

---

---

**Д.П. Хашимова, Х.Н. Рузметова**

**КЕЙС-ТЕХНОЛОГИЯ**  
**по учебной теме «Методы проектирования программных**  
**средств»**  
**предмета «Технологии программирования»**

***КЕЙС***  
***«Выбор оптимальной технологии***  
***программирования для***  
***создания информационной системы***  
***«Деканат»***

---

---

## I. КЕЙС

### «ВЫБОР ОПТИМАЛЬНОЙ ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ ДЛЯ СОЗДАНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ДЕКАНАТ»

*Кейс предназначен для студентов 2 курса, обучающихся по направлению образования 5521900 - «Информатика и информационные технологии».*

*Учебный предмет: «Технологии программирования».*

*Цель данного кейса: Развить умения и навыки по выбору оптимальной технологии программирования и разработке программного средства.*

*Учебные результаты: по результатам работы с кейсом студенты приобретают навыки по:*

- ✓ *выбору оптимального метода проектирования программного средства для конкретной ситуации;*
- ✓ *избранию средств языка программирования, наиболее подходящих для той или иной ситуации;*
- ✓ *умению самостоятельно разрабатывать готовые программные средства.*

*Перечень предзнаний и навыков. Для успешного решения кейса студент должен обладать знаниями о методах проектирования программных средств, о средствах программирования языка Паскаль.*

*Данный кейс отражает деятельность реальной институциональной системы. Объектом кейса является деканат факультета «Информационные системы и технологии» Ташкентского Государственного Экономического Университета.*

• *Источники информации: Рейтинговые ведомости и отчетная документация деканата.*

*Типологические признаки кейса: Данный кейс относится к категории полевых, сюжетных. Это организационно-институциональный кейс, построенный в виде аналитической записки.*

*Он малых размеров, не структурированный. По способу представления учебного задания – кейс-задание.*

*По дидактическим целям кейс относится к категории тренинговых, т.е. предназначен для отработки умений и навыков по учебной теме.*

*Способ оформления: электронный, печатный.*

*Кейс может быть использован при изучении дисциплин «Информатика», а также «Информационные системы и технологии».*

## ВВЕДЕНИЕ

Сегодня редкие предприятие, фирма или учебное заведение не используют вычислительные машины в своей повседневной деятельности для ведения бухгалтерского учета, контроля за выполнением заказов и договоров, а также для подготовки деловых документов. Несмотря на то, что большинство предприятий при создании информационных систем пользуется готовыми разработками (это и понятно, так как снижается стоимость затрат на их создание), информационные системы более эффективно функционируют в том случае, если в них отражены особенности системы управления объектом. Для этого возникает необходимость в создании дополнительных программных продуктов или адаптации и расширения функциональных возможностей имеющихся программных средств.

Проектирование программных средств – наиболее ответственный этап жизненного цикла программных продуктов, поскольку важно установить, насколько создаваемая программа соответствует спецификациям и требованиям со стороны пользователей. Затраты на создание, сопровождение и эксплуатацию программных продуктов, научно-технический уровень разработки, время морального устаревания и многое другое – все это также зависит от проектных решений.

Существует большое разнообразие методов проектирования программных средств. Выбор наиболее оптимального для данного конкретного случая метода проектирования - одна из сложнейших задач проектировщиков и от его правильности зависит эффективность работы данного программного средства.

Между тем кейс позволяет освоить методы проектирования с целью определить способ нахождения оптимального метода проектирования в конкретном учреждении. Знания, полученные в период теоретической подготовки, способствуют моделированию практической деятельности по диагностике ситуации, выделению проблемы, проектированию последовательности конкретных шагов по разрешению проблемы.

По результатам работы с кейсом студенты приобретают навыки по:

- ✓ выбору оптимального метода проектирования программного средства для конкретной ситуации;
- ✓ избранию средств языка программирования, наиболее подходящих для той или иной ситуации;
- ✓ умению самостоятельно разрабатывать готовые программные средства.

