**Приложение Д**

**Выполнение работ внутреннего проектирования программного продукта: разработка схем проекта, математическое моделирование.**

**Цель занятия:**  приобрести умения проектирования проекта по требованиям заказчика, разработки математической модели и ее реализации в системе программирования.

**Оборудование:** ПК, IntelliJ IDEA

**Источники:**

1. [Равноускоренное движение — формулы, законы и примеры (skysmart.ru)](https://skysmart.ru/articles/physics/ravnouskorennoe-dvizhenie?ysclid=lpmrog1gi249397170)
2. [Задачи на движение тела, брошенного под углом к горизонту (sfiz.ru)](https://sfiz.ru/materials/mehanika/zadachi_na_dvizhenie_tel?ysclid=lpmrzc41zg430126089)
3. [Использование диаграммы классов UML при проектировании и документировании программного обеспечения / Хабр (habr.com)](https://habr.com/ru/articles/572234/)
4. Руководство Java. <https://metanit.com/java/tutorial/>
5. Junit тестирование. <https://youtu.be/VFDi6AA6fY4>

**ЗАДАНИЯ К ПРАКТИЧЕСКОМУ ЗАНЯТИЮ**

Вы являетесь разработчиком программного обеспечения в фирме ИП «Программист». К Вам поступил заказ разработать программный продукт, который бы позволял решать типовые задачи по физике по разделам:

1. Равноускоренное движение. При этом заказчик предоставил для решения задач следующую информацию: [Равноускоренное движение — формулы, законы и примеры (skysmart.ru)](https://skysmart.ru/articles/physics/ravnouskorennoe-dvizhenie?ysclid=lpmrog1gi249397170).
2. Движение тел под углом к горизонту. При этом заказчик предоставил следующую информацию: [Задачи на движение тела, брошенного под углом к горизонту (sfiz.ru)](https://sfiz.ru/materials/mehanika/zadachi_na_dvizhenie_tel?ysclid=lpmrzc41zg430126089)

Ваша задача разработать готовый к реализации программный продукт по технологии ООП.

**Для решения задачи выполните следующие задания:**

**Задание 1.**  Работая с источниками, предоставленными заказчиком, изучите предметную область. Выделите типовые задачи. Определите вычисляемые переменные (что надо найти) и переменные – исходные данные. Помните, что любой алгоритм реализуется в общем виде (например: решите пример 5x=10 в алгоритмизации будет выглядеть как решите пример ax=b. При этом в математике решение примера будет выглядеть как, а программировании решение будет записано в виде математической модели: , где результат вычисления будет зависеть от параметров a, b вводимых пользователем с клавиатуры:

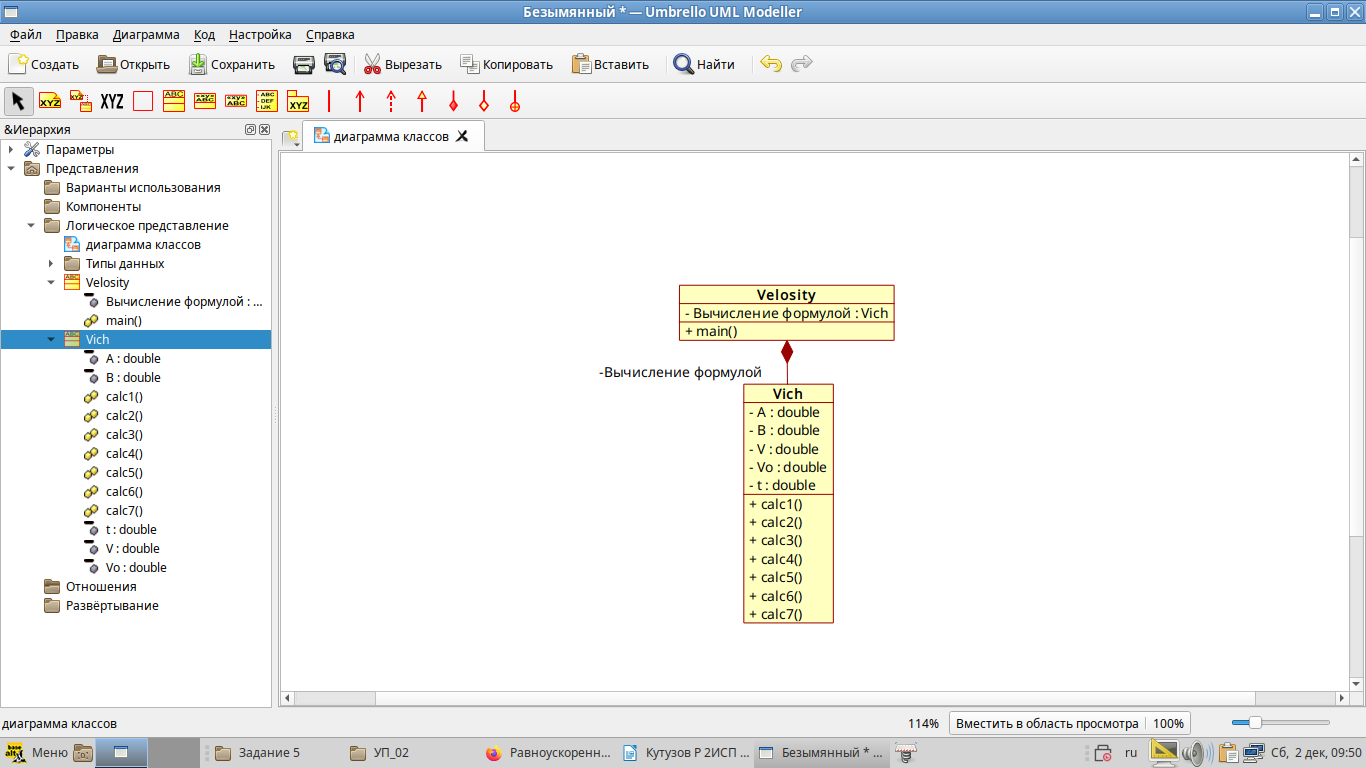
|  |  |  |  |
| --- | --- | --- | --- |
| Типовая задача (условие) | Математическая модель | Выводимые  Данные | Входные данные |
| Вычислить уравнение равноускоренного движения |  | a | V,Vo,t |
| Вычислить скорость |  | V(t) | Vo,a,t |
| Вычислить расстояние |  | S | Vo,t,a,t, |
| Вычислить расстояние |  | S | V,Vo,a |
| Вычислить расстояние |  | S | Vo,t,a,t, |
| Координата |  | x(t) | Xo,Vo,t,a |

