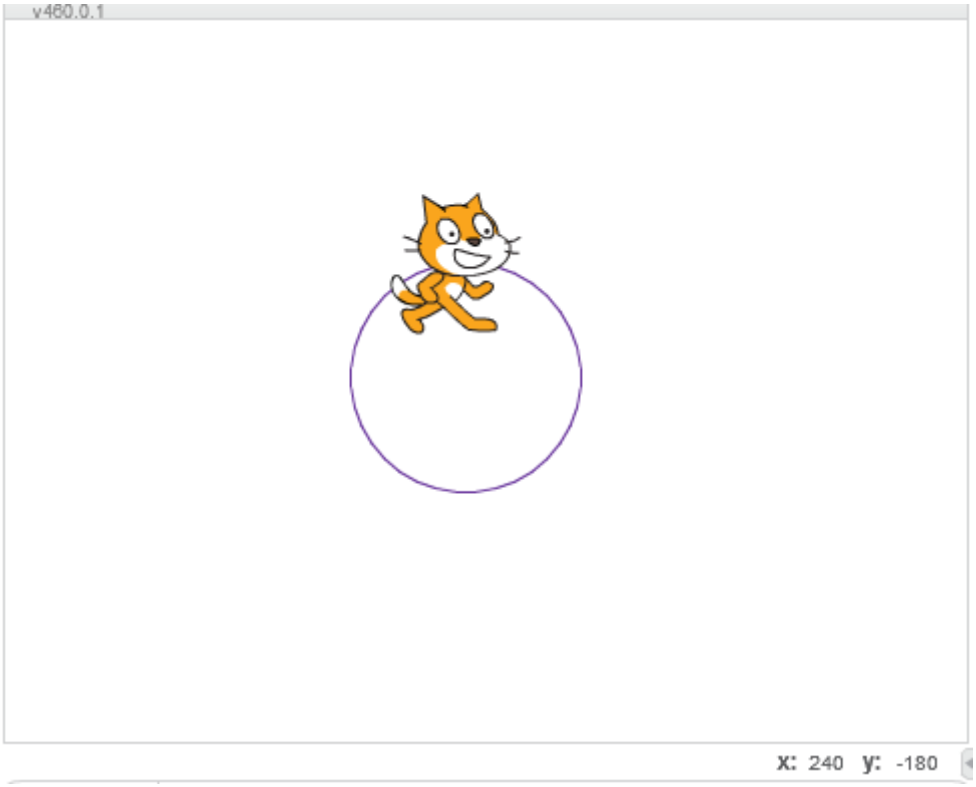
Результат нажатия на кнопку:



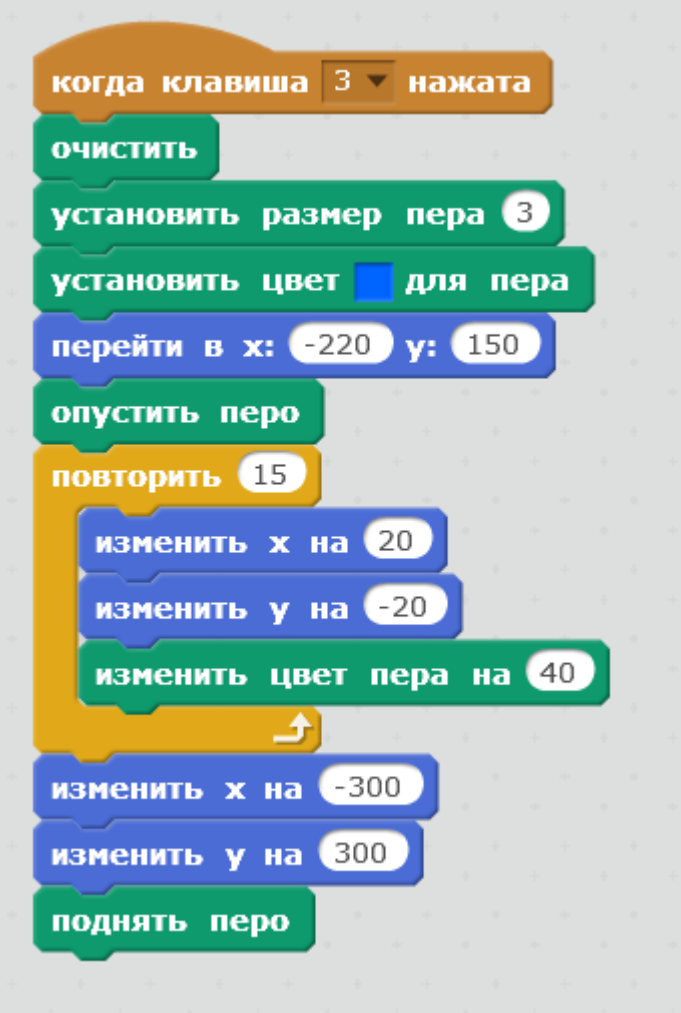
4) Рисование окружности по нажатию на клавишу 2 на клавиатуре.



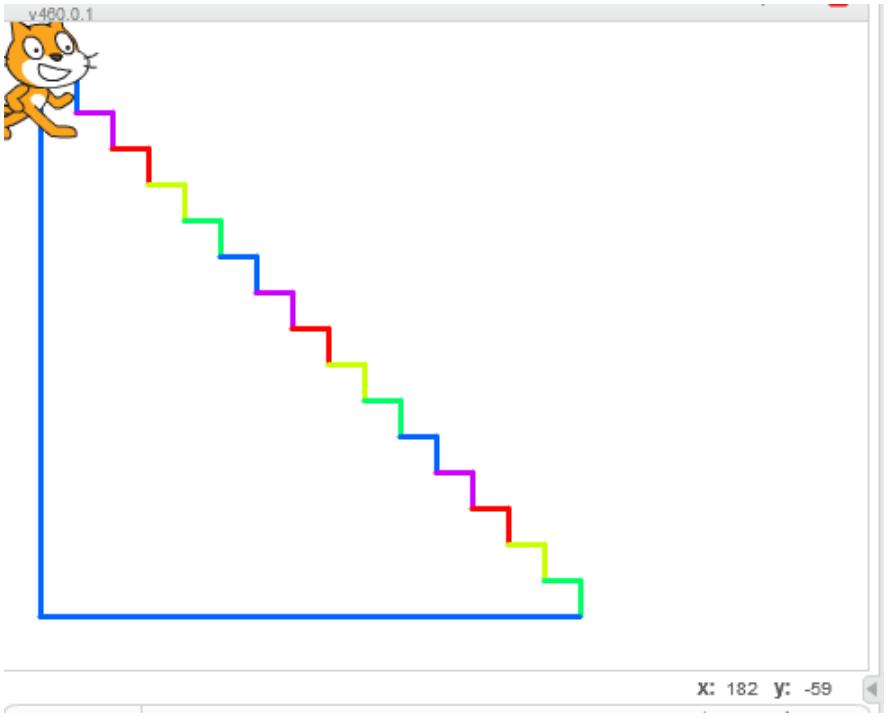
Результат нажатия на кнопку:



5) Рисование лестницы по нажатию на клавишу 3 на клавиатуре.



Результат нажатия на кнопку:



Сохранить проект под именем «ФамилияИО_Перо».

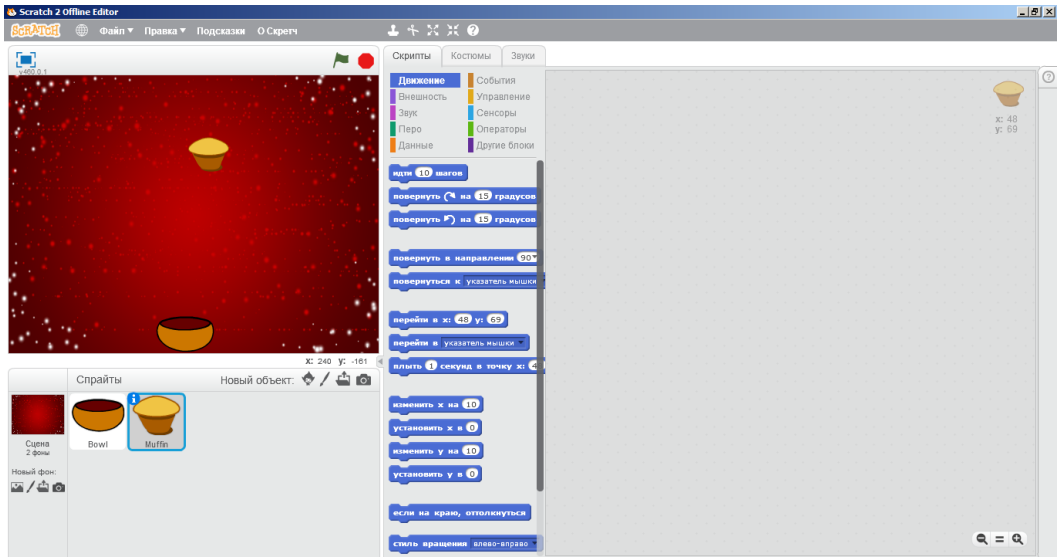
Практическая работа №3 «Переменные и создание игры в Scratch»

Часть 1 «Подготовка игрового поля»

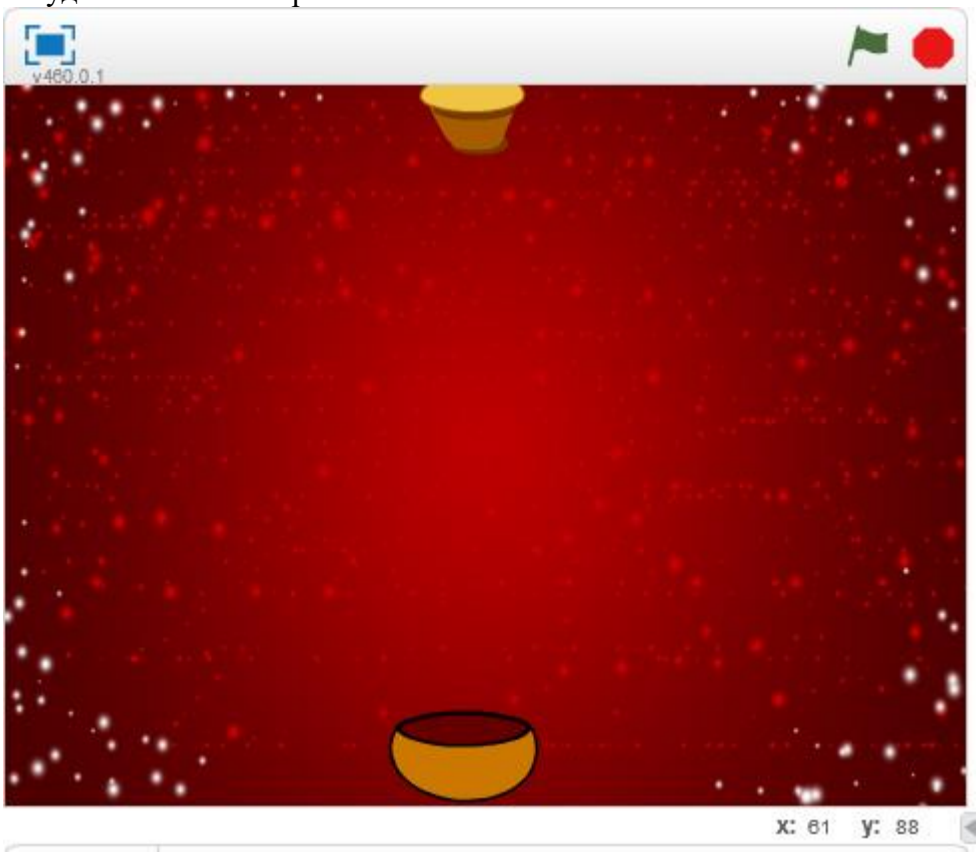
Шаг 1: Выбрать фон и спрайты:

1) Спрайт, который будет основой для падающих объектов (на скриншоте ниже таким спрайтом является кекс);

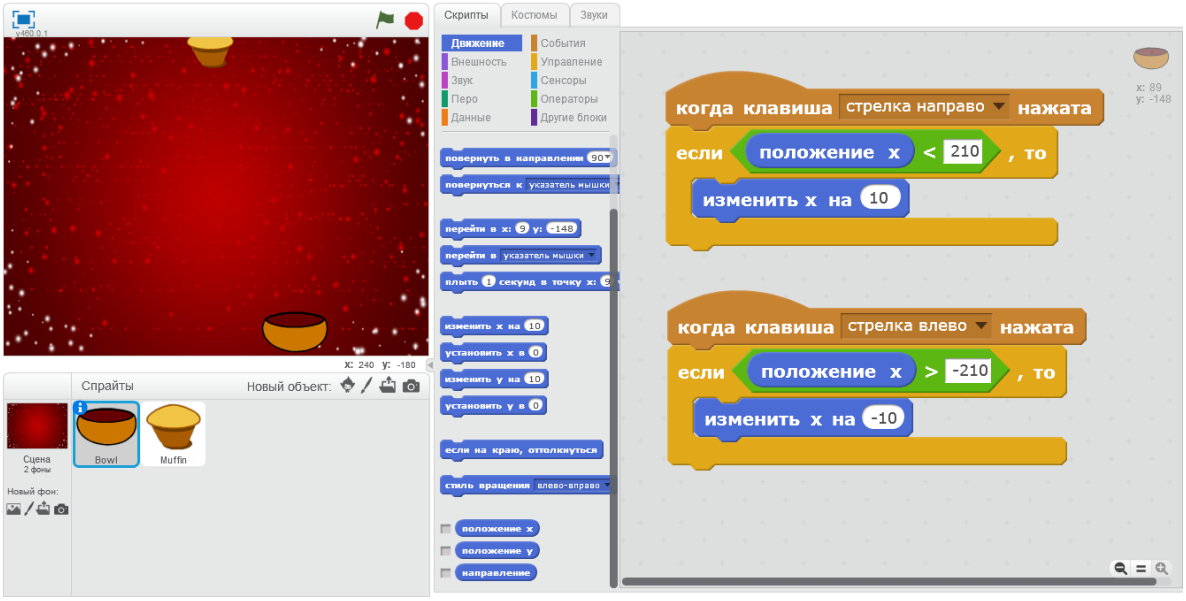
2) Спрайт, который будет их ловить.



Шаг 2: Расположить предмет для ловли объектов в нижней части сцены. Предмет, который нужно будет ловить – в верхней.



Шаг 3: Добавить нижнему спрайту команды:



Шаг 4: Проверить работу скриптов. Объект должен перемещаться нажатием стрелок на клавиатуре, но не уходить за край сцены.