## **СИТУАЦИЯ**

Заместители деканов факультетов ТГЭУ периодически выдают сведения в различных разрезах, в том числе о неуспевающих студентах, об отличниках и т.д.

Например, информация об отличниках приводится в следующем виде:

<b>Сведения об отличниках</b>		
<b>№</b>	<b>ФИО</b>	<b>№ зачетки</b>
1	И.Т. Махкамов	006897
2	О.П. Файзиев	006899

Проверено 20 студентов  
Из них 2 отличника

Учитывая, что по каждому учебному предмету заполняется отдельная ведомость и в разных группах имеются разные учебные дисциплины и для того, чтобы выдать эти сведения, необходимо проанализировать большое количество рейтинговых ведомостей. В рейтинговых ведомостях вместо оценок указываются общее количество баллов и рейтинговый балл. Из суммы общего количества баллов можно определить, какую именно оценку получил студент, исходя из следующих критериев:

- 86 - 100 - «отлично»;
- 71 - 85 - «хорошо»;
- 56 – 70 - «удовлетворительно»;
- 0 – 55 – «неудовлетворительно».

Помимо этого, деканат выдаёт сведения о студентах, пропустивших более 12, 18 или 30 часов аудиторных занятий. Аналогичным образом приводятся сведения и по другим разрезам.

На первом курсе факультета «Информационные технологии и менеджмент» обучение ведется в 24 академических группах, на втором – в 15, на третьем – в 18, а на четвертом курсе – в 21 группе.

Декан факультета для того, чтобы облегчить труд своих заместителей и увеличить оперативность получения информации, дал задание центру «Информационные технологии», образованному при деканате, создать информационную систему «Деканат», призванную обеспечить автоматизированный процесс выдачи сведений в различных разрезах. Кроме того, декан ограничил время предоставления готового программного продукта.

Чтобы своевременно выпонить данное задание, руководитель центра «Информационные технологии» разделил между его сотрудниками разработку отдельных фрагментов этой программы. Спустя определенное время, они предоставили руководителю разработанные ими фрагменты,

которые он объединил и произвёл тестирование их правильной сопряженности.

**Задание:** Выберите оптимальный метод проектирования программного средства «Деканат». Используя эффективные инструментальные средства программирования, разработайте фрагмент программного средства, предоставляющего сведения об успеваемости студентов.

## II. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТА

### *Проблема*

Выбор оптимального метода проектирования и средств языка программирования для разработки программы для информационной системы «Деканат».

### *Подпроблемы*

1. Избрание оптимального метода проектирования программного средства.
2. Выбор средств языка программирования.
3. Разработка текста программы.

### *Инструкция к самостоятельной работе по анализу и решению проблемной ситуации. Критерии оценки (письменной работы)*

Этапы решения	Содержание работы	Критерии и показатели оценки (макс. балл)
1. Обоснование проблемы и подпроблемы	Используя знания, полученные по учебной теме, обоснуйте (объясните) проблемы и подпроблемы.	-
2. Анализ ситуации и выбор оптимального метода проектирования программного средства	1. Анализ заданной конкретной ситуации и определение существенной, для решения задачи, информации. Обоснуйте ваш выбор. 2. Выбор оптимального метода проектирования программного средства. <i>Подсказка: Обратите внимание на действия руководителя центра «Информационные технологии» при выборе метода.</i>	Ясность и точность изложения, аргументирование выбора: 1 балл
3. Выбор средств языка программирования	1. Прежде, чем приступить к разработке программы, следует определиться, какими именно средствами языка	Ясность и точность изложения, аргументирование выбора:

	программирования вы будете пользоваться: функциями, процедурами или модулями. Обоснуйте свой выбор. 2. Определитесь, какими типами данных вы будете пользоваться для отображения исходных данных: массивы, записи или файлы. Обоснуйте свой выбор.	1 балл
4. Разработка текста программы	Разработайте фрагмент программы, которая по требованию выдаёт сведения об отличниках, хорошистах, троечниках или неуспевающих студентах. <i>Посказка: Пример формы выдачи сведений представлен в ситуации данного кейса.</i>	Простота и надежность: 3,5 баллов
5. Отладка и тестирование программы	Произведите отладку и тестирование правильности работы программы на контрольных примерах.	Правильность полученных результатов: 0,5 баллов

