

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

123317 Москва, Пресненская набережная, д 8, стр. 1, этаж 48, помещение 484с, комната 5  
ИНН 7701257303, ОГРН 1037739408189

Утверждаю:  
Директор ОЧУ «Специалист»



/Т.С.Григорьева/  
«01» февраля 2018 года

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ  
(ОБЩЕРАЗВИВАЮЩАЯ) ПРОГРАММА**

**«Проектное программирование на языке Java для  
школьников. Модуль 4»**

**для детей и взрослых**

Срок обучения: 4 недели

Форма обучения: очная

город Москва

## Аннотация

### 1. Пояснительная записка

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

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

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

Программа дополнительного образования «Проектное программирование на языке Java для школьников. Модуль 4» имеет научно-техническую направленность.

Возраст обучающихся: 12-16 лет.

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

### СОДЕРЖАНИЕ

1	Общая характеристика программы
---	--------------------------------

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

## 1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

Курс дополнительной общеобразовательной общеразвивающей программы для детей и взрослых «Проектное программирование на языке Java для школьников. Модуль 4» относится к программам дополнительных общеобразовательных курсов для детей и взрослых и представляет собой комплекс основных характеристик образования (объем, содержание, планируемые результаты), организационно-педагогических условий, форм аттестаций, разработанный и утвержденный ОЧУ «Специалист».

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

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

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

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

**Актуальность программы, соответствие государственному, социальному заказу/запросам.**

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

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

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

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

## **2. ЦЕЛЬ И ЗАДАЧИ ПРОГРАММЫ**

### **Цель и задачи курса:**

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

### **Задачи:**

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

## **3. УЧЕБНЫЙ ПЛАН**

В процессе преподавания курса «Проектное программирование на языке Java для школьников. Модуль 4» используются как классические методы обучения (лекции), так и различные виды практической работы обучающихся по заданию преподавателя, которые направлены на развитие в области программирования.

**Трудоемкость программы:** 24 академических часа (20 аудиторных).

**Срок обучения:** 4 недели

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

Занятия начинаются не ранее 8.00 часов утра и заканчиваются не позднее 20.00 часов. Для обучающихся в возрасте 14-18 лет допускается окончание занятий в 21.00

часов. При наличии двух смен занятий организуется не менее 30-минутный перерыв между сменами для уборки и проветривания помещений.

Продолжительность занятий детей в учебные дни - не более 3-х академических часов в день, в выходные и каникулярные дни - не более 4 академических часов в день. После 30-45 минут теоретических занятий организуется перерыв длительностью не менее 10 мин<sup>1</sup>.

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

### 3.1. Учебный план

№ п/п	Наименование модулей по программе	Общая трудоемкость (акад. часов)	Аудиторные часы			СРС, ч
			Всего	В том числе:		
				Лекций	Практических занятий	
1	<b>Модуль 1. Задачи первого (самого сложного) уровня - игры</b>	8	6	2	4	2
2	<b>Модуль 2. Задачи второго уровня - графика</b>	9	6	2	4	3
3	<b>Модуль 3. Задачи третьего уровня</b>	11	8	4	4	3
	<b>Итого:</b>	<b>28</b>	<b>20</b>	<b>8</b>	<b>12</b>	<b>8</b>
	Итоговая аттестация					

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

### 3.2. Рабочая программа курса/модулей

#### **Модуль 1. Задачи первого (самого сложного) уровня - игры**

Создание игры “Змейка”  
Создание игры “Морской бой”  
Создание игры “Пасьянс Косынка”.

#### **Модуль 2. Задачи второго уровня - графика**

Переворот экрана вверх ногами  
Глаза, следящие за курсором мыши  
Приложение с идущим человечком  
Создание двух лиц одного человека – на основании симметрии правой и левой половины

#### **Модуль 3. Задачи третьего уровня**

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

Приложение с убегающей кнопкой и с убегающим окном  
 Загрузка колоды карт и разложение колоды на игровом поле  
 Перемещение карты мышью, переворот карты  
 Создание пролетающей летающей тарелки  
 Создание падающей капли