Часть 2 «Клоны»

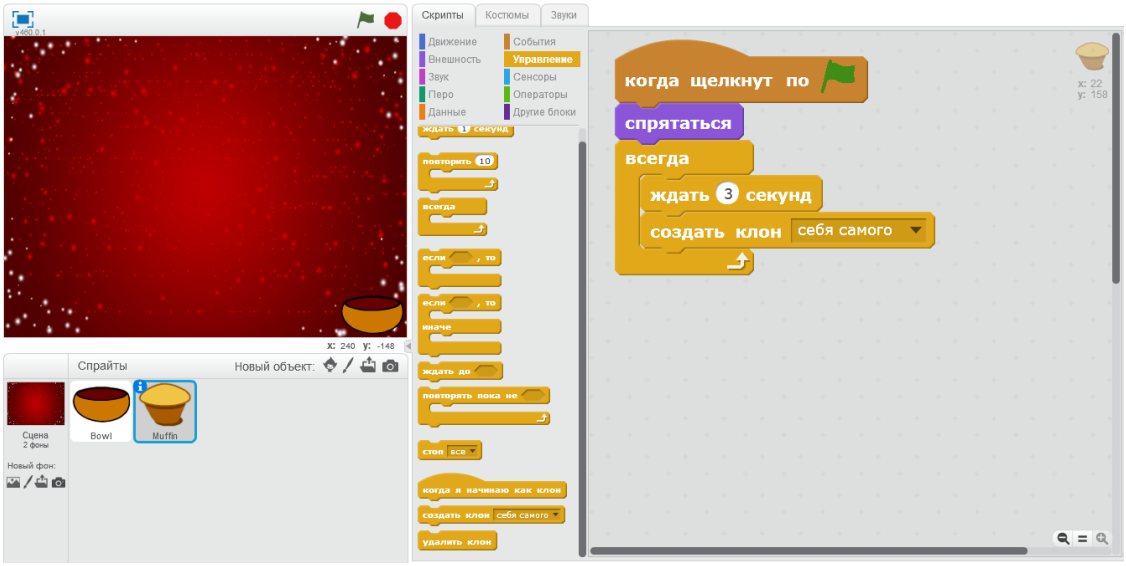
Помимо механизма дублирования объектов из раздела «Спрайты», в разделе «Управление» есть команда, которая называется «Создать клон «себя самого»».

В отличие от процесса дублирования при клонировании новый спрайт не создаётся. Клонирование приводит к возникновению на сцене точной копии клонируемого спрайта, обладающей такими же свойствами и скриптами.

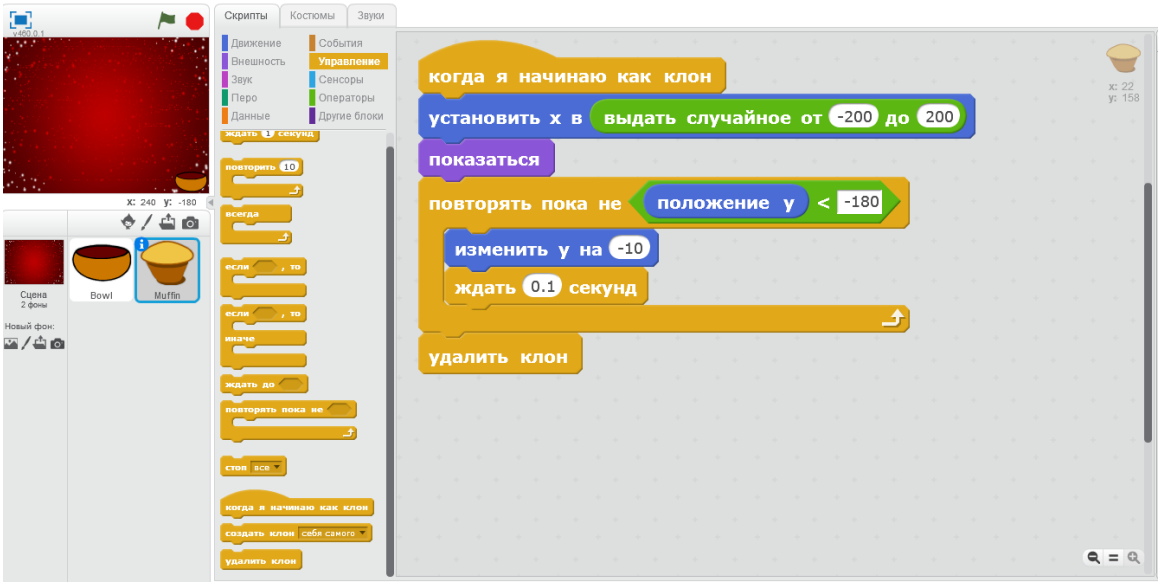
Клонировать спрайт можно только с помощью специальной команды в процессе работы программы. При её остановке с помощью кнопки  все клоны уничтожаются.

В рамках работы программы не рекомендуется допускать создание более **300** клонов для спрайтов.

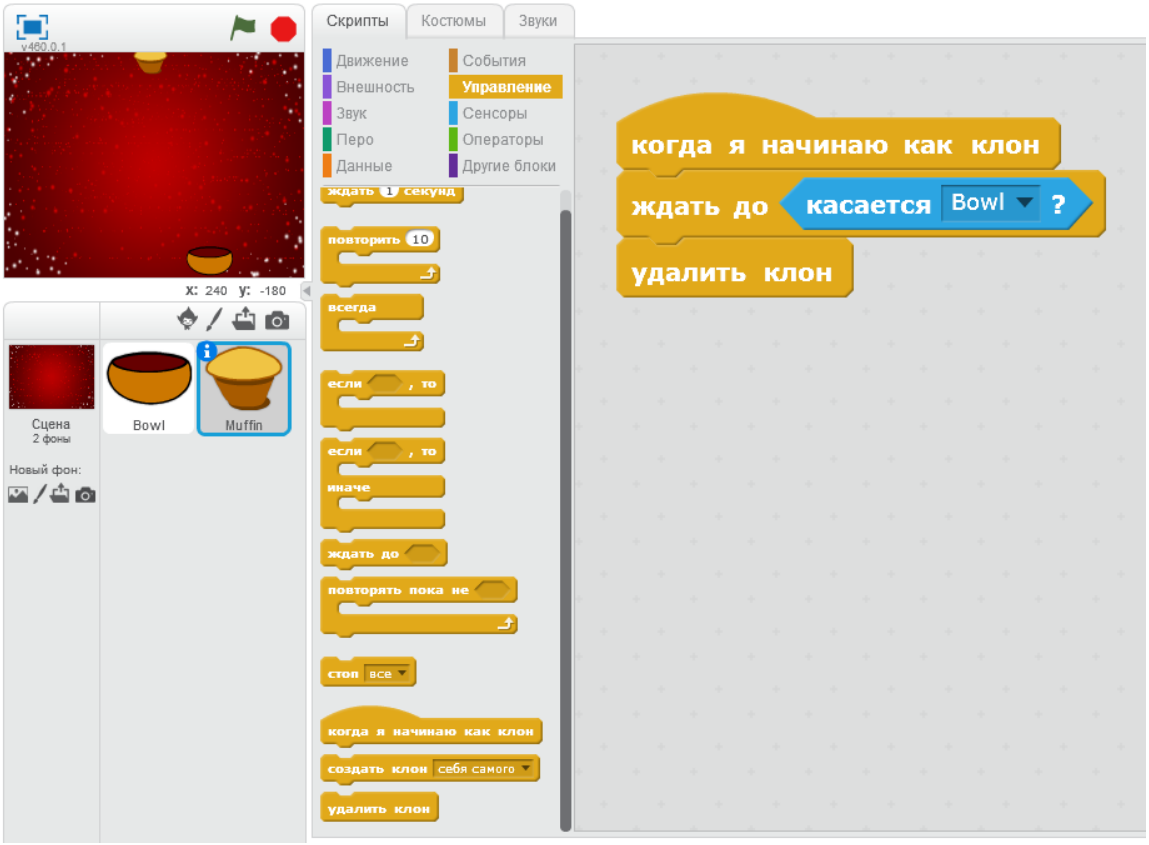
У падающего предмета программируем следующие действия:



В результате этих команд каждые 4 секунды будет появляться новый клон спрайта. Далее добавляем команды, чтобы этот клон падал вниз:



Остаётся добавить команды, отвечающие за ловлю этого объекта нижним спрайтом:

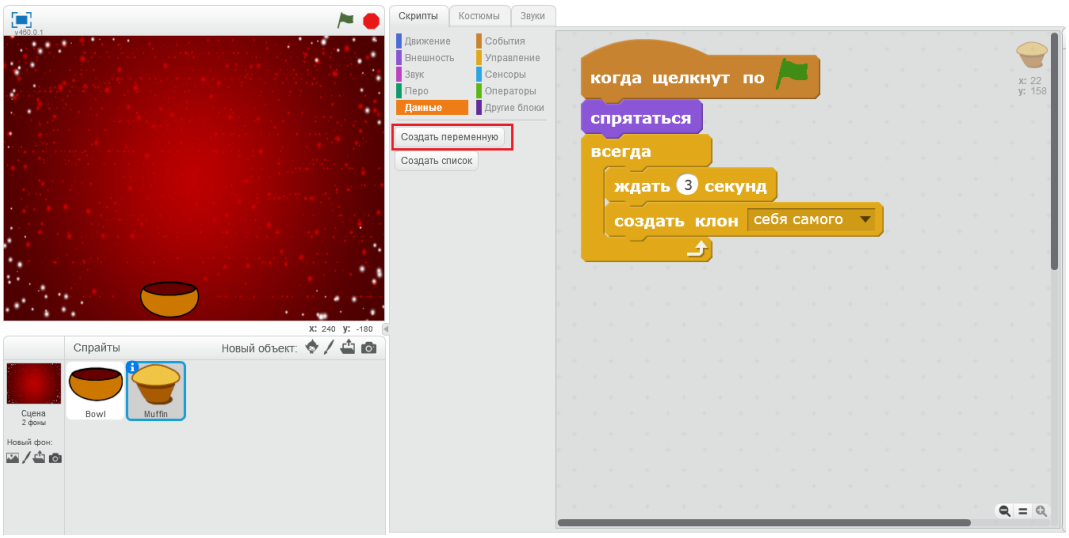


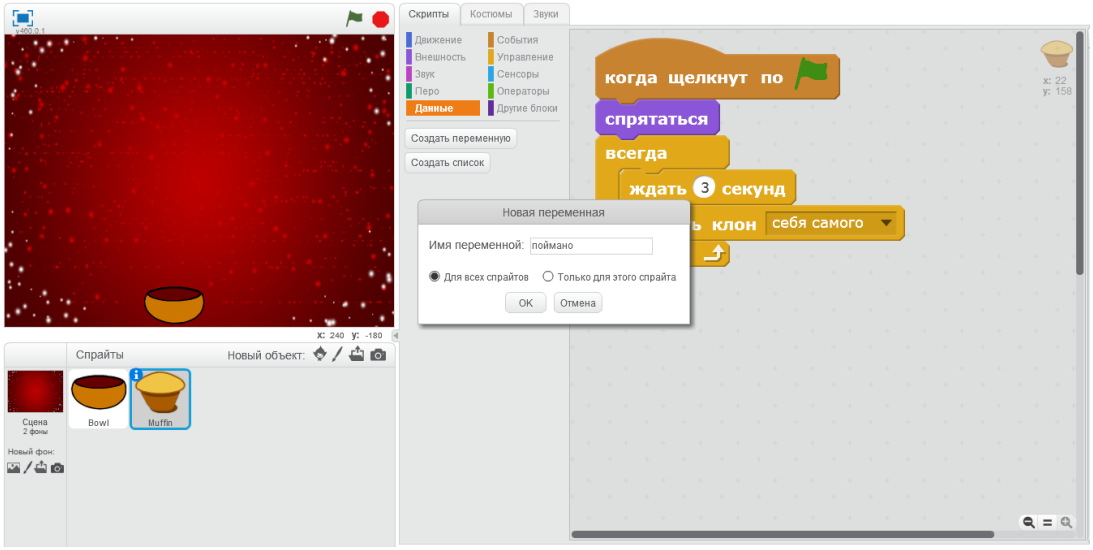
Часть 3 «Переменные»

Для создания полноценной игры осталось добавить подсчёт количества пойманных предметов. Чтобы хранить это число и впоследствии его изменять, в Scratch, как и во многих языках программирования, используются переменные.

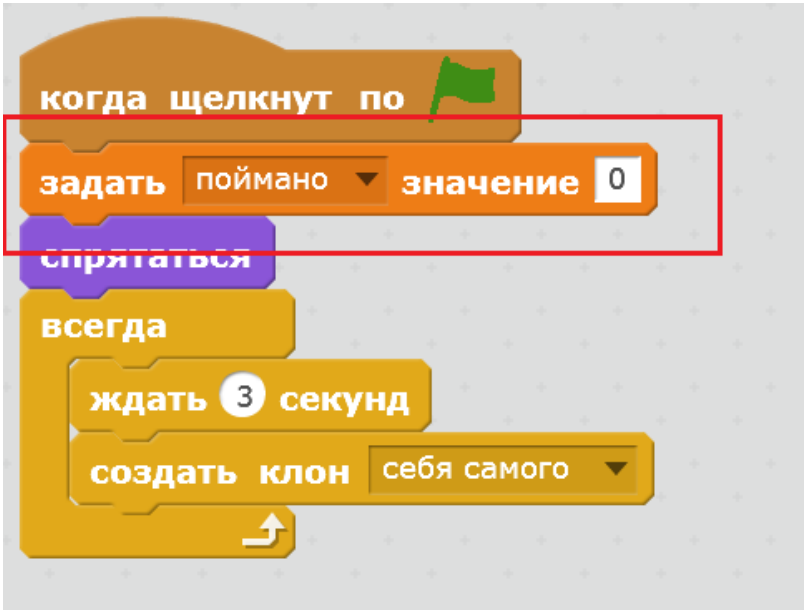
Переменная состоит из имени и значения. Значение – это и есть какая-либо сохраненная информация, чем в нашем случае является количество пойманных предметов. Через имя переменной мы обращаемся к ячейкам памяти компьютера, чтобы получить оттуда данные или изменить их.

В разделе «Данные» создаём переменную:

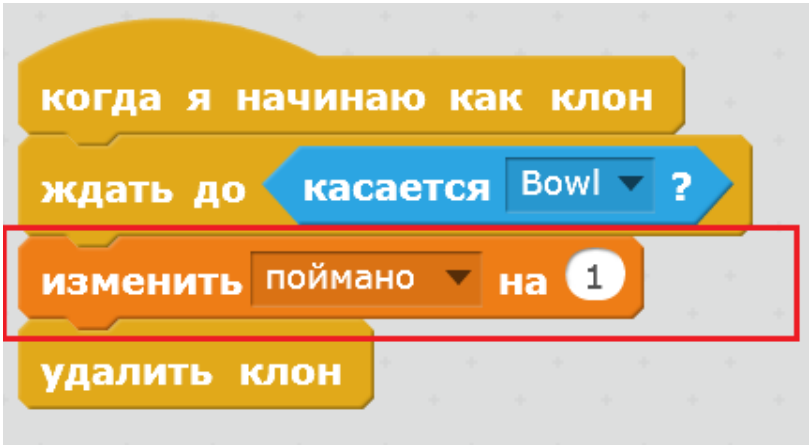




Падающему спрайту в первый блок добавляем команду:



А в последний блок:



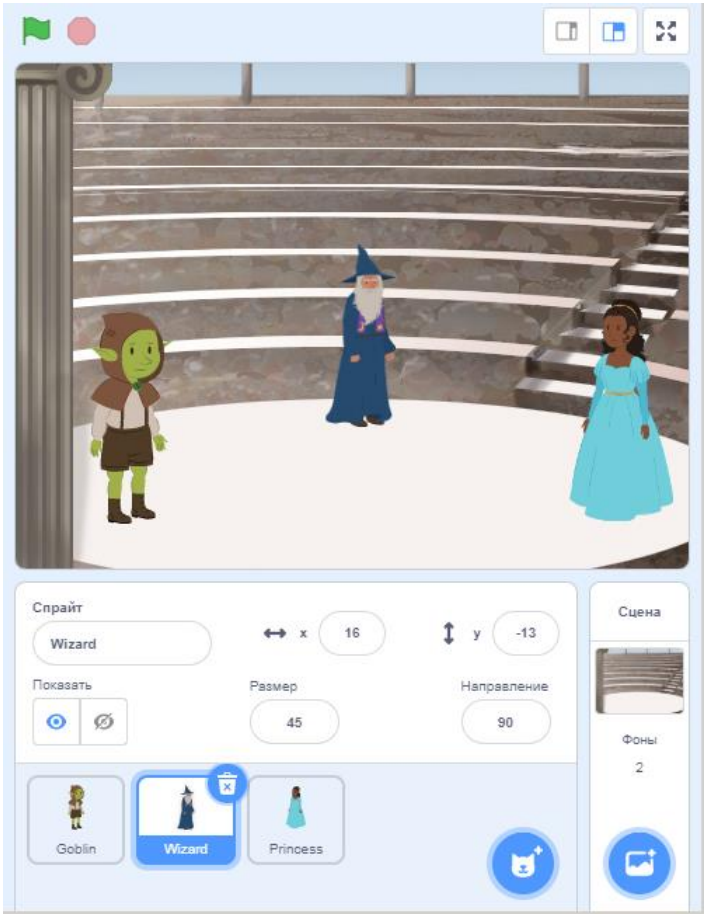
Сохранить проект под именем «ФамилияИО_Ловля предметов».