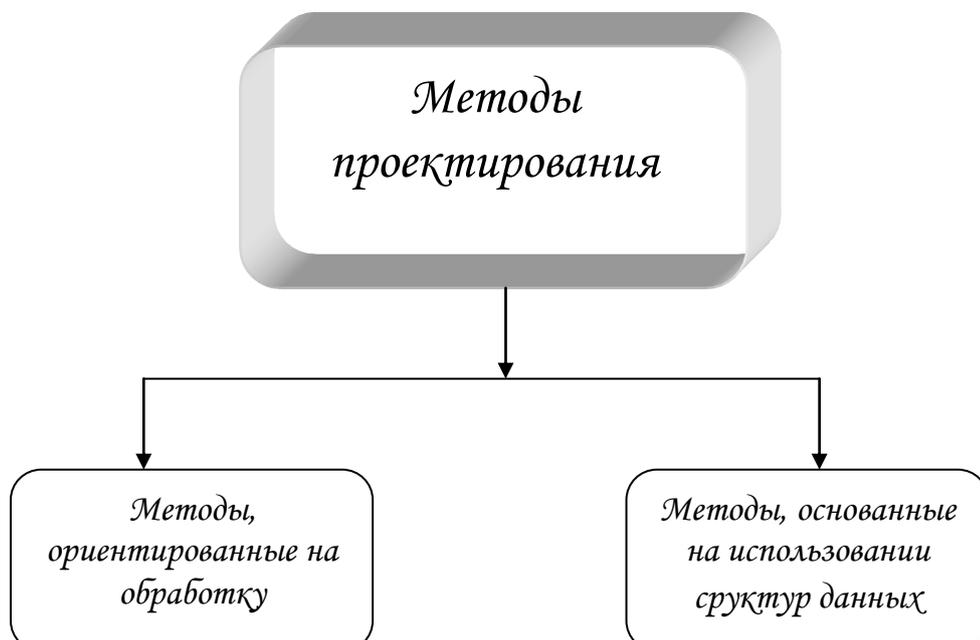
86-100% / 5,2 - 6 балла – «отлично»

71-85% / 4,3 - 5,1 балла – «хорошо»

55-70% / 3,3- 4,2 балла – «удовлетворительно»

### *Учебно-методические материалы*

#### **Методы проектирования программных средств**





### **Методология Джексона**

Здесь структура данных - ключевой элемент в построении проекта. Структура программы определяется структурой данных, подлежащих обработке. Программа представляется как механизм, с помощью которого входные данные преобразуются в выходные. В методе предусматривается:

- разработка и изображение структуры входных и выходных данных;
- изображение структуры программы путем соединения изображений этих структурных элементов;
- определение дискретных операций над структурами данных;
- построение алгоритмов обработки структур данных.

### **Методология Уорнера**

Подобна предыдущей методологии, но процедура проектирования более детализирована. Используются следующие виды представления проекта:

- диаграммы организации данных (описывают входные и выходные данные);
- диаграммы логического следования (логический поток этих данных);
- список инструкций (команды, используемые в проекте);
- псевдокод (описание проекта);
- определение входных данных системы;
- организация входных данных в иерархическую структуру;
- детальное определение формата элементов входного файла;
- то же самое для выходных данных;
- спецификация программы: чтение, ветвление, вычисление, выходы, вызовы подпрограмм;
- составление диаграммы (по типу блок-схем), указывающей на логическую последовательность инструкций.

### **Метод иерархических диаграмм**

На основе этого метода определяется связь между входными, выходными данными и процессом обработки с помощью иерархической декомпозиции системы (без детализации). По сути используются три элемента: вход, обработка и выход.