### 3.3. Практические занятия

<i>№ n/n</i>	<i>№ и название темы дисциплины</i>	<i>Тематика семинарских (практических) занятий</i>	<i>СРС</i>
1	Модуль 1 <b>Задачи первого (самого сложного) уровня - игры</b>	Создание игры “Змейка” Создание игры “Морской бой” Создание игры “Пасьянс Косынка”	Доработка игры “Змейка”, Морской бой”, Пасьянс Косынка
2	Модуль 2. <b>Задачи второго уровня - графика</b>	Переворот экрана вверх ногами Глаза, следящие за курсором мыши Приложение с идущим человечком Создание двух лиц одного человека – на основании симметрии правой и левой половины	Создание двух лиц одного человека – на основании симметрии правой и левой половины
3	Модуль 3 <b>Задачи третьего уровня</b>	Приложение с убегающей кнопкой и с убегающим окном Загрузка колоды карт и разложение колоды на игровом поле Перемещение карты мышью, переворот карты Создание пролетающей летающей тарелки Создание падающей капли	Создание пролетающей летающей тарелки

### 4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

<i>№</i>	<i>Этапы в процессе освоения курса (темы/задачи)</i>	<i>Компоненты</i>		
		<i>знания</i>	<i>умения</i>	<i>навыки</i>
1	Модуль 1 <b>Задачи первого (самого сложного) уровня - игры</b>	<u>Знать:</u> Правила создания игр “Змейка”, Морской бой”, Пасьянс Косынка	<u>Уметь:</u> Создавать игры “Змейка”, Морской бой”, Пасьянс Косынка	<u>Владеть навыками:</u> Создание игр
2	Модуль 2. <b>Задачи второго уровня - графика</b>	<u>Знать:</u> Переворот экрана вверх ногами Глаза, следящие за курсором мыши	<u>Уметь:</u> Создание двух лиц одного человека – на основании симметрии правой и левой половины	<u>Владеть навыками:</u> Переворот экрана. Создание двух лиц одного человека
3	Модуль 3 <b>Задачи третьего уровня</b>	<u>Знать:</u> Приложение с убегающей	<u>Уметь:</u>	<u>Владеть навыками:</u>

		кнопкой и с убегающим окном Загрузка колоды карт и разложение колоды на игровом поле	Перемещение карты мышью, переверот карты Создание пролетающей летающей тарелки	пролетающей летающей тарелки
--	--	---	---	------------------------------

## 5. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

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

Неделя обучения	1	2	3	4	5	6	7	Итого часов
	пн	вт	ср	чт	пт	сб	вс	
1 неделя	-	-	-	-	-	4	-	4
СРС	-	-	-	-	-	1	-	2
2 неделя	-	-	-	-	-	4	-	4
СРС	-	-	-	-	-	1	-	2
3 неделя	-	-	-	-	-	4	-	4
СРС	-	-	-	-	-	1	-	2
4 неделя	-	-	-	-	-	4ИА	-	4
СРС	-	-	-	-	-	1	-	2
Итого:	-	-	-	-	-	28	-	28
Примечание: ИА – Итоговая аттестация (портфолио, практическое занятие)								

## 6. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

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

### 6.1. Требования к квалификации педагогических кадров

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

Требования к квалификации педагогических кадров, обеспечивающих реализацию образовательного процесса:

- наличие ученой степени (ученого звания);
- опыт работы в образовательной организации высшего образования не менее 5

лет;

– наличие документа о прохождении повышения квалификации или профессиональной переподготовки за последние 3 года.

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

## **6.2. Материально-техническая и ресурсная база**

Для реализации программы используется собственная материально-техническая база.

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

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

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

## **6.3. Общие требования к организации образовательного процесса**

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

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

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

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

Образовательная деятельность обучающихся предусматривает следующие виды учебных занятий и учебных работ: лекции, практические и семинарские занятия, круглые столы, мастер-классы, тренинги, семинары по обмену опытом, проведение открытых



занятий, консультации и другие виды учебных занятий и методической работы, определенные учебным планом.

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

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

**7.1.** Освоение образовательной программы, в том числе отдельной части или всего объема учебного предмета, курса, дисциплины (модуля) образовательной программы, сопровождается текущим контролем успеваемости и промежуточной аттестацией слушателей<sup>2</sup>.

**Итоговая аттестация** настоящей программой предусмотрена, предполагает проверку результатов освоения общеразвивающей программы для детей и взрослых «Проектное программирование на языке Java для школьников. Модуль 4» и проводится в форме **практической работы по созданию игр «Змейка», «Морской бой», «Пасьянс Косынка»**, самостоятельно разработанного обучающимся.