Практическая работа №4 «Ввод и обработка данных в Scratch»

Часть 1 «Подготовка игрового поля»

Выбрать произвольный фон и любых трёх персонажей, с которыми игрок будет взаимодействовать.

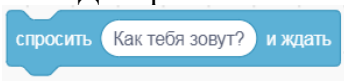
Расположить персонажей можно произвольно.



Часть 2 «Программирование первого персонажа»

Конечный результат: при клике персонаж спрашивает у игрока два числа. После этого говорит их сумму.

Для решения этой задачи в Scratch используется переменная 

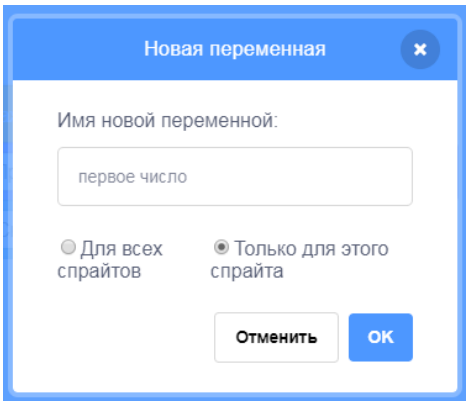
и команда 

, которые находятся в разделе «сенсоры».

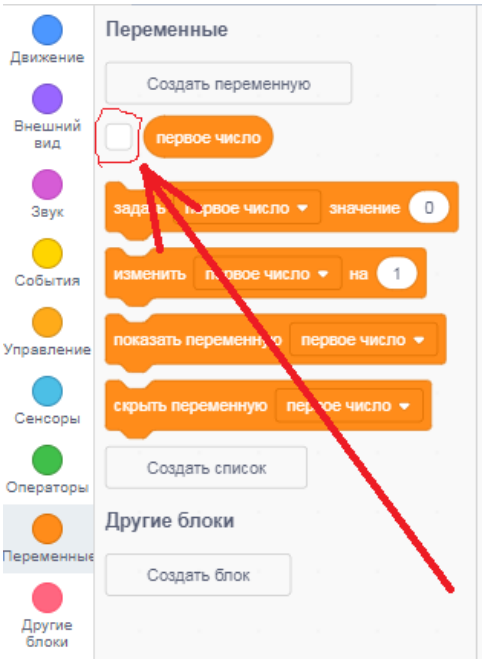
Переменная 

может запоминать только одно значение. Поэтому для хранения двух чисел нам необходимо создать ещё одну переменную. Напоминаем, что сделать это можно в разделе «Переменные».

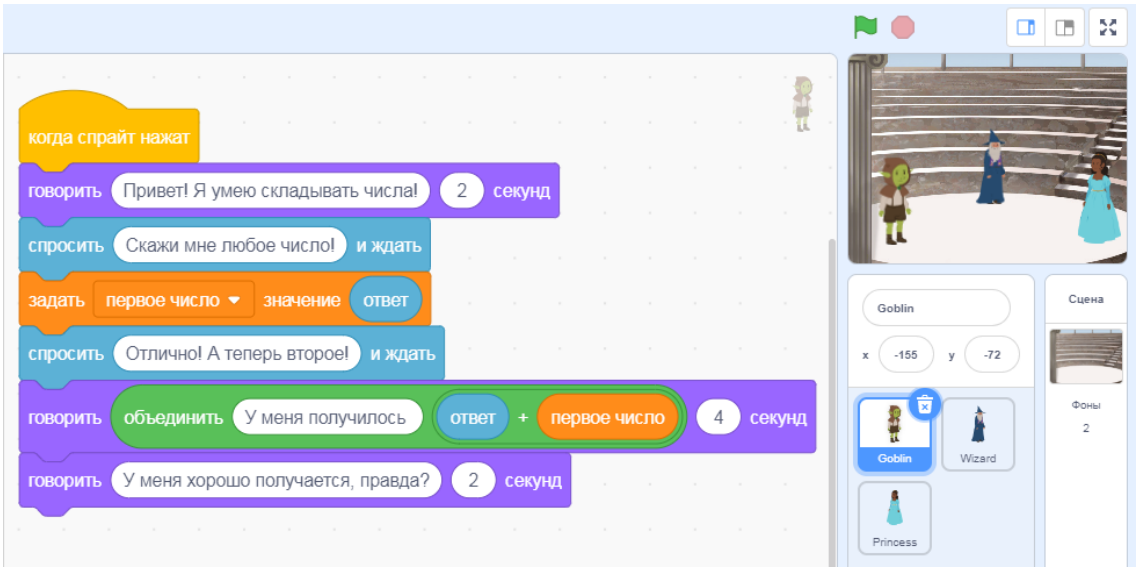
В этой работе, в отличие от прошлой, нам будет необходимо использовать несколько переменных. Поскольку использовать созданную нами переменную будет только тот персонаж, который занимается сложением чисел, необходимо выбрать пункт «Только для этого спрайта». В ином случае данная переменная будет отображаться и у других спрайтов, что будет мешать найти другие переменные.



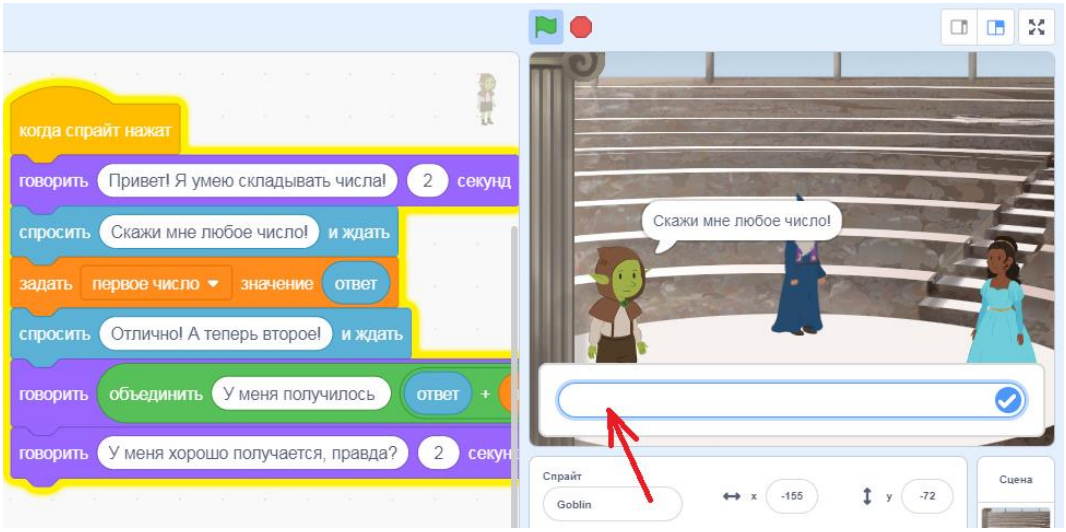
Необходимости отображать переменную на экране нет. Поэтому галочка с поля, сигнализирующая о её отображении на сцене, должна быть снята.



Программируем спрайт на выполнение вычислений:



Проверяем работу скрипта, кликнув на персонажа. Вводить число необходимо в появившееся поле для ввода.



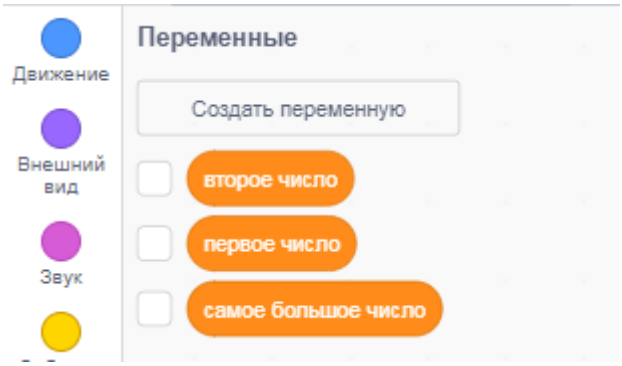
После ввода числа можно нажать «Enter» или на синюю кнопку справа. В дробных числах для разделения дробной и целой частей используется точка.

Самостоятельно замените сложение умножением.

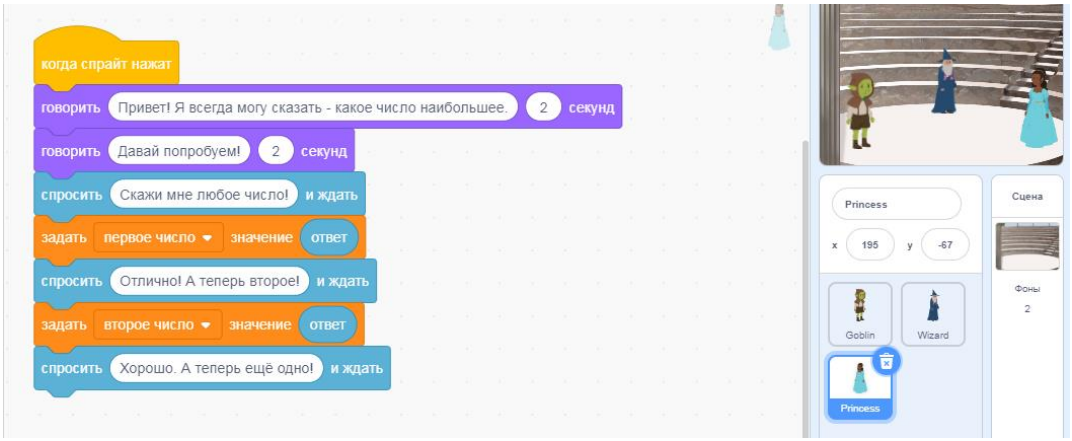
Часть 3 «Программирование второго персонажа»

Конечный результат: при клике персонаж спрашивает у игрока три числа. После этого говорит максимальное из них.

Для этого персонажа необходимо создать три переменных: «первое число», «второе число» и «самое большое число». Все переменные будут задействованы только в данном спрайте.

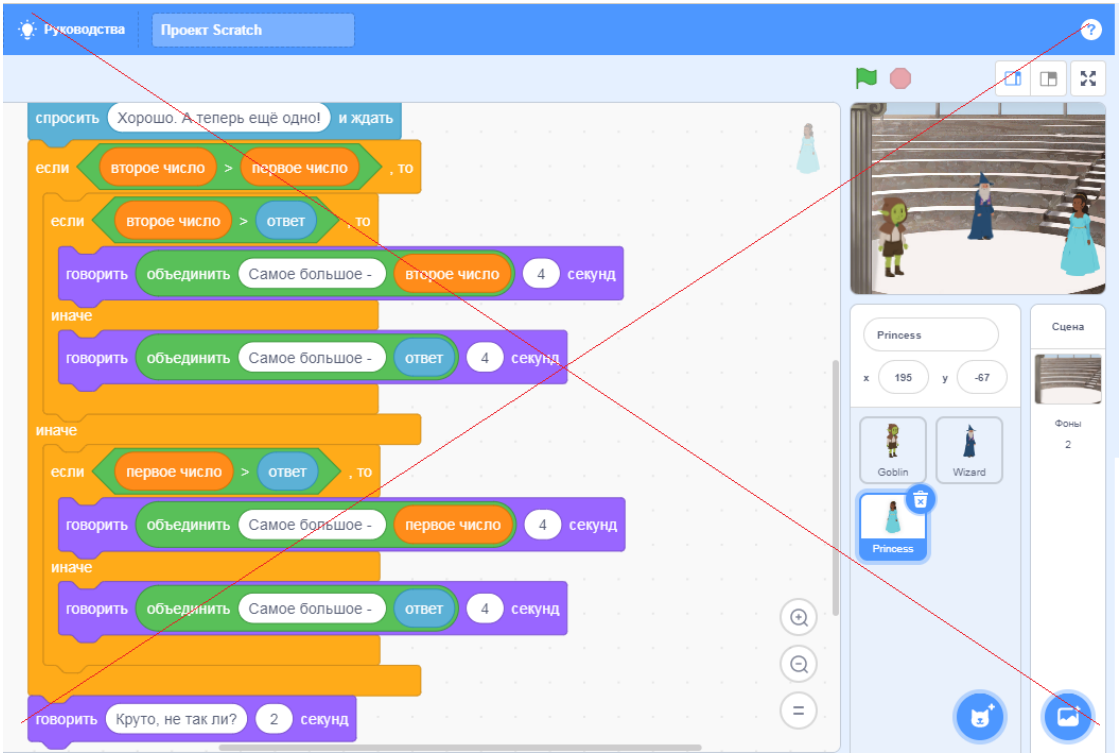


Для персонажа начало скрипта будет выглядеть следующим образом:



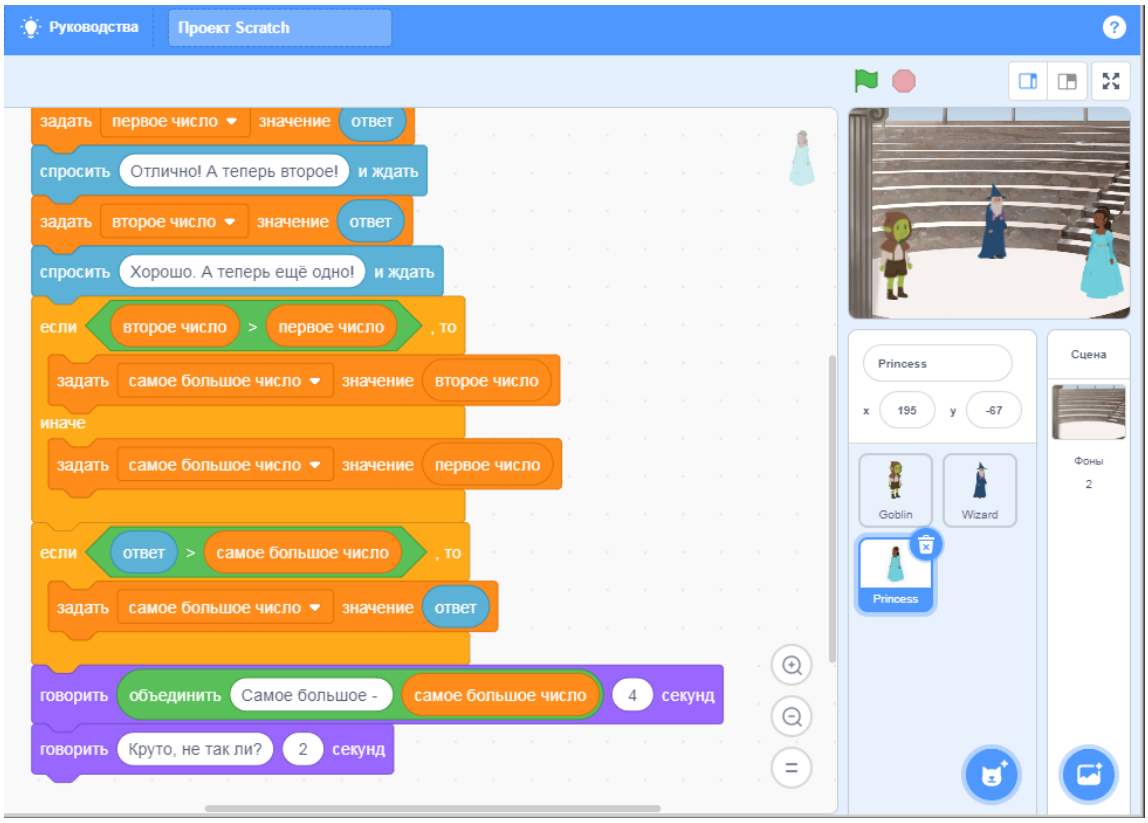
После этого у программиста есть два пути написания скрипта. Первый вариант – неоптимальный. Можно найти с использованием конструкции «если ..., то ... иначе ...» максимальное из первых двух чисел, а затем сравнивать его с третьим с помощью ещё одной конструкции. Переменная «самое большое число» в этом случае не понадобится.