Алгоритм проектирования по данному методу заключается в следующих шагах:

- начать с наивысшего уровня абстракции, определив вход, выход и обработку;
- соединить каждый элемент входа и выхода с соответствующей обработкой;
- документировать каждый элемент системы, используя диаграммы;
- детализировать диаграммы, используя шаги 1-3.

### **Объектно-ориентированная методология проектирования.**

Основана на концепции упрятывания информации и абстрактных типов данных. В качестве объектов, рассматриваются данные, модули и системы. Каждый объект содержит некоторую структуру данных с набором процедур, определяющих, как именно работать с этими данными. Согласно этой методологии, создаются абстракции по заданной проблемной области:

- установление проблемы;
- развитие неформальной стратегии, отвечающей требованиям к системе;
- формализация стратегии;

- создание объектов и их атрибутов;
- определение операций над объектами;
- установка интерфейсов;
- реализация операций.

## Средства программирования языка Паскаль

В языке программирования должны быть средства, позволяющие разбивать программу на логически связанные блоки, каждый из которых решает некоторую подзадачу. Помимо этого, порой бывает, что такого рода подзадачи должны выполняться в программе неоднократно.

В Турбо-Паскале такими средствами являются *процедуры, функции и модули*.

**Функция** — это независимый программный блок (программная единица), позволяющий получить только один результат.

Процедуры (и функции) размещаются в программе сразу же после описания переменных (в разделе **Var**) и перед словом **Begin**, указывающим начало ее основной части.

## Структура процедуры

```
Procedure Имя (Формальные_параметры); {заголовок процедуры}  
<Описание локальных переменных>  
Begin  
  Операторы {Тело процедуры}  
End;
```



## Структура функции

```
Function Имя(формальные_параметры):Тип_Результата;  
  <Описание локальных переменных>  
Begin  
  ....  
  Имя := Выражение;  
  ....  
End;
```

**Модули** являются основой *модульного программирования* в Турбо-Паскале. Они используются для создания библиотек, которые могут включаться в различные программы (при этом совсем необязательно иметь в наличии исходный текст), большие же программы могут разделяться на логически связанные модули.

В модулях описываются переменные, константы, типы, классы, процедуры и функции. Для того, чтобы эти объекты можно было использовать в вызывающем модуле (которым может быть и основная программа), следует указать имя файла модуля (без расширения .pas) в разделе *uses* вызывающего модуля. Файл модуля должен находиться либо в том же каталоге, что и основная программа, либо в подкаталоге *Units* системного каталога программы Pascal ABC.

Модуль в Pascal ABC представляет собой файл со следующим содержанием:

```
unit имя модуля;  
раздел подключения модулей  
раздел описаний  
раздел инициализации  
раздел финализации  
end.
```

Первая строка обязательна и называется заголовком модуля.

Раздел подключения модулей начинается со служебного слова *uses*, за которым следует список имен модулей, перечисляемых через запятую.

Раздел описаний может включать разделы описания переменных, констант, типов, процедур и функций, которые следуют друг за другом в произвольном порядке.

Раздел инициализации состоит из служебного слова *initialization*, после которого следуют операторы, разделяемые символом «точка с запятой». Операторы из раздела инициализации модуля выполняются до начала основной программы.

Раздел финализации состоит из служебного слова *finalization*, после которого следуют операторы, разделяемые символом «точка с запятой». Операторы из раздела финализации модуля выполняются после окончания основной программы.

Раздел финализации может отсутствовать, либо оба раздела инициализации и финализации могут отсутствовать. Раздел инициализации

может также начинаться со служебного слова *begin*, но в этом случае раздел финализации отсутствует.

Поскольку система Pascal ABC не создает кода на диске, то по существу, модули являются аналогом включаемых файлов. В частности, они всякий раз компилируются при компиляции основной программы. Однако, если при этом один и тот же модуль подключается в нескольких модулях, то данный модуль компилируется лишь один раз.