**Создание игры " создания Змейка”, Морской бой”, Пасьянс Косынка.**

- Создание игры “Змейка”
- Создание игры “Морской бой”
- Создание игры “Пасьянс Косынка”
- Создание падающей капли

**Практическая работа обучающегося** – это файловая папка (электронная) с документально подтвержденной информацией о личных, общественно значимых и академических достижениях учащегося, продемонстрированных за время освоения ДО программы. Эти сведения служат одним из инструментов объективного оценивания результатов обучения. Методика особенно эффективна для прикладных направлений: архитектуры, дизайна, технологий, программирования, строительства, журналистики и т. д. Портфолио/практическая работа или папка личных достижений, способствует объективности оценки результатов, достигнутых обучающимся в ходе учебной деятельности. Этот своеобразный презентационный портфель документов (кодов, классов и др.) может формироваться в бумажном и/или электронном виде.

№/п	Конкретные задания/выполнение работы	Баллы за выполнение
1	Создание игры “Змейка”	3 балл
2	Создание игры “Морской бой	2 балл
3	Создание игры “Пасьянс Косынка”	3 балл
4	Создание падающей капли	3 балла
<b>ИТОГО</b>		<b>0-7 не зачтено</b>

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

	8-11 -зачтено
--	---------------

## 7.2. Оценочные материалы для итоговой аттестации

### 7.2.1. Критерии и показатели результата обучения

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

### 7.2.2. Балльная система оценивания освоения программы

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

## 8. СПИСОК ЛИТЕРАТУРЫ

Методические материалы к курсу обучения.

1. "Основы программирования на языке Java для школьников" Модуль 1
  2. "Основы программирования на языке Java для школьников" Модуль 2
  3. «Основы программирования на языке Java для школьников» Модуль 3
  4. «Проектное программирование на языке Java для школьников. Модуль 4
- Интернет ресурс: <http://v8.1c.ru>
- Нормативная документация в электронной форме:
    - Федеральный закон от 27 июля 2006 г. N 152-ФЗ "О персональных данных"
    - План мероприятий "Развитие отрасли информационных технологий", утвержденный распоряжением Правительства России от 30.12.2013 №2602-р.
  - Профильные периодические электронные издания:
    - 1С:ИТС (Информационно-технологическое сопровождение пользователей 1С:Предприятия), <http://its.1c.ru/>
    - <http://v8.1c.ru/>

- Список рекомендуемой литературы:
  - Герберт Шилдт, "Java. Руководство для начинающих"

## Приложение 1.

### Методические материалы к модулям 1-3.

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

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

### Написание игры «змейка» class food

Java Выделить код

```
package snake;

import java.awt.*;
import java.util.Random;

public class food {
    int foodx ;
    int foody ;
    int size =10;
    Random generator = new Random();

    public food(){
        foodx = 10;
        foody = 10;
    }

    public void paintFood(Graphics g){
        //System.out.println("food is prepared");
        g.setColor(new Color(179,5,118));
        g.fillRect(foodx*size,foody*size,size,size);
    }

    public void changePosition(){
        int Nextx = generator.nextInt(46);
        int Nexty = generator.nextInt(25);
        foodx = Nextx;
        foody = Nexty;
        boolean at = SnakeFrame.getEaten();
```

Java Выделить код

```
        System.out.println(at);
        at = false;
        SnakeFrame.setEaten(at);
    }
}
```

## **class MainFrame**

Java Выделить код

```
package snake;
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
```

```
public class MainFrame{
    public static void main(String args[]){
        SnakeFrame frame = new SnakeFrame();

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);

    }
}
```

```
class SnakeFrame extends JFrame {
```

```
private static class NewMenuListener implements ActionListener {
    public NewMenuListener() {
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        throw new UnsupportedOperationException("Not supported yet.");
    }
}

    int i = 0;
    SnakePanel panel;
    boolean live = true;
    SnakeBody snake = new SnakeBody();
    SnakeDirection direction = new SnakeDirection(0);
    public food f = new food();
    public static boolean eaten = false;
    public boolean alive = true;
    public boolean done = true;
```

## Java ВЫДЕЛИТЬ КОД

```
private JMenuBar menu = null;
private final String fileItems[] = new String []{"Новая игра", "Выход"};
```

```
public SnakeFrame(){
    setSize(500,300);
    setTitle("Змейка");
createMenu();
setJMenuBar(menu);

    panel = new SnakePanel();
    panel.setBackground(new Color(10, 10, 10));
    add(panel);

}
public static boolean getEaten(){
    return eaten;
}
public static void setEaten(boolean eat){
    eaten = eat;
}
//Second Thread
class SnakeThread implements Runnable{
    Graphics g=null;

    public SnakeThread(){
        Thread t;
        t = new Thread(this);
        System.out.println("Second thread");
        t.start();
    }

    public void run(){
        System.out.println("Внутри run");
        while(alive == true){

            try {

                Thread.sleep(200);
                //System.out.println("Inside try"+i);

            }
            catch (InterruptedException e) {
```