Использовать этот вариант не стоит.



Второй вариант – мы ищем сначала наибольшее только из чисел «первое число» и «второе число». А затем помещаем его в переменную «самое большое число».

После этого мы забываем о существовании переменных «первое число» и «второе число», а сравниваем между собой только переменные «самое большое число» и «ответ». Максимальное из этих двух чисел и будет ответом.



До такого варианта сложнее догадаться, но именно в этом случае скрипт стоит считать написанным более оптимально. В олимпиадах по программированию и конкурсах проектов оценивается не только количество команд, но и качество программы, которая с их помощью составлена.

Часть 4 «Программирование третьего персонажа»

Конечный результат:

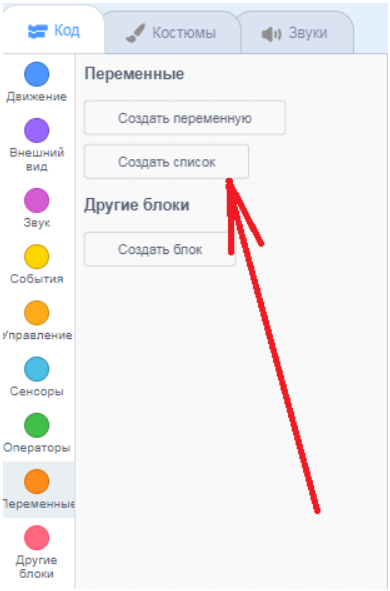
Шаг 1. При клике персонаж спрашивает у игрока количество чисел, которые он будет вводить (обозначим это значение как N).

Шаг 2. Персонаж спрашивает у игрока значение каждого из этих чисел (то есть задаст вопрос о вводе числа N раз).

Шаг 3. Персонаж даёт ответ на вопрос: сколько чисел, которые ввел пользователь на шаге 2, являются чётными.

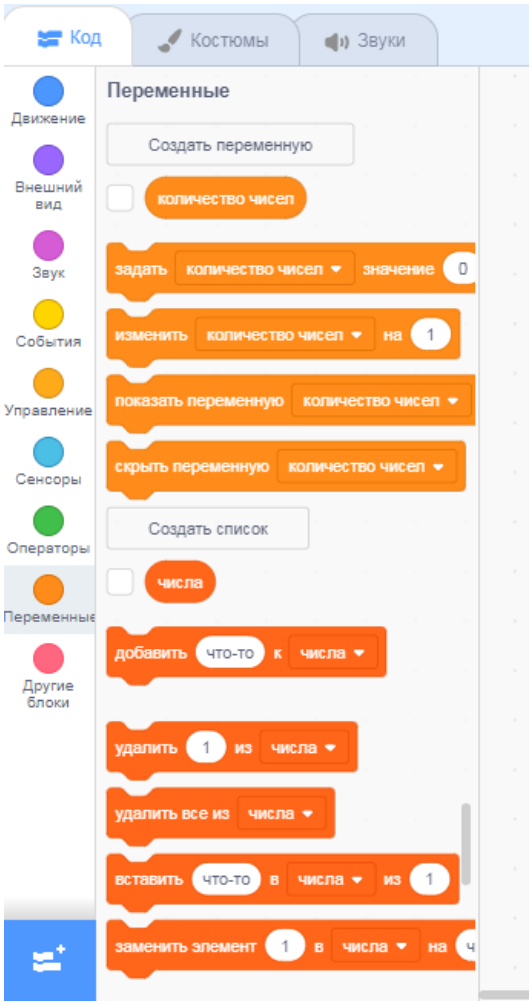
В этом скрипте мы заранее не знаем количество чисел, которые потребуется запомнить персонажу. Поэтому создать нужное количество переменных заранее не получится.

Для таких ситуаций в Scratch используется список. Кнопка для его создания находится в разделе «Переменные».

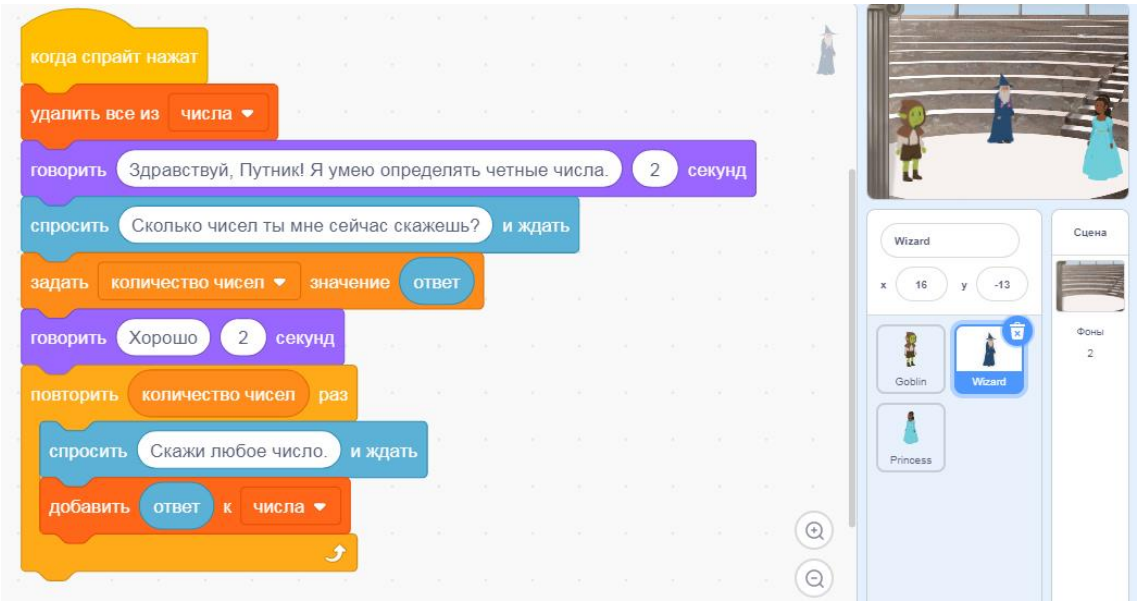


Кроме этого потребуется переменная для хранения числа, которое введет пользователь.

В итоге имеем:

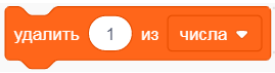


В начале скрипта, как и в предыдущих частях, будет ввод данных со стороны игрока.

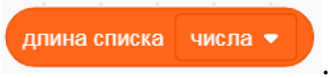


При добавлении чисел в список стоит учитывать, что в нём уже могут присутствовать числа, оставшиеся после предыдущих запусков скрипта. Именно с этой целью в качестве первой команды здесь вставлено «Удалить все из "числа"».

После этого нам необходимо проанализировать все элементы, добавленные в список и каждый из них проверить на четность. В любом списке Scratch, длина которого составляет N элементов, элементы пронумерованы от 1 до N. Используя номер элемента, можно получить из списка его значение (

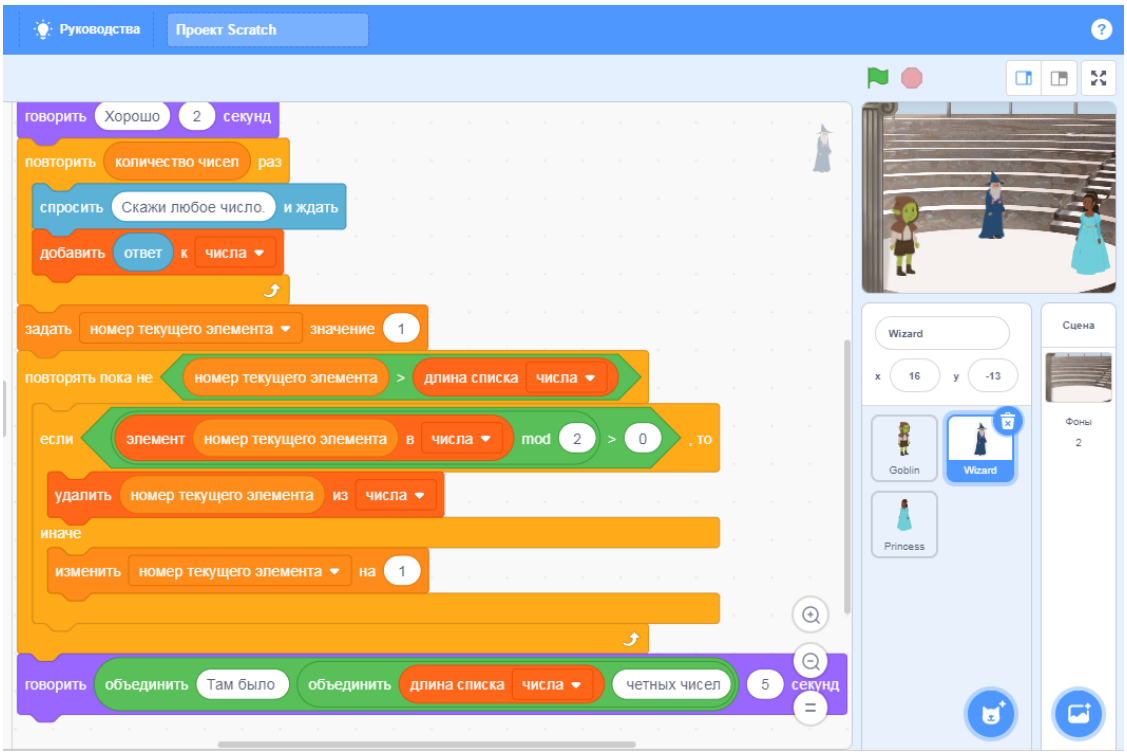
), а также удалить его (

).

В любой момент можно узнать количество элементов в списке с помощью команды 

.

Для выбора каждого элемента из списка будем использовать новую переменную «номер текущего элемента».



Оператор mod позволяет найти остаток от деления числа слева от оператора на число справа от оператора. В скрипте мы удаляем из списка те элементы, которые не являются четными (то есть их остаток от деления на 2 больше нуля).

Самостоятельно доделать проект так, чтобы в случае, если игрок ввел только нечетные числа, персонаж сообщал «Четных чисел не было».

Сохранить проект под именем «ФамилияИО_Математики».

Практическая работа №5 «Подпрограммы в Scratch»

Теоретическая часть

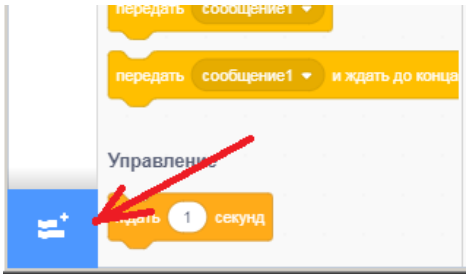
Подпрограмма — поименованная часть компьютерной программы, содержащая описание определённого набора действий. Подпрограмма может быть многократно вызвана из разных частей программы.

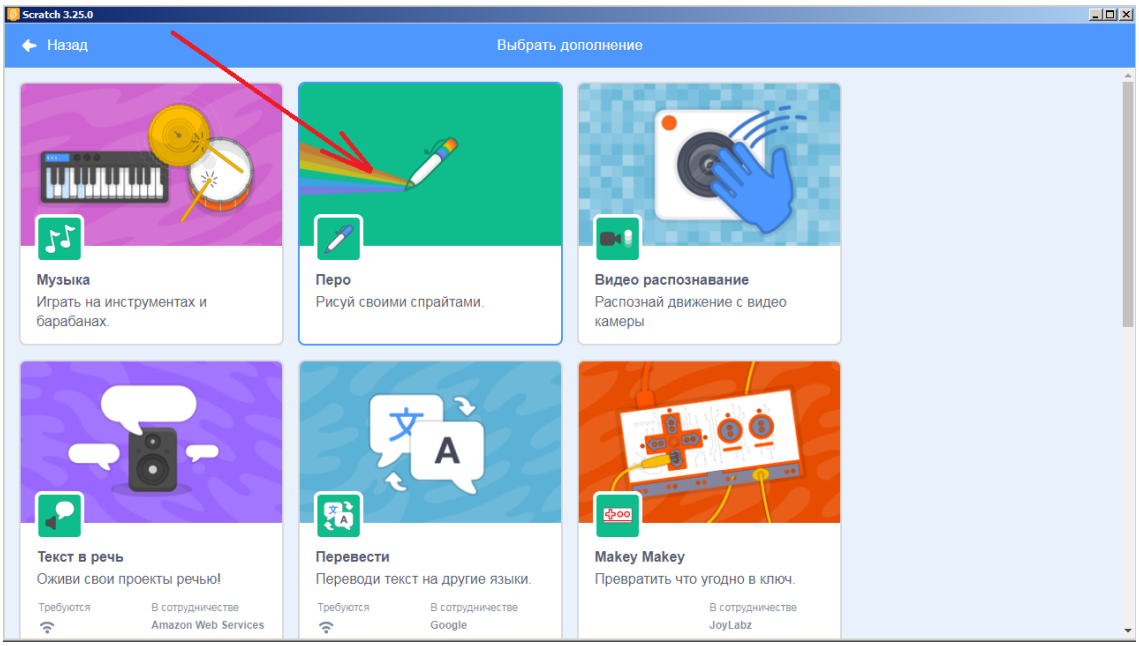
В применении к Scratch можно считать, что подпрограммой, является блок, созданный нами на основе определенного набора других имеющихся в среде блоков.

Задача: запрограммировать у кота рисование на сцене нескольких окружностей.