### **III. ВАРИАНТ РЕШЕНИЯ КЕЙСА ПРЕПОДАВАТЕЛЕМ-КЕЙСОЛОГОМ**

#### ***1. Анализ ситуации и выбор оптимальной модели проектирования программного средства.***

Анализ ситуации показал, что руководитель центра «Информационные технологии» для разработки информационной системы «Деканат» выбрал метод модульного проектирования: он разделил между сотрудниками центра разработку отдельных фрагментов, с тем чтобы сократить время разработки и объем программы. Эти действия свойственны для модульного проектирования. Он принял правильное решение, так как у него было ограничено время, и руководитель был вынужден разбить задачу на подзадачи и каждому программисту поручить разработку отдельного фрагмента, т.е. решить подзадачу в виде модуля, разработка которого не зависит от разработки других модулей. Следовательно, каждый программист разрабатывает и отлаживает свой фрагмент независимо от других, что позволяет сократить время разработки, увеличить надежность и упростить процесс разработки программы.

#### ***2. Выбор средств языка программирования.***

В языке Паскаль существуют такие средства, как процедуры, функции или модули, облегчающие программисту процесс разработки программ. Для нашего программного средства целесообразнее выбрать процедуры и модули, так как модульное программирование основано на разработке программ, состоящих из модулей. Модули, в свою очередь, должны состоять из процедур и (или) функций. Функции для нашей программы не нужны, поскольку после выполнения они возвращают одно и то же значение.

Для того, чтобы программа не получилась громоздкой и для облегчения работы с исходными данными, целесообразно отображать их в типизированных файлах. Причем поле файла, содержащее оценки студента по учебным предметам, должно представлять собой одномерный массив. Следовательно, раздел описания типов будет иметь следующий вид:

```
Type
  Stud=Record
    Fam, Im, Ot : String[10];
    NumZ : integer;
```

```

        Otm : Array[1..50] Of Integer;
    End;

```

где Fam - фамилия студента,  
 Im - имя студента,  
 Ot - отчество студента,  
 NumZ - номер зачетной книжки,  
 Otm - массив, содержащий общее количество набранных баллов,  
 по учебным предметам.

### ***3. Разработка текста программы.***

Для этого воспользуемся языком ABC pascal. Фрагмент программы, выдающий сведения об успеваемости студентов, оформим в виде модуля **Sved**, состоящего из четырех процедур. Процедура Otl - выдает сведения об отличниках, Hor - предоставляет данные о хорошистах, Tr - содержит информацию о троечниках, Dv - выдает сведения о неуспевающих студентах.

```

Unit Sved;
Type
Stud = Record
Fam, Im, Ot : String[10];
Numz : LongInt;
Otm : Array[1..50] Of Integer;
End;
Var
Vedom : File Of Stud;
A : Stud;

Procedure Otl(FilName :String[10];M:integer);
Var
i,j,kdv,K2 : Integer;
Begin
Assign(Vedom,FilName);
Reset (Vedom); {открытие файла, 1-я запись - в
буфере}
kdv := 0;
I := 0;
Writeln('Список фамилий и номеров зачетов
отличников');
While Not Eof(Vedom) Do
Begin
K2 := 0;
Read(Vedom,A);
I := i+1;

```

```

With A Do
Begin
For j := 1 To M Do
If Otm[j] > 85 Then
K2 := K2+1;
If K2 = M Then
Begin
kdv := kdv+1;
Writeln (Fam:15, Numz:8);
End;
End; {Whith}
End; {While}
Close (Vedom);
Writeln ('Проверено ', i, ' студентов. ');
Writeln ('Из них отличников ', kdv, ' человек ');
WriteLn ('Нажмите Enter ');
Readln;
End;

Procedure Hor (FilName :String[10]; M:integer);
Var
i, j, kdv, K2, k3 : Integer;
Begin
Assign (Vedom, FilName);
Reset (Vedom); {открытие файла, 1-я запись - в
буфере}
kdv := 0;
I := 0;
Writeln ('Список фамилий и номеров зачетов
хорошистов ');
While Not Eof (Vedom) Do
Begin
K2 := 0; K3:=0;
Read (Vedom, A);
I := i+1;
With A Do
Begin
For j := 1 To M Do
If Otm[j] > 70 Then
K2 := K2+1;
If Otm[j] > 85 Then
K3 := K3+1;
If (K3<> M) and (k2=M) Then
Begin
kdv := kdv+1;