**Задача 2. Выделите классы, которые будут использоваться для решения задачи**

|  |  |  |  |
| --- | --- | --- | --- |
| Класс | Назначение | Поля класса | Методы класса |
| Velosity | Вызов вычисляемых действий; |  | Main() |
| Vich | Класс для вычисления выражений | a, b – исходные данные | Deli(): double |

**Задача 3. Постройте диаграмму классов. Вставьте скриншот.**

Например (для задачи примера):



**Задача 4. Создайте тестовые сценарии для контроля правильности работы приложения**

Таблица 1 — Тестовый сценарий для класса

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номер теста | Наименование метода | Исходные данные | Ожидаемый результат (expected) | Фактический результат (actual) | Результат тестирования (Success/Fail) |
| Для примера  1. | Deli() | a=5; b=10 | 2 | 2 | Заполняется после выполнения тестирования |
| 2. | Deli() | a=0; b=10 | Ошибка! Деление на ноль!!! | Ошибка! Деление на ноль!!! | Заполняется после выполнения тестирования |

**Примечание:**

1. Тест должен обеспечивать покрытие всех функций (проверка хотя бы один раз)
2. Тест должен обеспечить покрытие всех переходов (проверка хотя бы один раз)
3. Тест должен обеспечить контроль выполнения алгоритма при граничных условиях

**Задача 8. Выполните разработку кода программы по технологии ООП. Зафиксируйте результаты разработки**

|  |  |
| --- | --- |
| Созданные классы (название) | Листинг (код, скрин) |
| **Velosity** | *\*\** *\* Class Velosity* *\*/*public class Velosity { // // Fields // private Vich Вычисление\_формулой;  // // Constructors // public Velosity () { };  // // Methods // // // Accessor methods // */\*\** *\* Set the value of Вычисление\_формулой* *\* @param newVar the new value of Вычисление\_формулой* *\*/*public void setВычисление\_формулой (Vich newVar) { Вычисление\_формулой = newVar; } */\*\** *\* Get the value of Вычисление\_формулой* *\* @return the value of Вычисление\_формулой* *\*/*public Vich getВычисление\_формулой () { return Вычисление\_формулой; } // // Other methods // */\*\** *\*/*public void main() { }} |
| **Vich** | */\*\** *\* Class Vich* *\*/*public class Vich { // // Fields // private double A; private double B; private double V; private double Vo; private double t; private double x0;  // // Constructors // public Vich(double v, double \_t, double v0, double a, double b, double \_x0) { A=a; B=b; V=v; Vo=v0; t=\_t; x0=\_x0; } // // Methods // // // Accessor methods // */\*\** *\* Set the value of A* *\* @param newVar the new value of A* *\*/*public void setA (double newVar) { A = newVar; } */\*\** *\* Get the value of A* *\* @return the value of A* *\*/*public double getA () { return A; } */\*\** *\* Set the value of B* *\* @param newVar the new value of B* *\*/*public void setB (double newVar) { B = newVar; } */\*\** *\* Get the value of B* *\* @return the value of B* *\*/*public double getB () { return B; } */\*\** *\* Set the value of V* *\* @param newVar the new value of V* *\*/*public void setV (double newVar) { V = newVar; } */\*\** *\* Get the value of V* *\* @return the value of V* *\*/*public double getV () { return V; } */\*\** *\* Set the value of Vo* *\* @param newVar the new value of Vo* *\*/*public void setVo (double newVar) { Vo = newVar; } */\*\** *\* Get the value of Vo* *\* @return the value of Vo* *\*/*public double getVo () { return Vo; } */\*\** *\* Set the value of t* *\* @param newVar the new value of t* *\*/*public void setT (double newVar) { t = newVar; } */\*\** *\* Get the value of t* *\* @return the value of t* *\*/*public double getT () { return t; } // // Other methods // */\*\** *\*/*public double calc1() { return (V - Vo) / t; } */\*\** *\*/*public double calc2() { return (Vo+A\*t); } */\*\** *\*/*public double calc3() { return Vo\*t+ (A\*Math.*pow*(t,2))/2; } */\*\** *\*/*public double calc4() { double S; S = (Math.*pow*(V,2) - Math.*pow*(Vo,2)) / 2 \* A; return S; } */\*\** *\*/*public double calc5() { return Vo\*t+(Math.*pow*(A,2)\*t)/2; } */\*\** *\*/*public double calc6() { return x0+Vo\*t+(A\*Math.*pow*(t,2))/2; } */\*\** *\*/*public double calc7() { return 0; }} |

Для задачи примера:

|  |  |
| --- | --- |
| Созданные классы (название) | Листинг (код, скрин) |
| Vich**Test** |  |