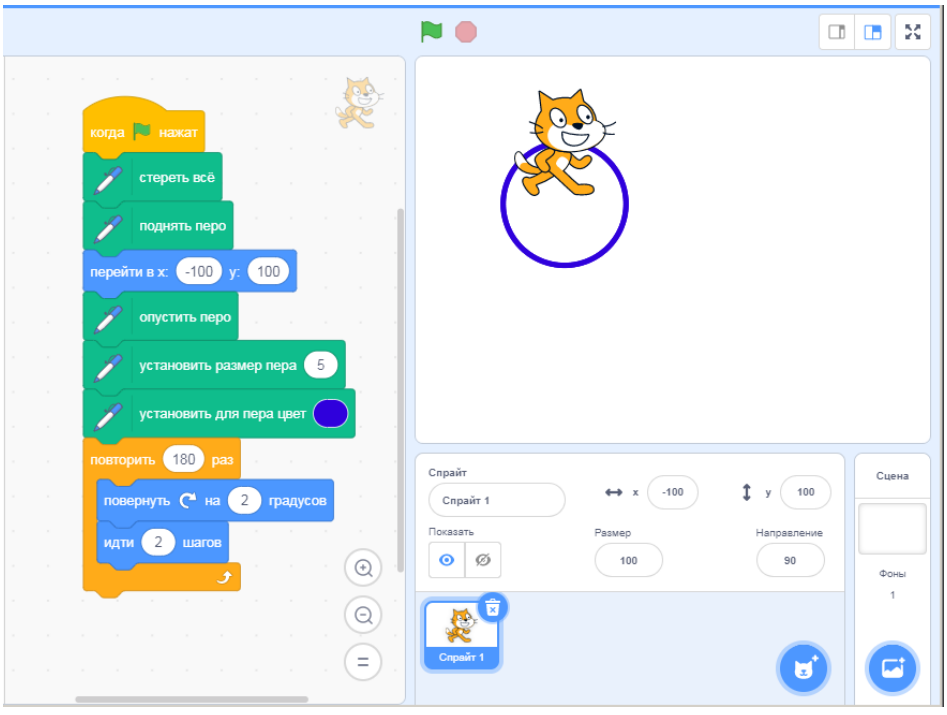
Решение:

Подключаем расширение «Перо»:

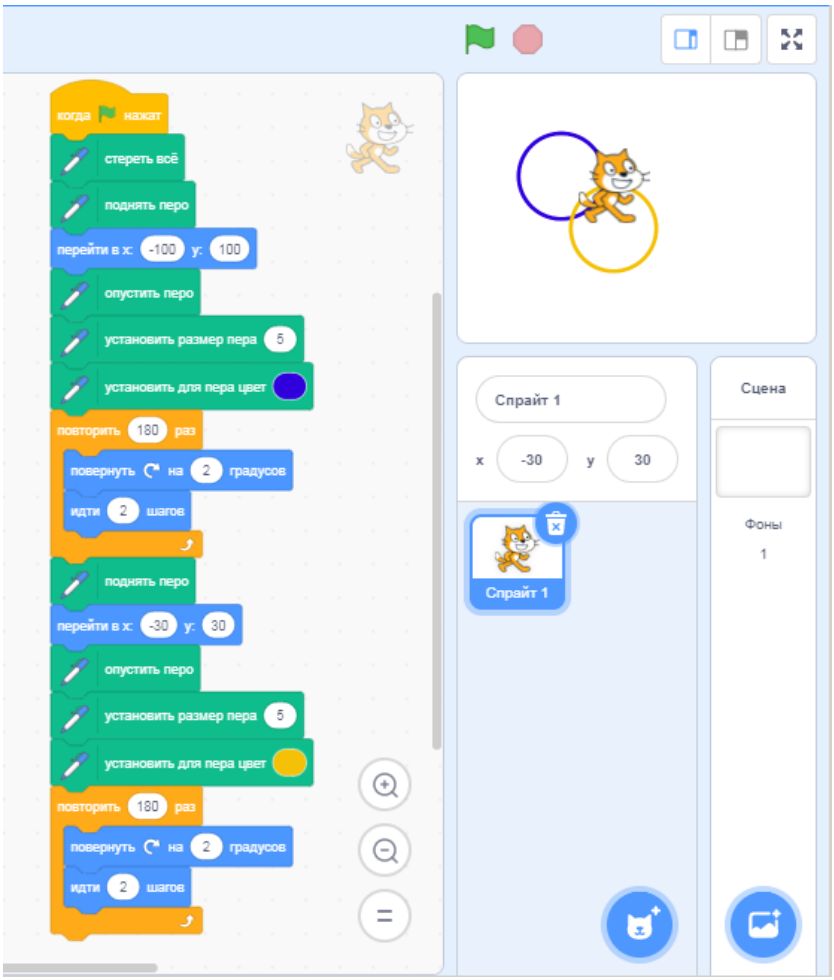




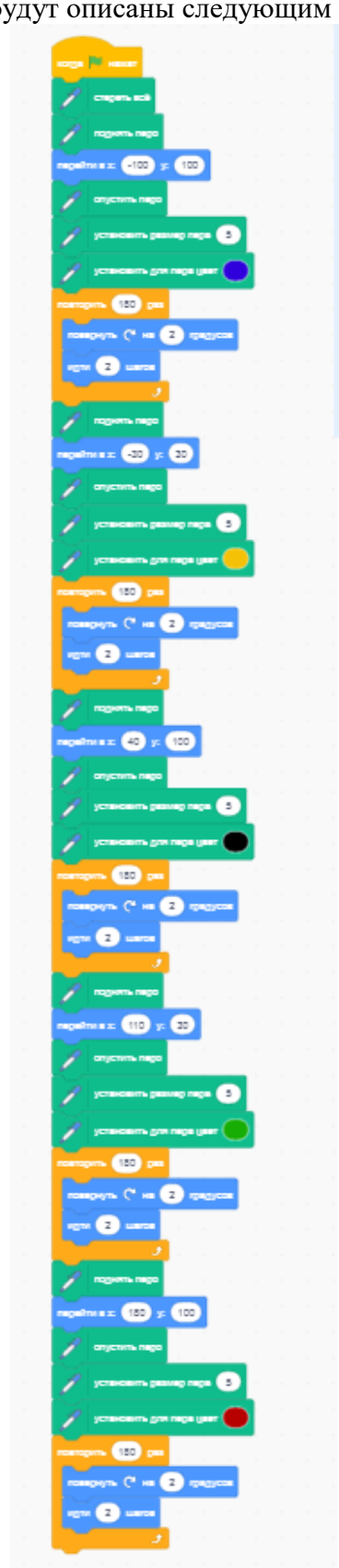
Для одного кольца программа будет выглядеть следующим образом:

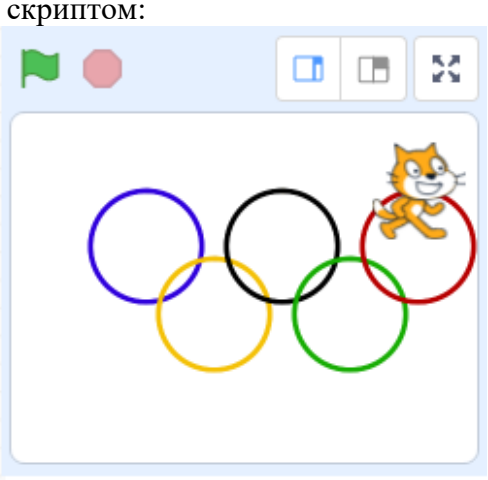


Для двух колец:



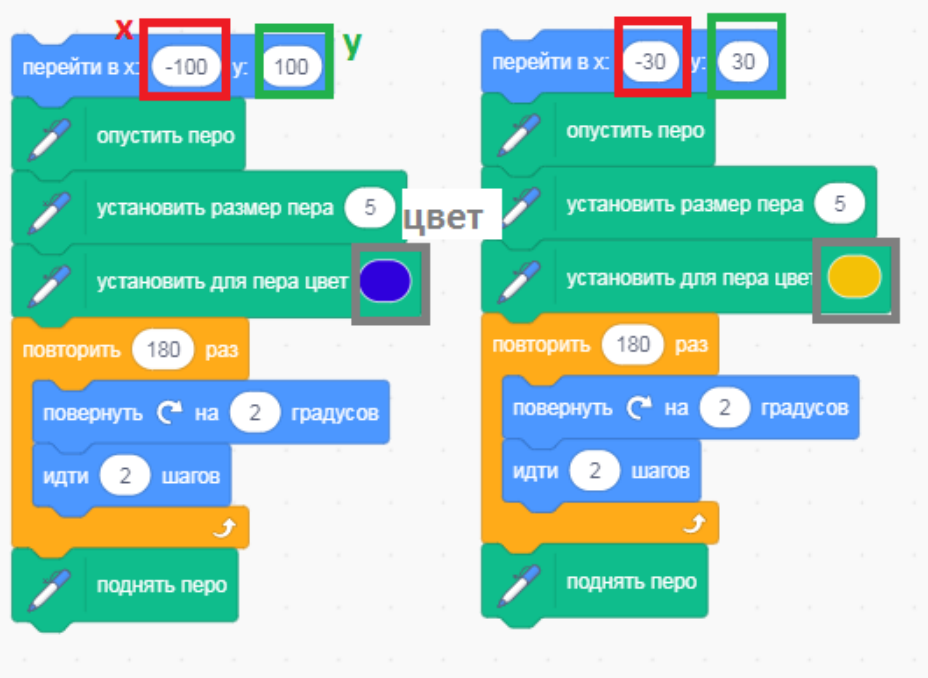
Пять колец будут описаны следующим скриптом:



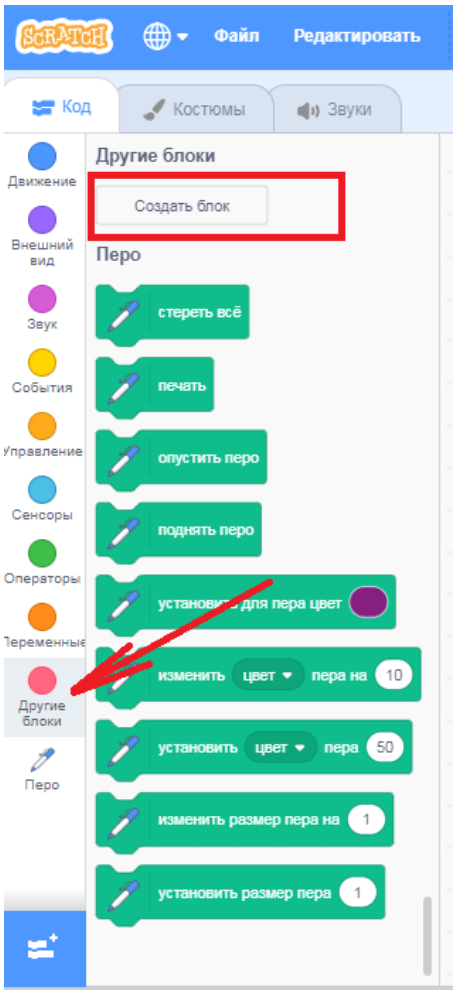


Подобный код выглядит длинным и сложен для восприятия. Упростить код можно, выделив в нём повторяющиеся элементы и оформив их в качестве подпрограмм.

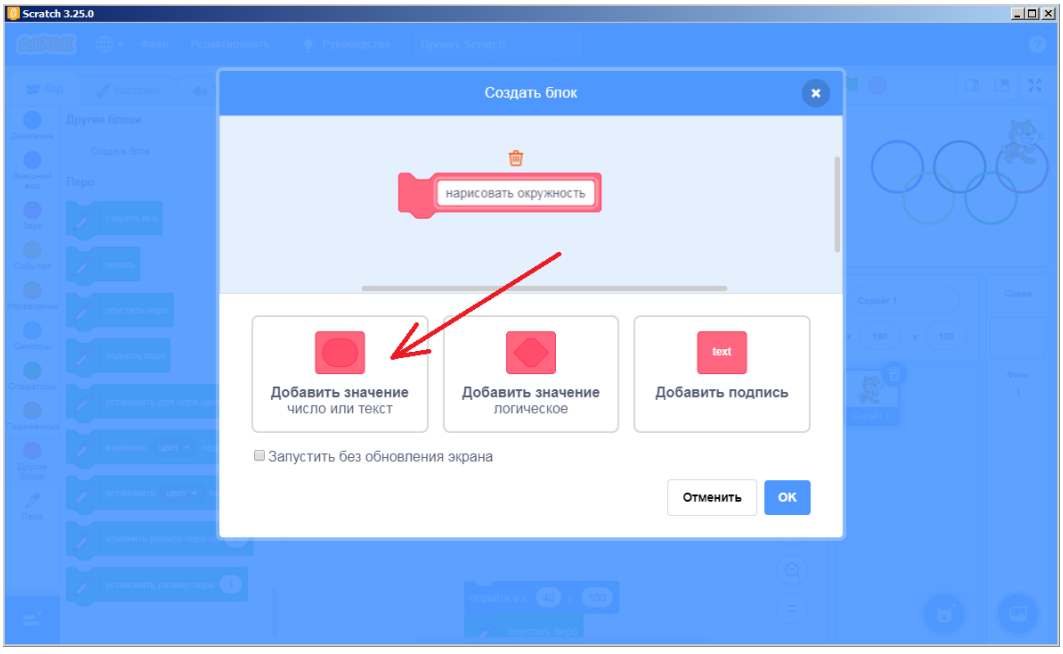
Заметим, какие отличия содержат блоки рисования круга:



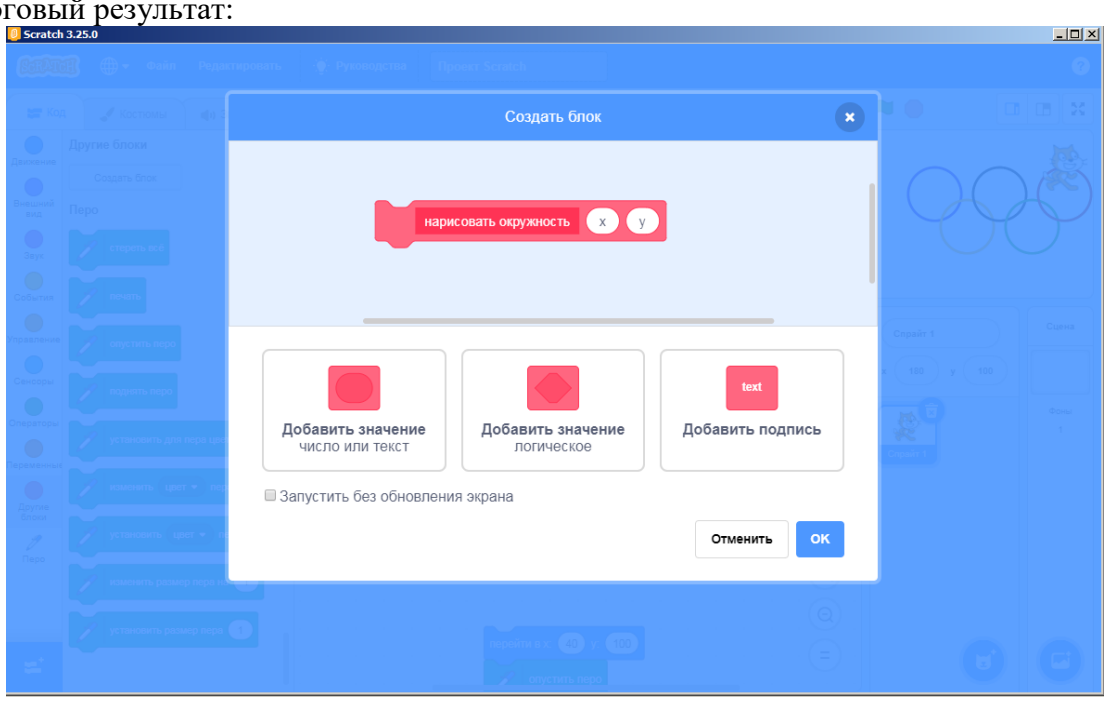
Создадим новый блок в разделе «Другие блоки»:



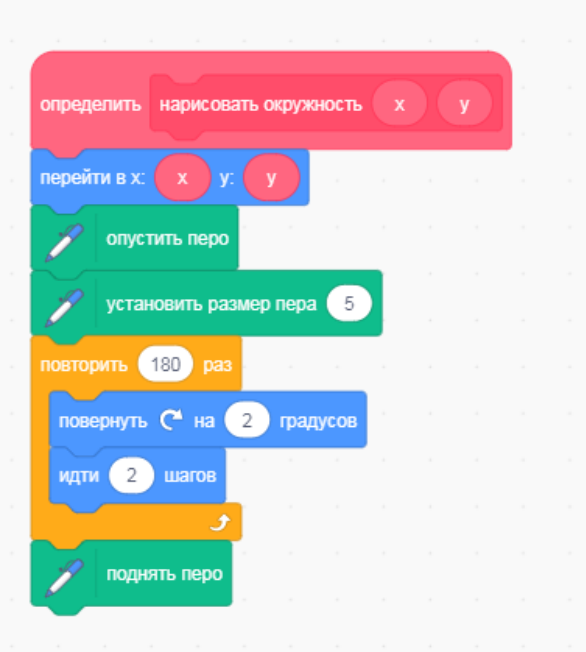
И добавим в него две переменные:



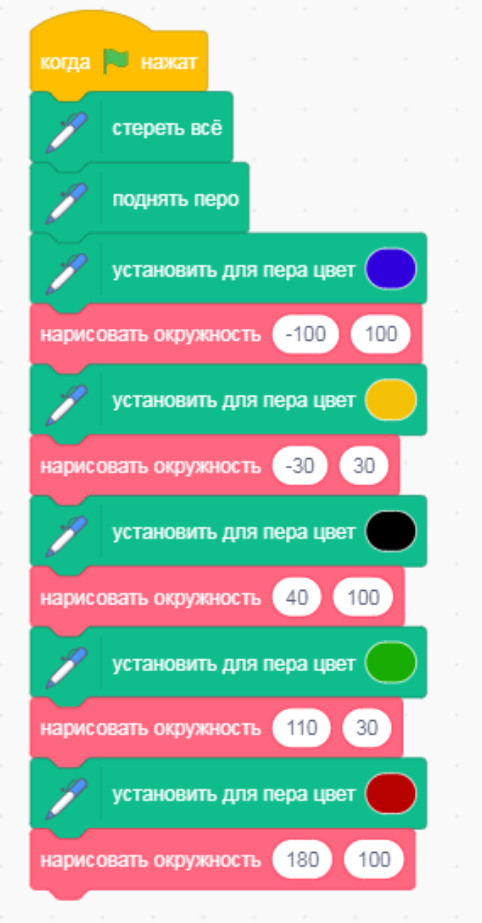
Итоговый результат:



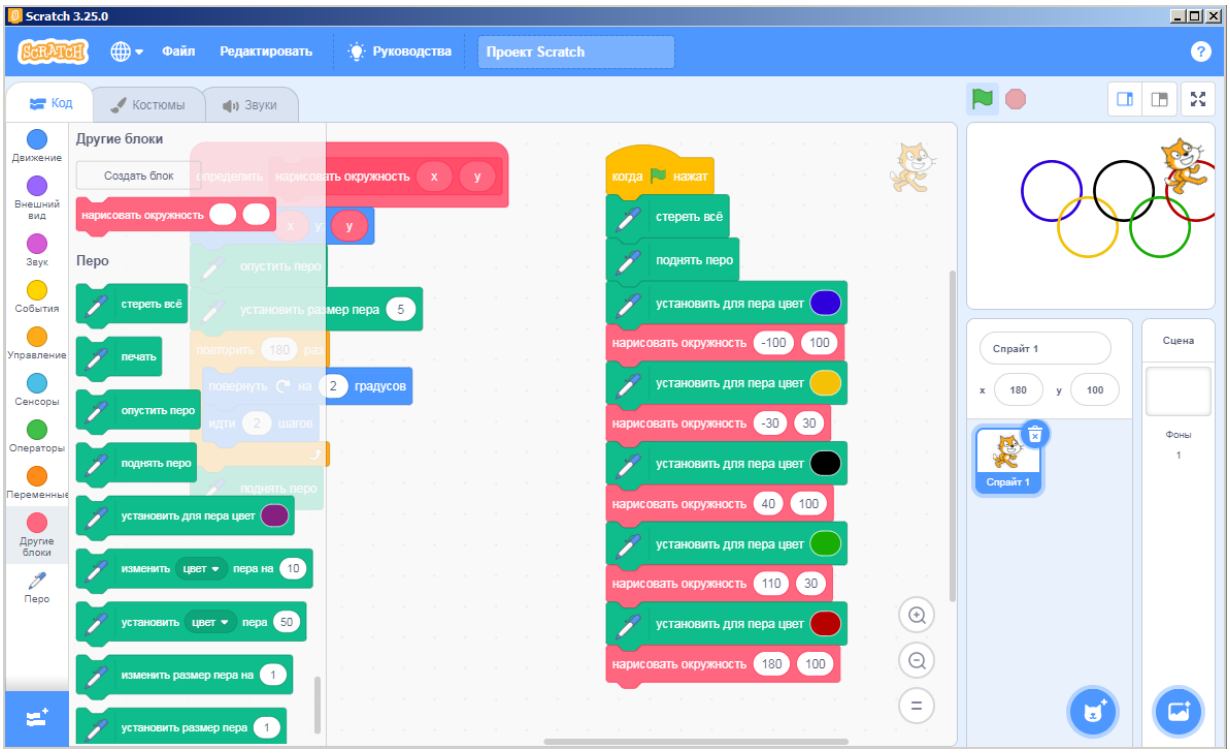
К созданному блоку переносим все команды рисования круга, за исключением команды установки цвета:



А код для персонажа переделываем следующим образом:


когда нажат
стереть всё
поднять перо
установить для пера цвет
нарисовать окружность -100 100
установить для пера цвет
нарисовать окружность -30 30
установить для пера цвет
нарисовать окружность 40 100
установить для пера цвет
нарисовать окружность 110 30
установить для пера цвет
нарисовать окружность 180 100

Итог:

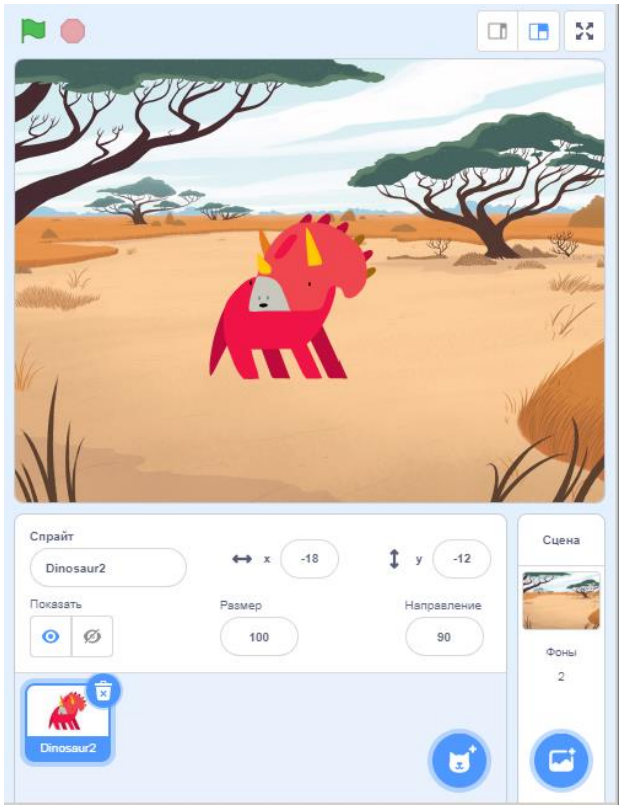

Scratch 3.25.0
Файл
Редактировать
Руководства
Проект Scratch
Код
Костюмы
Звуки
Другие блоки
Перо
Спрайт 1
Сцена

Таким образом, нам удалось существенно упростить программный код за счёт использования собственных блоков (подпрограмм).

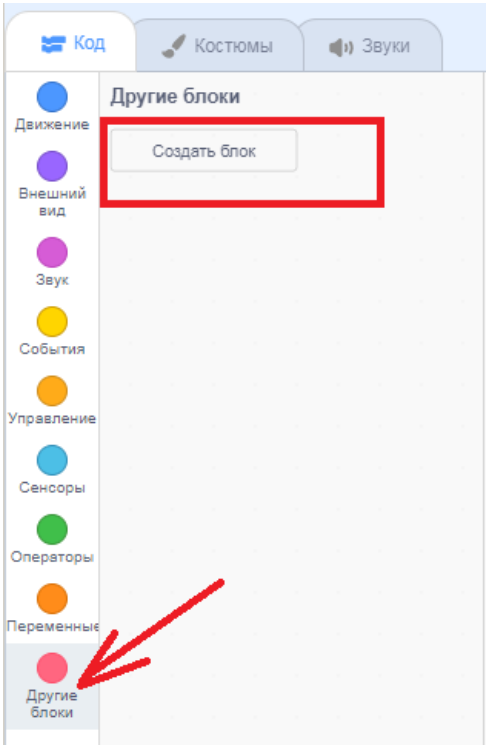
Задание для выполнения №1 (практическая часть)

Цель: создать игру, где персонаж задаёт игроку вопросы и проверяет правильность ответа на них.

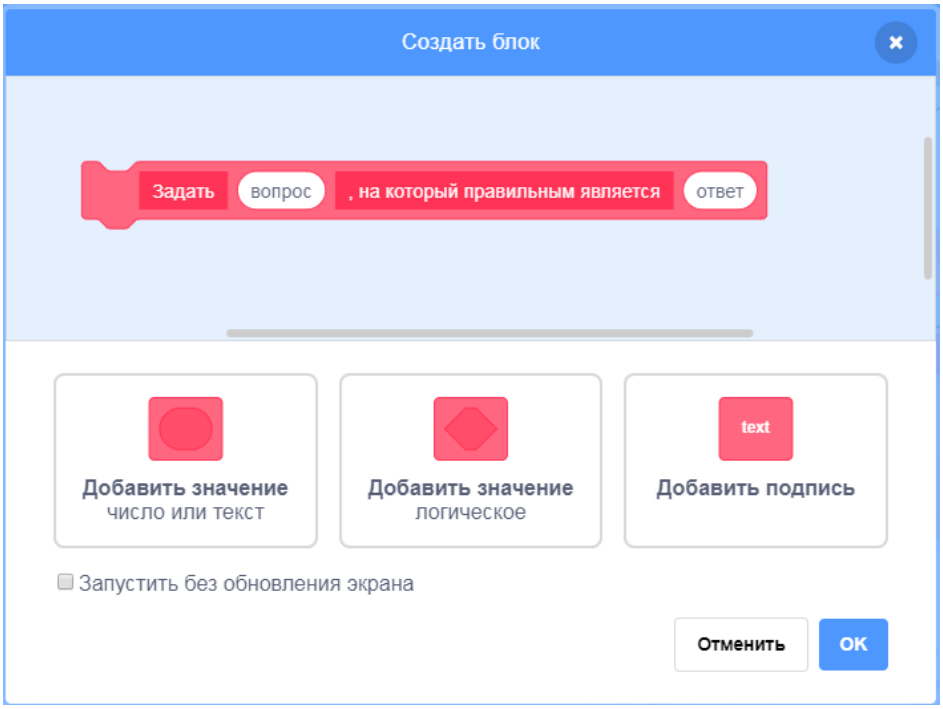
Шаг №1: Подготовка игрового поля.
В проекте можно использовать любой фон и любого персонажа.



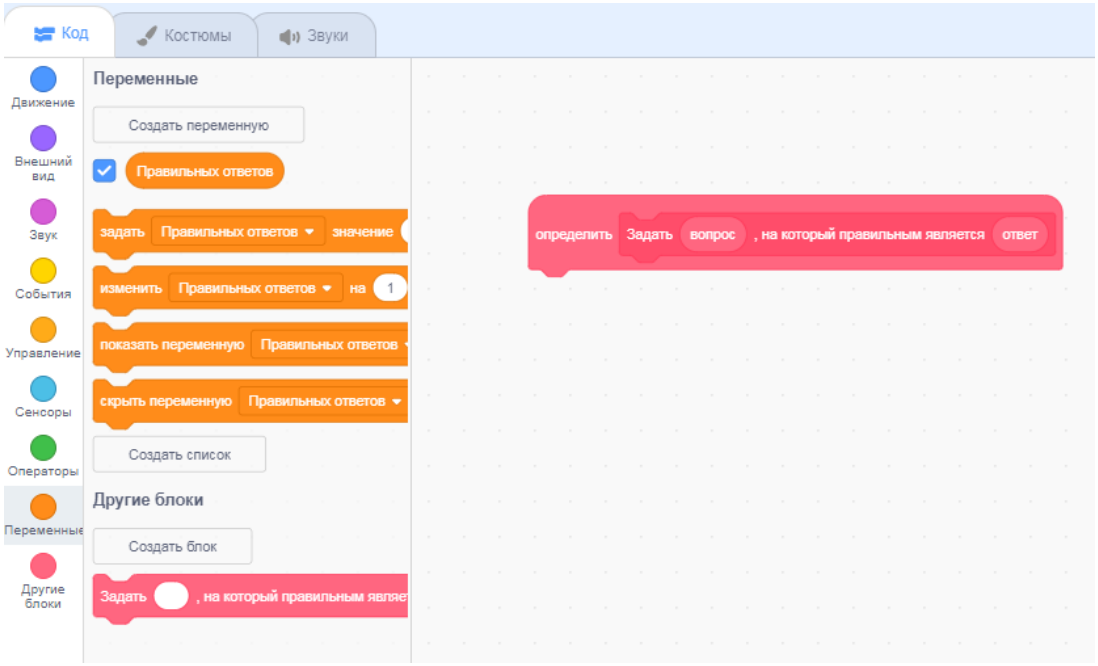
Шаг №2: Создаём новый блок:



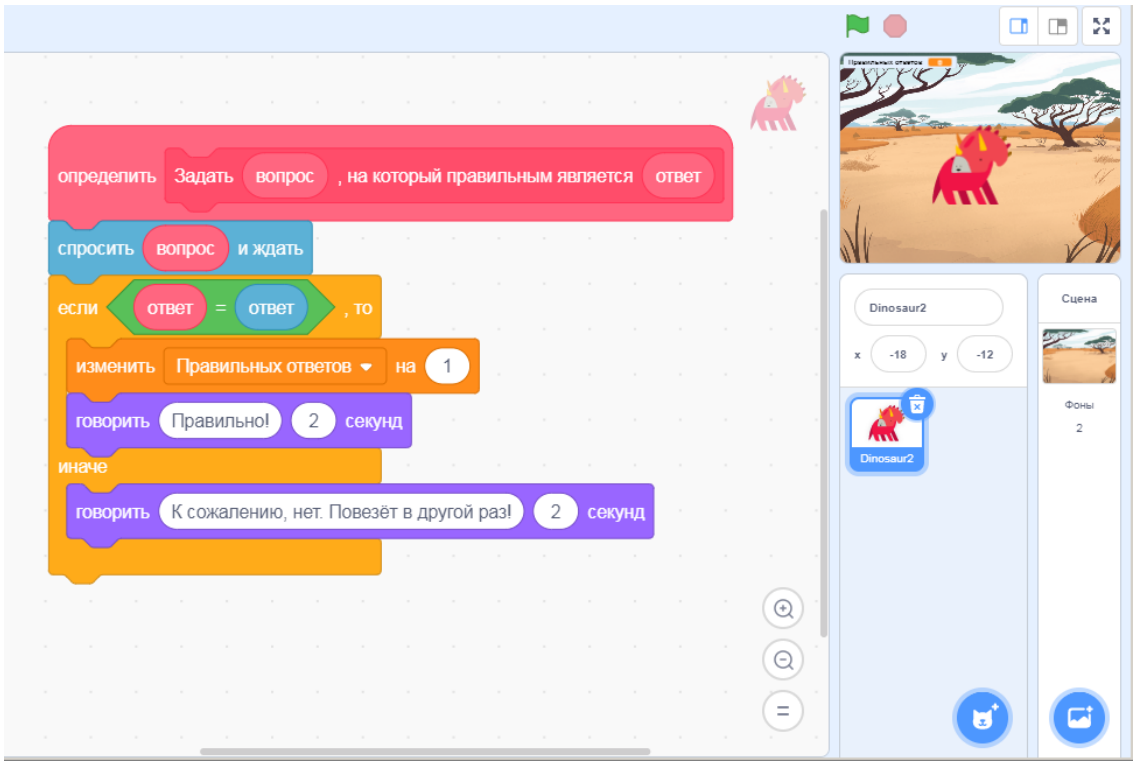
При этом следует указать следующие настройки:



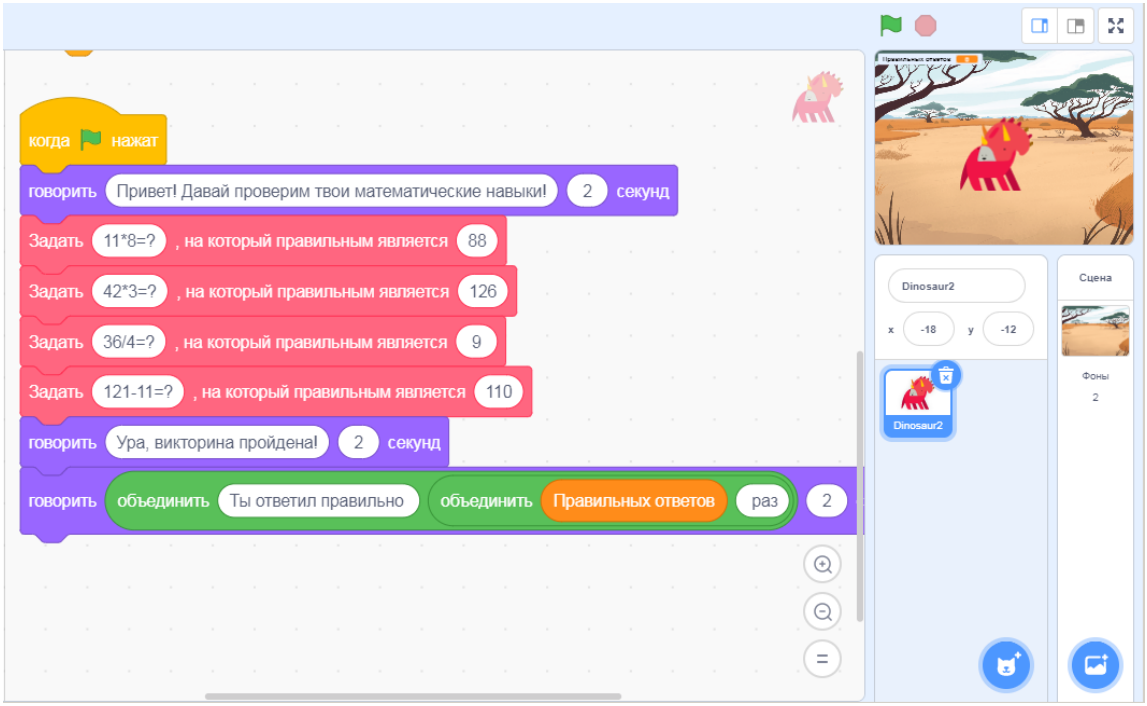
Кроме того, необходимо создать переменную «Правильных ответов».



Шаг №3: Код созданного блока описываем следующим образом:



Шаг №4: Для персонажа программируем следующий скрипт:



Вопросы и тему викторины меняем в соответствии со своими предпочтениями.

Сохранить проект под именем «ФамилияИО_Подпрограммы».

Задание для выполнения №2 (педагогическая часть)

По ссылкам представлены проекты, созданные в среде программирования Scratch. Представьте, что вы стали экспертом данного конкурса и вы должны оценить данные проекты.

№ проекта	Название проекта	Ссылка на проект
1	Виселица	https://scratch.mit.edu/projects/203488286/
2		https://scratch.mit.edu/projects/619110893/
3	Лабиринт	https://scratch.mit.edu/projects/121306341/
4		https://scratch.mit.edu/projects/151257926/
5	Гонки	https://scratch.mit.edu/projects/485095815/
6	Крестики-нолики	https://scratch.mit.edu/projects/137552069/
7	Ракета	https://scratch.mit.edu/projects/389732391/
8	Ловля предмета	https://scratch.mit.edu/projects/185698890/

9	Разбивание кирпичей	https://scratch.mit.edu/projects/163443000/
10	Платформер	https://scratch.mit.edu/projects/160857714/

Наиболее распространенными критериями оценки проектов в Scratch являются:

- Оригинальность идеи (К1);
- Качество программного кода (К2);
- Сложность и объем функционала проекта (К3);
- Устойчивость к действиям пользователя (например, отсутствие ошибок при некорректном вводе) (К4);
- Дизайн проекта и использование оригинальных графических изображений (К5).

В качестве ответа на задание рекомендуется в файле Microsoft Word оформить таблицу:

№ проекта	**Оценка по критериям**					**Суммарно**	**Место**	**Комментарий**
	К1	**К2**	**К3**	**К4**	**К5**			
Max баллов								
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								

В этой таблице:

- Оценка по критериям – баллы по каждому из критериев, перечисленных выше;
- Суммарно – общее количество баллов;
- Место – положение участника в турнирной таблице при ранжировании по количеству набранных баллов;
- Комментарий – общее впечатление от проекта и проделанной работы участником;
- Max баллов – строка для информирования участников о максимальном количестве баллов в сумме и по каждому из критериев.

Практическая работа №6 «Реализация линейных, разветвляющихся и циклических алгоритмов в Кумир»

1. Напишите программу, которая вычисляет периметр прямоугольника. Протестируйте ее работу на своих примерах.

2. Напишите программу, которая вычисляет периметр треугольника.

3. Напишите программу, которая вычисляет сумму цифр введенного трехзначного числа.

4. Напишите программу, которая меняет местами первую и третью цифры введенного трехзначного числа.

5. По введенным трем положительным числам определить, можно ли построить треугольник с этими сторонами.

6. По введенным трем положительным числам определить, будут ли эти три числа составлять стороны прямоугольного треугольника.

7. Написать программу для решения линейного уравнения вида **ax=b**. Проверить решение на числах: а) 0 и 0; б) 1 и 0; в) 10 и 15; г) 2 и 10. Объяснить подбор тестовых примеров.

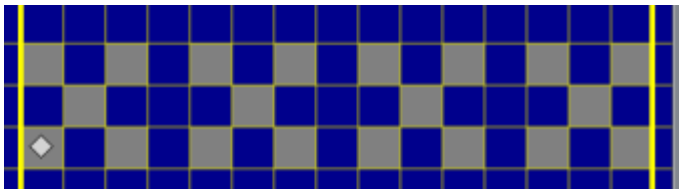
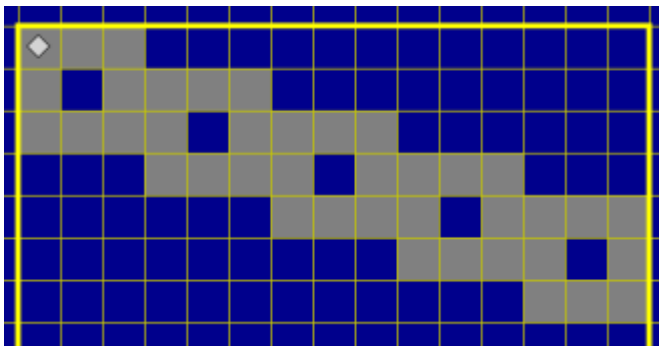
8. При помощи программы с циклом найти сумму квадратов чисел от 1 до 5.

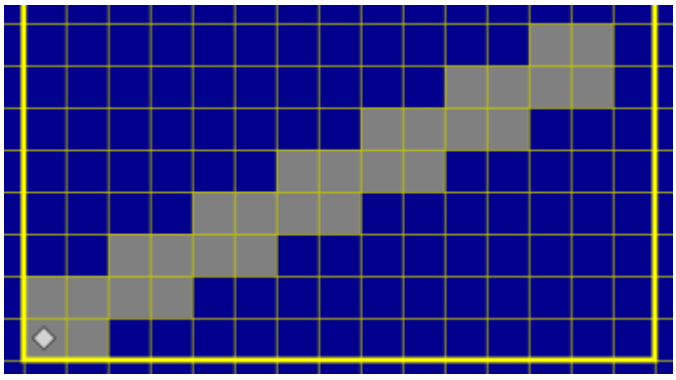
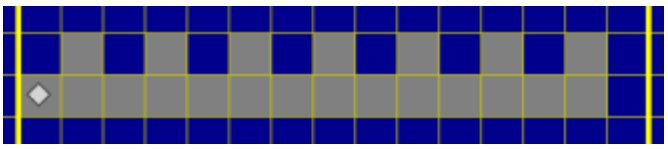
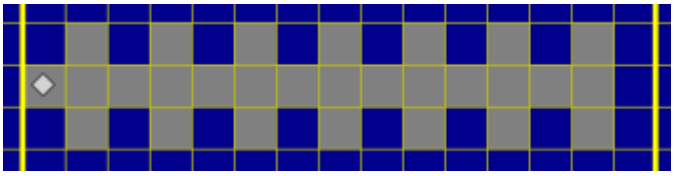
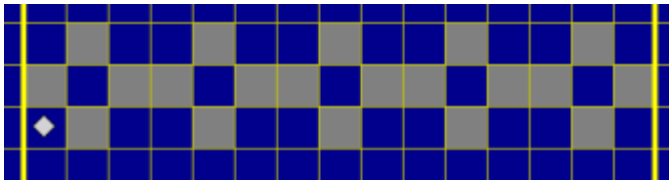
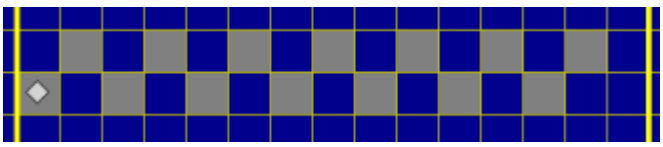
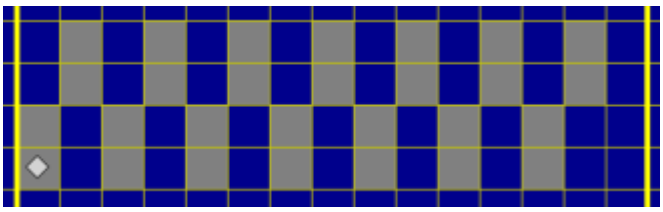
9. При помощи программы с циклом найти сумму дробей: 1+1/2+1/3+1/4+1/5.

10. Напишите программу, которая определяет количество делителей введенного числа ***a***.

Практическая работа №7 «Исполнитель "Робот" в среде программирования Кумир»

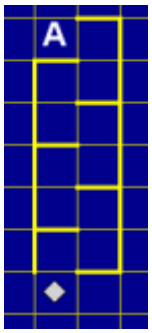
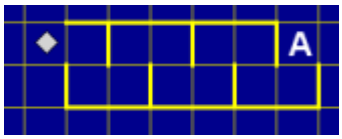
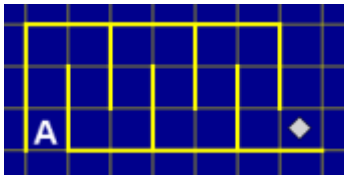
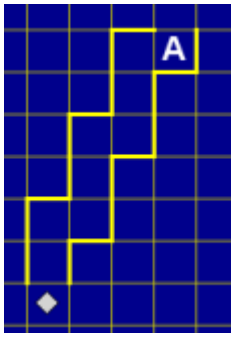
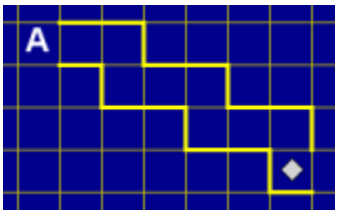
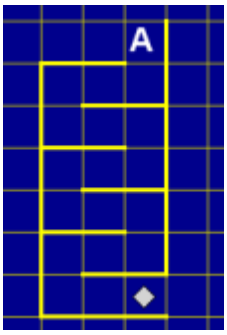
Задание №1: Составьте программу рисования узора. Начальное положение Робота отмечено символом ◊.

1	
2	

3	
4	
5	
6	
7	
8	

В качестве ответа представить файлы с кодом (.kum).

Задание №2: Необходимо провести Робота по лабиринту из начального положения (◊) в точку А.

1		2	
3		4	
5		6	

В качестве ответа представить файлы с кодом (.kum) и стартовой позицией робота (.fil).

Практическая работа №8 «Массивы в Кумир»

Массив — это пронумерованная последовательность величин одинакового типа, обозначаемая одним именем. Элементы массива располагаются в последовательных ячейках памяти, обозначаются именем массива и индексом. Каждое из значений, составляющих массив, называется его компонентой (или элементом массива).

С использованием имени массива можно ссылаться как на массив данных в целом, так и на любую из его компонент. Например, A[7] —элемент массива A с индексом 7; D[6] — элемент массива D с индексом 6.

В среде КуМир массив объявляют как `целтаб a[1:100]` (в квадратных скобках указаны индекс первого и последнего элементов массива, а `целтаб` означает, что числа в массиве будут целые).

Пример задачи №1: Дан целочисленный массив из N элементов. Увеличить все элементы в 2 раза.

Решение в среде КуМир:

```
алг пример_10_1
нач
  целтаб a[1:100], цел пер_цикла, кол_элем
  вывод "Введите количество элементов массива"
  ввод кол_элем
  вывод "Введите элементы:"
  нц для пер_цикла от 1 до кол_элем
    ввод a[пер_цикла]
  кц
  вывод "Полученный массив:"
  нц для пер_цикла от 1 до кол_элем
    a[пер_цикла] := a[пер_цикла] * 2
    вывод a[пер_цикла]
    вывод нс
  кц
кон
```

Пример задачи №2: Дан целочисленный массив из N элементов. Найти максимальный элемент массива.

Решение в среде КуМир:

```
алг пример_10_2
нач
  целтаб a[1:100], цел пер_цикла, кол_элем, макс
  вывод "Введите количество элементов массива"
  ввод кол_элем
  вывод "Введите элементы:"
  нц для пер_цикла от 1 до кол_элем
    ввод a[пер_цикла]
  кц
  | Предположим, что первый элемент
  | является максимальным
  макс := a[1]
  | Сравниваем его с остальными элементами
  нц для пер_цикла от 2 до кол_элем
    если a[пер_цикла] > макс то
      макс := a[пер_цикла]
    все
  кц
  вывод "Максимальный элемент: ", макс
кон
```

Решить задачи:

1. Дан целочисленный массив. Возвести в квадрат отрицательные элементы.
2. Дан целочисленный массив. Найти максимум среди четных элементов массива.
3. Создать одномерный целочисленный массив, состоящий из n случайных чисел в диапазоне от -50 до 50. Заменить все элементы массива на противоположные по знаку.

4. Создать одномерный целочисленный массив, состоящий из n случайных чисел в диапазоне от 1 до 20. Найти номера четных элементов, стоящих на нечетных местах.

Практическая работа №9 «Символьные величины в Кумир»

Часто при программировании надо решать не только вычислительные задачи, но и обрабатывать текстовую информацию. Например, требуется найти слово или часть его, сформировать из нескольких отдельных слов предложение или узнать длину текста. Для этого используют символьные и литерные величины.

Символьные величины при написании программ уже встречались ранее для ввода или вывода соответствующих значений. Однако с символьными величинами можно производить много полезных и интересных действий.

Разберем некоторые операции, которые можно производить с символьными данными.

Первая операция похожа на самое простое арифметическое действие – сложение и обозначается знаком « + ». Это операция **сцепление**. В результате сцепления к символьной величине, находящейся слева от знака « + », будет присоединена символьная величина, находящаяся справа от этого знака. Результат такого действия может быть присвоен только величине типа **лит**, которую **нужно** задать в разделе описания переменных.