```

```

Writeln(Fam:15,Numz:8);
End;
End;{Whith}
End;{While}
Close(Vedom);
Writeln('Проверено ',i,' студентов. ');
Writeln('Из них хорошистов ',kdv,' человек');
WriteLn('Нажмите Enter');
Readln;
End;

Procedure Tr(FilName :String[10];M:integer);
Var
i,j,kdv,K2,k3,k4: Integer;
Begin
Assign(Vedom,FilName);
Reset (Vedom); {открытие файла, 1-я запись - в
буфере}
kdv := 0;
I := 0;
Writeln('Список фамилий и номеров зачетов
троечников');
While Not Eof(Vedom) Do
Begin
K2 := 0; K3:=0; K4:=0;
Read(Vedom,A);
I := i+1;
With A Do
Begin
For j := 1 To M Do
If Otm[j] > 55 Then
k2 := K2+1;
If Otm[j] > 70 Then
k3 := K3+1;
If Otm[j] > 86 Then
k4 := K4+1;
If (k3<>M) and (k4<>M)and (k2=M) Then
Begin
kdv := kdv+1;
Writeln(Fam:15,Numz:8);
End;
End;{Whith}
End;{While}
Close(Vedom);
Writeln('Проверено ',i,' студентов. ');

```

```

Writeln('Из них троечников ',kdv,' человек');
Writeln('Нажмите Enter');
Readln;
End;

Procedure Dv(FilName :String[10];M:integer);
Var
i,j,kdv,K2 : Integer;
Begin
Assign(Vedom,FilName);
Reset (Vedom); {открытие файла, 1-я запись - в
буфере}
kdv := 0;
I := 0;
Writeln('Список фамилий и номеров зачетов
неуспевающих студентов');
While Not Eof(Vedom) Do
Begin
K2 := 0;
Read(Vedom,A);
I := i+1;
With A Do
Begin
For j := 1 To M Do
If Otm[j] < 55 Then
K2 := K2+1;
If K2 > 0 Then
Begin
kdv := kdv+1;
Writeln(Fam:15,Numz:8);
End;
End;{Whith}
End;{While}
Close(Vedom);
Writeln('Проверено ',i,' студентов. ');
Writeln('Из них неуспевающих ',kdv,' человек');
Writeln('Нажмите Enter');
Readln;
End;

Begin
{ секция инициализации не содержит операторов }
End.

```

#### **4. Отладка текста программы.**

Для того, чтобы удостовериться в правильности работы полученного модуля, разработаем фрагмент текста основной программы, к которому уже подключен разработанный модуль и осуществляется вызов его процедур.

```
Program Sozd_Fil;
Uses Sved;
Type
  Stud=Record
    Fam, Im, Ot : String[10];
    NumZ : LongInt;
    Otm : Array[1..50] Of Integer;
  End;
Var
  Vedom : File Of Stud;
  A : Stud;
  FilName : String[10]; { имя набора данных }
  Rej,j,i,n,M : Integer;
Begin
  Writeln('Введите имя набора данных');
  Readln(FilName);
  Assign(Vedom,FilName);
  Rewrite(Vedom); {Открытие файла для записи}
  Writeln('Введите количество студентов');
  Readln(n);
  Writeln('Введите           количество           учебных
предметов');
  Readln(M);
  Writeln ('Введите сведения о каждом студенте');
  For i:=1 to n do
    Begin
      With A do
        Begin
          Write('Фамилия: ');
          ReadLn(Fam);
          Write('Имя: ');
          ReadLn(Im);
          Write('Отчество: ');
          ReadLn(Ot);
          Write('Номер зачетки: ');
          ReadLn(Numz);
          Writeln('Введите набранные баллы по
одной: ');
          For J:= 1 to M do
            Begin
```

```

WriteLn('Введите набранные баллы по
',J,'-му предмету');
ReadLn(Otm[j]);
End;
End;
Write(Vedom,A);      {Запись      в      файл
переписывается целиком}
End;

While True do
Begin
Writeln (' Укажите режим:');
Writeln (' 1: Сведения об отличниках');
Writeln (' 2: Сведения о хорошистах');
Writeln (' 3: Сведения о троечниках');
Writeln (' 4: Сведения о неуспевающих студентах');
Writeln ('5: Выход из программы');
ReadLn (Rej);
Case Rej of
1: Otl(Filename,M);
2: Hor(Filename,M);
3: Tr(Filename,M);
4: Dv(Filename,M);
5: exit;
end
end
End.

