**Приложение Д**

**Практическое занятие 5**

**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 |
| Для примера (поставленная задача):  Вычислить уравнение 5x=10 |  | x | b, a |
|  |  |  |  |

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

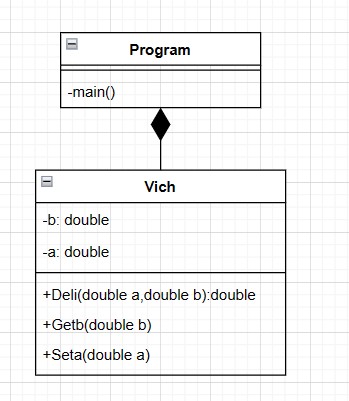
|  |  |  |  |
| --- | --- | --- | --- |
| Класс | Назначение | Поля класса | Методы класса |
|  |  |  |  |
|  |  |  |  |

**Для примера:**

|  |  |  |  |
| --- | --- | --- | --- |
| Класс | Назначение | Поля класса | Методы класса |
| Program | Вызов вычисляемых действий; | private:  double S;  double X;  double V;  double T;  double A;  double Angle; | Main()  public:Program()  void variablesInput() |
| Vich | Класс для вычисления выражений | a, b – исходные данные | Deli(): double  public:  void calcS(){}  void calcX(){}  void calcV(){}  void calcT(){}  void calcA(){}  void calcAngle(){}  }; |

**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) |
|  |  |
|  |  |
|  |  |

Для задачи примера:

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

**Задача 9. Выполните разработку сценария Junit теста функций классов программы. Для этого разработайте пакет Test. Выполните тестирование. Зафиксируйте отчет системы по тестированию.**

Используй материал видеоурока: <https://youtu.be/VFDi6AA6fY4>

Пример реализации сценария:

**Задание 10. Сделайте заключение о возможности внедрения программы**