Например, в результате действия следующего фрагмента программы, в которой все величины имеют тип литерный.

```
x := "пар"
y := "воз"
z := x + "о" + y
вывод z
```

величина z получит значение "паровоз" и соответствующее слово появится на экране.

Разберем пример:

```
алг
нач                          | Раздел описания переменных,
  лит x, y, z                | x,y,z имеют литерный тип данных
  x := "план"                | Присвоение переменной x значения
  y := "выполнен"            | Присвоение переменной y значения
  z := x + "не" + y          | Выполнение операции сцепления
  вывод z                    | Вывод значения литерной переменной z
кон                          | Конец алгоритма
```

В результате работы алгоритма переменная величина z получит значение "план не выполнен" и соответствующее словосочетание появится на экране.

С выведенной фразой что-то не так? Подумайте, почему.

С помощью другой операции, операции сравнения " = ", можно определить тождественность двух символьных величин.

Символьные величины считаются равными при абсолютном совпадении всех знаков символьных величин. Если же они хоть в одном знаке будут отличаться, то значит, равенство не выполняется.

Часто для написания программ с символьными значениями надо определять длину величины, то есть количество символов в ней. Для этой цели в языке КуМир имеется функция **длин**

Разберем пример:

```
алг
нач
  лит x, цел n, k
  x := "информатика"

  n := длин(x)



  k := длин("программа работает")



  вывод n, нс
  вывод k
кон
```

Начало алгоритма
Описание переменных
Присваивание значения величине x
Определение длины символьной строки и сохранение результата в переменную n
Определение длины символьной строки и сохранение результата в переменную k
Вывод значений переменных
Конец алгоритма

переменная n получит значение 11, а k будет равна 18 (не забудьте, что пробел тоже полноправный символ).

Обратите внимание, что аргументом функции **длин** может быть только символьная величина, а результатом – число.

Иногда приходится выделять фрагменты из символьной строки и работать с одним или с группой знаков. Для этого существует специальная конструкция **вырезка**.

Общий вид вырезки из символьной переменной А таков:
А [*с какого*: *по какой*], где

с какого – номер символа, с которого начинается вырезка
по какой символ – номер символа, которым заканчивается вырезка

Эта конструкция может использоваться для вырезки части текста символьной строки или ее замены

Предположим А="ИНФОРМАТИКА"

B := A[3:7] - данная запись предполагает, что в символьную переменную B, переписываются из символьной переменной A символы со 3 по 7 включительно. И ее значение будет равно «ФОРМА»

A[8:11] := "ЦИЯ" – данная запись оператора заменяет с 8 по 11 символы в символьной строке A. После этой операции символьная переменная A будет иметь значение равное слову “ИНФОРМАЦИЯ”

Разберем пример.

Сколько раз встречается буква «а» во введенной с клавиатуры строке?

Для того чтобы сосчитать количество букв “а”, надо из строки x поочередно выделять по одной букве в переменную y, сравнивать с образцом и в случае совпадения, увеличивать переменную - счетчик k на единицу (k := k + 1). Чтобы проверить всю строку нужно сначала определить ее длину l. Переменные x и y должны быть описаны как строковые (тип *лит*), а k и l – как целые (*цел*).

```
алг определение длины строки
нач
  цел i, k, l, лит x, y

  вывод "Введите строку символов"
  ввод x

  l := длин(x)



  k := 0

  нц для i от 1 до l

    y := x[ i: i]


    если y="a" то


      k := k + 1



    все
кц
```

Заголовок программы
Начало алгоритма
Раздел описания переменных

Ввод символьной строки

Определение длины символьной строки и сохранение результата в переменную l
Обнуление счетчика k

Начало цикла
Выделение одной буквы из строки на позиции номер i

Определение: является ли выделенная буква буквой «а»

Увеличение значения счетчика на 1 в случае выполненного условия

Конец цикла

Вывод количества букв «а» в строке

```
    вывод нс, "k=", k
кон
```

Конец алгоритма

Поскольку выделение одного символа встречается довольно часто, в КуМире разрешено вместо A[i:i] писать просто A[i].

Решить задачи:

1. С клавиатуры вводятся 4 слова. Определите, какое из них максимальной длины.
2. Используя операции соединения, сконструируйте сложные слова из введенных корней: пар, ход, воз, лет, вод (Используйте соединительную гласную «о», пример такого соединения приведен в тексте практической работы).
3. Определите, сколько раз встречается во введенной строке слог «ба». (Для проверки работы программы можете использовать такой пример: баран упал на барабан – слог «ба» встречается 3 раза).
4. Замените все пробелы во введенном с клавиатуры предложении на знаки подчеркивания.

Практическая работа №10 «Программирование линейных алгоритмов на языке программирования Python»

Цель работы: выработать практические навыки работы с системой программирования, научиться создавать, вводить в компьютер, выполнять и исправлять простейшие программы на языке Python в режиме диалога, познакомиться с диагностическими сообщениями компилятора об ошибках при выполнении программ, реализующих линейные алгоритмы.

Общие сведения: Линейным называется алгоритм, в котором результат получается путем однократного выполнения заданной последовательности действий при любых значениях исходных данных. Операторы программы выполняются последовательно, один за другим, в соответствии с их расположением в программе.

Задачи:

1. Дана длина ребра куба. Найти объем куба и площадь его боковой поверхности.
2. Три сопротивления R1, R2, R3 соединены параллельно. Найти сопротивление соединения.
3. Определить время падения камня на поверхность земли с высоты h.
4. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.
5. Треугольник задан координатами своих вершин. Найти:
    a. периметр треугольника;
    b. площадь треугольника.

Практическая работа №11 «Программирование линейных алгоритмов на языке программирования Python - 2»

Цель работы: закрепить практические навыки работы с системой программирования, создания, ввода в компьютер, выполнения и тестирования простейших программ на языке Python в режиме диалога.

Задачи:

1. Вычислить высоту треугольника, опущенную на сторону a, по известным значениям длин его сторон a, b, c.
2. Вычислить объем цилиндра с радиусом основания r и высотой h.
3. Определить расстояние, пройденное физическим телом за время t, если тело движется с постоянным ускорением a и имеет в начальный момент времени скорость V0
4. Вычислить площадь треугольника по формуле Герона, если заданы его стороны.
5. Определить координаты вершины параболы $y=ax^2+bx+c$. Коэффициенты a,b,c заданы пользователем.
6. По данным сторонам прямоугольника вычислить его периметр, площадь и длину диагонали.

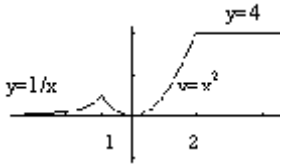
Практическая работа №12 «Программирование разветвляющихся алгоритмов на языке программирования Python»

Цель работы: научиться правильно использовать условный оператор if; научиться составлять программы для решения задач на разветвляющиеся алгоритмы.

Общие сведения: Алгоритм называется разветвляющимся, если он содержит несколько ветвей, отличающихся друг от друга содержанием вычислений. Выход вычислительного процесса на ту или иную ветвь алгоритма определяется исходными данными задачи. Перед выполнением работы необходимо ознакомится с правилами записи логических выражений, операций сравнения.

Задачи:

1. Даны действительные положительные числа x, y, z. Выяснить, существует ли треугольник с длинами сторон x, y, z.
2. Дано действительное a. Для функции f(a), график которой представлен на рисунке, вычислить f(a).



3. Даны три действительных числа. Выбрать из них те, которые принадлежат интервалу (1, 3).
4. Даны действительные числа x, y. Если x, y отрицательны, то каждое значение заменить его модулем; если отрицательное только одно из них, то оба значения увеличить на 0.5; если оба значения не отрицательны и ни одно из них не принадлежит отрезку [0.5, 2.0], то оба значения уменьшить в 10 раз; в остальных случаях x, y оставить без изменения.
5. Определить, какая из двух фигур (круг или квадрат) имеет большую площадь. Известно, что сторона квадрата равна a, радиус круга r. Вывести на экран название и значение площади большей фигуры.

Практическая работа №13 «Программирование разветвляющихся алгоритмов на языке программирования Python - 2»

Цель работы: закрепить практические навыки работы с условным оператором if; составления программ для решения задач на разветвляющиеся алгоритмы.

Задачи:

1. Определить, попадает ли точка M(x,y) в круг радиусом r с центром в точке (x0,y0)
2. Даны действительные числа a, b, c, x, y. Выяснить, пройдет ли кирпич с ребрами a,b,c в прямоугольное отверстие со сторонами x и y. Просовывать кирпич в отверстие разрешается только так, чтобы каждое из его ребер было параллельно или перпендикулярно каждой из сторон отверстия.
3. Сможет ли шар радиуса R пройти в ромбообразное отверстие со стороной P и острым углом Q?
4. Две точки заданы на плоскости своими координатами, которые могут быть как декартовыми, так и полярными. Требуется вычислить расстояние между этими двумя точками.

Практическая работа №14 «Программирование циклических алгоритмов»

Цель работы: закрепить практические навыки работы с системой программирования, научиться правильно использовать различные операторы циклов; научиться составлять программы решения задач с использование циклических структур.

Общие сведения: Алгоритм называется циклическим, если он содержит многократное выполнение одних и тех же операторов при различных значениях промежуточных данных. Число повторений этих операторов может быть задано в явной (цикл с известным заранее числом повторений) или неявной (цикл с неизвестным заранее числом повторений) форме.

Задачи:

1. Найти все двузначные числа, сумма цифр которых не меняется при умножении числа на 2,3,4,5,6,7,8,9.
2. Найти все трехзначные числа, сумма цифр которых равна данному целому числу.
3. Найти все трехзначные числа, средняя цифра которых равна сумме первой и второй цифр.
4. Найти все трехзначные числа, которые можно представить разностью между квадратом числа, образованного первыми двумя цифрами и квадратом третьей цифры.
5. Найти все двузначные числа, сумма квадратов цифр которых делится на 17.
6. Найти все трехзначные числа, представимые в виде сумм факториалов своих цифр.
7. Найти двузначное число, обладающее тем свойством, что куб суммы его цифр равен квадрату самого числа.
8. Найти двузначное число, равное утроенному произведению его цифр.
9. Определить, является ли заданное число совершенным, т.е. равным сумме всех своих (положительных) делителей, кроме самого этого числа (например, число 6 совершенно: 6=1+2+3).
10. Дано натуральное k. Напечатать k-ю цифру последовательности 1234567891011121314..., в которой выписаны подряд все натуральные числа.

Практическая работа №15 «Программирование с использованием списков»

Цель работы: научиться правильно использовать списки при написании программ на языке Python; научиться составлять программы решения задач, требующих хранения последовательно введённых чисел.

Общие сведения: Списки в Python - упорядоченные изменяемые коллекции объектов произвольных типов (почти как массив в других языках, но количество элементов и их типы могут отличаться).

Задачи:

1. Введите с клавиатуры в список пять целочисленных значений, введите их в одну сторону через запятую и найдите среднее арифметическое элементов списка.
2. Дан список из 10 целых чисел. Подсчитайте наибольшее число одинаковых идущих подряд в нем чисел.
3. Подсчитать сколько дней в январе было дней с температурой ниже, выше и равной средней температуре месяца (пользователь вводит по одной температуре, соответствующей определённому дню).
4. По фигурному катанию N судей выставляют оценки. Наибольшая и наименьшая отбрасываются, а по остальным выставляется средний бал. Напишите программу, вычисляющую среднюю оценку и выдайте номер судей, которые поставили наибольшую и наименьшую оценки.
5. Дан список целых чисел. Найти сумму максимума среди чётных и минимума среди нечётных.
6. Дана последовательность n-чисел. Заменить все её члены, больше заданного числа z, этим числом. Посчитать количество замен. Вывести изменённую последовательность.

Практическая работа №16 «Программирование строк и строковых функций»

Цель работы: научиться правильно использовать строковые функции при написании программ на языке Python.

Задачи:

1. Во введённой строке заменить пробелы или несколько пробелов символом *.
2. Вывести все числа и цифры из введённой строки.
3. Определить длину самого короткого слова в строке (строка состоит только из латинских символов и пробелов).
4. Определить процент строчных и прописных букв в строке от их общего числа.
5. Во введённой строке найти все слова, которые входят в неё только один раз.

Задания для самостоятельной работы студентов:

1. Мальчик купил в магазине n порций мороженного по цене 1200 руб. и k плиток шоколада по цене 3800 руб. Сколько всего потратил мальчик?

2. Составьте программу нахождения площади равнобедренной трапеции через стороны a, b, c.

3. Дано действительное число a. Не пользуясь никакими другими операциями, кроме умножения, получить: а) a^4 за две операции; б) a^6 за три операции; в) a^7 за четыре операции; г) a^8 за три операции; д) a^9 за четыре операции; е) a^{10} за четыре операции; ж) a^{13} за пять операций; з) a^{15} за пять операций; и) a^{21} за шесть операций; к) a^{28} за шесть операций; л) a^{64} за шесть операций;

4. Написать программу, которая печатает True или False в зависимости от того, выполняются или нет заданные условия:

- квадрат заданного трехзначного числа равен кубу суммы цифр этого числа;
- сумма двух первых цифр заданного четырехзначного числа равна сумме двух его последних цифр;
- среди цифр заданного трехзначного числа есть одинаковые;
- среди первых трех цифр из дробной части заданного положительного вещественного числа есть цифра 0.

5. Проверить, можно ли из четырех данных отрезков составить параллелограмм.

6. Найдите все значения натуральных чисел X, Y, Z из интервала от 1 до 50, удовлетворяющих равенству: $X^2 + Y^2 = Z^2$.

7. Составьте программу определения количества четных и нечетных цифр в записи данного натурального числа.

8. Вывести 15 первых чисел (начиная с 1) так, чтобы каждое последующее было в 2 раза больше предыдущего.

9. Дано вещественное число A и целое число N (> 0). Вывести $1 + A + A^2 + A^3 + ... + A^N$.

10. Ввести натуральное число с клавиатуры. Найти количество его цифр и их сумму

11. Даны натуральное число n, действительное число x. Вычислить:

а) $\sum_{i=1}^{n} \frac{(2i)! + |x|}{(i^2)!}$;

б) $\frac{1}{n!} \sum_{k=1}^{n} (-1)^k \frac{x^k}{(k!+1)!}$;

в) $\sum_{k=1}^{n} k^k x^{2k}$;

г) $\sum_{k=1}^{n} \sum_{m=k}^{n} \frac{x+k}{m}$.

12. Дано натуральное n. Вычислить: 1 + 1/2 + 1/3 + 1/4 + ... + 1/n.

13. Бизнесмен взял ссуду m тысяч рублей в банке под 20% годовых. Через сколько лет его долг превысит s тысяч рублей, если за это время он не будет отдавать долг?

14. Имеется кусок ткани длинной M метров. От него последовательно отрезаются куски равной длины. Все данные по использованию ткани заносятся в компьютер (вводятся пользователем в консоль с нажатием кнопки Enter). Компьютер должен выдать сообщение о том, что материала не хватает, если будет затребован кусок ткани больше длины, чем имеется.

15. Составить функцию создания прямоугольника из текста в консоли, окаймленного рамкой из псевдографических символов. В параметры функции ввести символ заполнения рамки и символ заполнения содержимого прямоугольника, а также его ширину и высоту.

Пример для параметров 'x', '$', 8, 5:

```
xxxxxxxx
x$$$$$$x
x$$$$$$x
x$$$$$$x
xxxxxxxx
```

16. Подсчитать общее количество цифр и знаков '+', '-', и '*', входящих в строку s.