```

#### **IV. КЕЙС-ТЕХНОЛОГИЯ ОБУЧЕНИЯ НА ЛАБОРАТОРНОМ ЗАНЯТИИ**

**по теме: «Выбор оптимальной технологии программирования для  
создания информационной системы «Деканат»**

##### **4.1. Модель технологии обучения**

<i>Количество студентов: до 12 чел.</i>	<i>Время 2 часа</i>
<i>Форма учебного занятия</i>	Лабораторное занятие, основанное на разрешении проблемных ситуаций («кейс-стади»)
<i>План лабораторного занятия</i>	1. Введение в учебное занятие. 2. Актуализация знаний.

	<p>3. Работа с кейсом в минигруппах.</p> <p>4. Презентация результатов.</p> <p>5. Дискуссия, оценка и выбор приоритетной идеи разрешения проблемной ситуации.</p> <p>6. Заключение. Оценка деятельности групп и студентов, степени достижения цели учебного занятия.</p>
<p><i>Цель учебного занятия:</i> Освоение методов проектирования программных средств с целью найти оптимальный метод для проектирования программ для конкретной ситуации.</p>	
<p><i>Задачи преподавателя:</i></p> <ul style="list-style-type: none"> <li>• объяснить методы проектирования программных средств;</li> <li>• ознакомить с ситуацией кейса, научить выделять проблему и задачи по её решению;</li> <li>• объяснить алгоритм действий по разрешению проблемы;</li> <li>• предоставить возможность определить способ нахождения оптимального метода проектирования для конкретных ситуаций</li> </ul>	<p><i>Результаты учебной деятельности:</i></p> <ul style="list-style-type: none"> <li>• рассказывают о методах проектирования программных средств;</li> <li>• исходя из ситуации, характеризуют проблему и задачи по её решению;</li> <li>• определяют последовательность конкретных действий по разрешению проблемы;</li> <li>• определяют способы нахождения оптимального метода проектирования программных средств для конкретных ситуаций.</li> </ul>
<p><i>Методы обучения</i></p>	<p>Кейс, дискуссия, практический метод</p>
<p><i>Средства обучения</i></p>	<p>Кейс, учебные материалы, доска, флипчарт, маркеры, скотч, компьютерные технологии</p>
<p><i>Формы обучения</i></p>	<p>Индивидуальная, коллективная, работа в группах</p>
<p><i>Условия обучения</i></p>	<p>Аудитория с техническим оснащением (компьютер, проектор), приспособленная для организации групповой работы</p>
<p><i>МиО</i></p>	<p>Наблюдение, блиц-опрос, презентация, взаимооценка, оценка</p>