## Java Выделить код

```
        // Do nothing
    }

    snake.move(direction);
    eat();
    die();
    panel.repaint();
    done = true;

}

}

}

class SnakePanel extends JPanel {

    public SnakePanel(){
        new SnakeThread();
        KeyHendler listener = new KeyHendler();
        addKeyListener(listener);
        setFocusable(true);

    }
    public void paintComponent(Graphics g){
        //super.paintComponent(g);
        g.clearRect(0,0,500,300);
        snake.paint(g);

        if(eaten == true)
            f.changePosition();
        f.paintFood(g);

    }

}

public void die(){
    if(snake.body.getLast().x >47 || snake.body.getLast().x <0 ||
        snake.body.getLast().y >25 ||snake.body.getLast().y <0 )
        alive = false;
    for(int i =1;i<snake.body.size()-1;i++){

        if(snake.body.getLast().x ==snake.body.get(i).x  &&  snake.body.getLast().y
        ==snake.body.get(i).y){
            System.out.println(i);
```

## Java ВЫДЕЛИТЬ КОД

```
        alive = false;
    }

}

}

public void eat(){
    //f = SnakeFrame.;
    if(f.foodx == snake.body.getLast().x && f.foody == snake.body.getLast().y){
        Point p = new Point (f.foodx,f.foody);
        System.out.println("Coordinats equals");
        //int numberOfParts = body.size();
        snake.body.add(0,p);
        eaten = true;
        //fod.changePosition();

    }
}

private class KeyHendler implements KeyListener{
    public void keyPressed(KeyEvent event){

        int keyCode = event.getKeyCode();

        if(keyCode == KeyEvent.VK_RIGHT&&done == true){ direction.changeToRight();done = false;}
        if(keyCode == KeyEvent.VK_LEFT&&done == true){ direction.changeToLeft();done = false;}
        if(keyCode == KeyEvent.VK_UP&&done == true){ direction.changeToUp();done = false;}
        if(keyCode == KeyEvent.VK_DOWN&&done == true) {direction.changeToDown();done =
false;}

    }
    public void keyReleased(KeyEvent event){
    }
    public void keyTyped(KeyEvent event){

    }
}

public class SnakeDirection{
    public final int right = 0;
    public final int down = 1;
    public final int left = 2;
    public final int up = 3;

    public int state = 0;

    public SnakeDirection(int s){
        state =s;
    }
}
```

Java Выделить код

```
    }

    public void changeToRight(){
        if(state != left)
            state = right;
    }
    public void changeToLeft(){
        if(state != right)
            state = left;
    }
    public void changeToUp(){
        if(state != down)
            state = up;
    }
    public void changeToDown(){
        if(state != up)
            state = down;
    }
}

    private void createMenu() {
menu = new JMenuBar();
JMenu fileMenu = new JMenu("Игра");
for (int i = 0; i < fileItems.length; i++) {
JMenuItem item = new JMenuItem(fileItems[i]);
item.setActionCommand(fileItems[i].toLowerCase());
item.addActionListener(new NewMenuListener());
fileMenu.add(item);
}
fileMenu.insertSeparator(1);

menu.add(fileMenu);
}
}
```

### **class SnakeBody**

Java Выделить код

```
package snake;

import java.util.LinkedList;
import java.util.ListIterator;
import java.awt.*;

public class SnakeBody extends Object {
    //List witch contains snake body;
    LinkedList<Point> body = new LinkedList<Point>();
}
```



## Java Выделить код

```
//part of snake
final int size = 10;
//co-ordinate X of one body part
public int x;
//co-ordinate Y of one body part
public int y;
```

```
food fod ;
```

```
// constructor that initialize snake body
SnakeBody(){
    Point point;

    point = new Point(3,6); body.add(0,point);
    point = new Point(4,6); body.add(1,point);
    point = new Point(5,6); body.add(2,point);
    point = new Point(6,6); body.add(3,point);
    point = new Point(7,6); body.add(4,point);
}
```

```
public void paint(Graphics g){
```