#### 4.2. Технологическая карта учебного занятия

Этап, время	Деятельность	
	преподавателя	студентов
Подготовительный этап	Подготавливает и заблаговременно раздаёт студентам материалы кейса для ознакомления и решения. Даёт задание самостоятельно провести анализ и решить кейс в соответствии с заданной инструкцией к самостоятельной работе. Знакомит с критериями оценки результатов индивидуальной (письменной) работы.	Самостоятельно знакомятся с содержанием кейса, индивидуально его решают
1 этап. Введение в учебное занятие (10 мин.)	1.1. Называет тему лабораторного занятия, его цель, задачи и планируемые результаты учебной деятельности. 1.2. Знакомит с режимом работы на данном занятии и критериями оценки результатов (Приложение 1).	Слушают, задают уточняющие вопросы Знакомятся
2 этап Основной этап (65 мин.)	2.1. Проводит блиц-опрос с целью активизировать знания обучающихся по теме (Приложение 2). 2.2. Разделяет обучающихся на группы и назначает спикера. Знакомит с правилами работы в группе (Приложение 3) и презентации (Приложение 4). Даёт задание: обсудить результаты индивидуальной работы с кейсом, оценить и скомбинировать отдельные фрагменты индивидуально разработанных программ для оптимизации структуры групповой программы и даёт время для подготовки к презентации. 2.3. Координирует, консультирует и направляет учебную деятельность, а также проверяет и оценивает результаты индивидуальной работы. 2.4. Организует презентацию, обсуждение и взаимооценку презентаций. Комментирует, корректирует ответы, особо обращает внимание на выводы, осуществленные в процессе анализа и	Отвечают на вопросы Выполняют учебное задание  Группы проводят презентацию результатов работы. Участвуют в обсуждении,

	разрешения ситуации.	задают вопросы и оценивают.
3 этап Заключительно-оценочный (5 мин.)	3.1. Подводит итоги, обобщает результаты, объявляет оценки индивидуальной и совместной работы. Отмечает значимость полученных знаний для будущей профессиональной и учебной деятельности.	Слушают Высказывают своё мнение

## Приложение 1

### Таблица оценки работы в мини группах

Список группы	Показатели оценки			Общая сумма баллов (1,5)	Оценка
	Активность (0,4)	Умение обосновывать свой вариант решения (0,6)	Презентация (0,5)		
1					
2					
3					

1,3– 1,5 баллов - «отлично»;

1 – 1,2 балла – «хорошо»;

0,7 – 0,9 баллов – «удовлетворительно»;

0 – 0,6 балла – «неудовлетворительно».

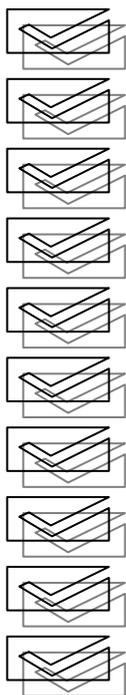
## Приложение 2

### Вопросы для актуализации знаний (блиц-опрос)

1. Какими важнейшими признаками классифицируются методы проектирования программных средств?
2. Какие подходы к проектированию программ наиболее распространены?
3. В чем заключается суть структурного проектирования и какими методами оно представлено?
4. В чем заключается суть объектно-ориентированного подхода?

### Правила работы в группах

- ✓ Каждый должен слушать своих товарищей, проявляя вежливость и доброжелательность;
- ✓ Каждый должен работать активно, совместно и ответственно относиться к порученному заданию;
- ✓ Каждый должен просить о помощи, когда она ему необходима;
- ✓ Каждый должен оказать помощь, если его об этом попросят;
- ✓ Каждый должен принимая участие в оценке результатов работы группы;
- ✓ Каждый должен четко понимать:
- ✓ Помогая другим, учимся сами!
- ✓ Мы в одной лодке: или выплываем, или вместе утонем!



#### Правила презентации:

- Презентация начинается с самопрезентации - представления исполнителей проекта: «Я,.....».
- Выступающие соблюдают регламент. Общая продолжительность презентации не менее 7-10 минут и не более 25 минут.
- Во время презентации должны быть обеспечены четкое распределение функций между членами группы и согласованность их действий.
- Информация должна быть представлена графически (схемы, таблицы, графики т.п.) и прокомментирована.
- Во время презентации не допускаются

