МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»

УТВЕРЖДАЮ
Директор ИЭиАС
С.И. Лукьянов

26.02.2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

ПАТТЕРНОЕ ПРОГРАММИРОВАНИЕ

Направление подготовки (специальность)
09.03.01 Информатика и вычислительная техника

Направленность (профиль/специализация) программы
Программное обеспечение средств вычислительной техники и автоматизированных систем

Уровень высшего образования - бакалавриат

Форма обучения
очная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Вычислительной техники и программирования
Курс	4
Семестр	8

Магнитогорск
2020 год

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 929)

Рабочая программа рассмотрена и одобрена на заседании кафедры Вычислительной техники и программирования
19.02.2020 г. протокол № 5

Зав. кафедрой _____ О.С. Логунова

Рабочая программа одобрена методической комиссией ИЭ и АС
26.02.2020 г. протокол № 5

Председатель _____ С.И. Лукьянов

Рабочая программа составлена:
ст. преподаватель кафедры ВТиП, _____ В.Е. Торчинский

Рецензент:
начальник отдела технологических платформ
ООО «Компас Плюс» , канд. техн. наук _____ Д.С. Сафонов

Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2021 - 2022 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2022 - 2023 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2023 - 2024 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2024 - 2025 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ О.С. Логунова

1 Цели освоения дисциплины (модуля)

Целями освоения дисциплины (модуля) «Паттерное программирование» является освоение студентами методики проектирования и реализации сложных программных комплексов.

Для достижения поставленной цели в курсе «Паттерное программирование» решаются задачи приобретения:

расширенных знаний об основных парадигмах объектно-ориентированного программирования;

представлений о объектной модели C++;

умений проектировать иерархию классов с использованием стандартных паттернов проектирования;

навыков написания программного кода с возможностями модификации и расширения.

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Паттерное программирование входит в часть учебного плана формируемую участниками образовательных отношений образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Объектно-ориентированное программирование

Объектно-ориентированное программное обеспечение

Программирование

Проектирование программных средств

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Подготовка к сдаче и сдача государственного экзамена

Выполнение и защита выпускной квалификационной работы

3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Паттерное программирование» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ПК-6 Способность к формализации и алгоритмизации поставленных задач, к написанию программного кода с использованием языков программирования, определения и манипулирования данными и оформлению программного кода в соответствии установленными требованиями	
ПК-6.1	Оценивает качество математической модели при формализации задачи предметной области
ПК-6.2	Оценивает качество разработанных алгоритмов для последующего кодирования
ПК-6.3	Оценивает выбор программных средств для программирования и манипулирования данными в соответствии установленными требованиями
ПК-7 Владеет способами разработки процедур интеграции программных модулей, компонент и верификации выпусков программного продукта, включая базы данных	
ПК-7.1	Оценивает выбор программных средств для разработки и верификации интеграционного слоя автоматизированных систем

4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 3 зачетных единиц 108 акад. часов, в том числе:

– контактная работа – 51,1 акад. часов:

– аудиторная – 48 акад. часов;

– внеаудиторная – 3,1 акад. часов

– самостоятельная работа – 21,2 акад. часов;

– подготовка к экзамену – 35,7 акад. часа

Форма аттестации - экзамен

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа студента	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код компетенции
		Лек.	лаб. зан.	практ. зан.				
1. Полиморфизм								
1.1 Раннее и позднее связывание. Таблица виртуальных функций	8	2	4		2	1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию	Проверка индивидуальных заданий	ПК-6.1, ПК-6.2, ПК-6.3, ПК-7.1
1.2 Абстрактные классы. Чистые виртуальные функции. Пример «Звездное небо»		2	4		3,2	1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию	Проверка индивидуальных заданий	ПК-6.1, ПК-6.2, ПК-6.3, ПК-7.1
Итого по разделу		4	8		5,2			
2. Множественное и виртуальное наследование								
2.1 Принцип множественного наследования. Область видимости класса при множественном наследовании	8	2	4		2	1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию	Проверка индивидуальных заданий	ПК-6.1, ПК-6.2, ПК-6.3, ПК-7.1

2.2 Виртуальное наследование		2	4		3	1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию	Проверка индивидуальных заданий	ПК-6.1, ПК-6.2, ПК-6.3, ПК-7.1
Итого по разделу		4	8		5			
3. Шаблоны (паттерны) проектирования								
3.1 Понятие шаблона проектирования. Каталог паттернов проектирования. Паттерн «Одиночка» (Singleton)	8	2	4		2	1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию	Проверка индивидуальных заданий	ПК-6.1, ПК-6.2, ПК-6.3, ПК-7.1
3.2 Паттерн «Стратегия» (Strategy)		2	4		3	1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию	Проверка индивидуальных заданий	ПК-6.1, ПК-6.2, ПК-6.3, ПК-7.1
3.3 Паттерн «Наблюдатель» (Observer)		2	4		3	1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию	Проверка индивидуальных заданий	ПК-6.1, ПК-6.2, ПК-6.3, ПК-7.1
3.4 Паттерн «Декоратор» (Decorator)		1	2		2	1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию	Проверка индивидуальных заданий	ПК-6.1, ПК-6.2, ПК-6.3, ПК-7.1
3.5 Паттерн Команда (Command)		1	2		1	1. Самостоятельное изучение учебной и научной литературы. 2. Подготовка к лабораторному занятию	Проверка индивидуальных заданий	ПК-6.1, ПК-6.2, ПК-6.3, ПК-7.1
Итого по разделу		8	16		11			
Итого за семестр		16	32		21,2		экзамен	
Итого по дисциплине		16	32		21,2		экзамен	

5 Образовательные технологии

1. Традиционные образовательные технологии, ориентированные на организацию образовательного процесса и предполагающую прямую трансляцию знаний от преподавателя к студенту.

Формы учебных занятий с использованием традиционных технологий:

Информационная лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами (монолог преподавателя).

Практическое занятие, посвященное освоению конкретных умений и навыков по предложенному алгоритму.

2. Технологии проблемного обучения – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности студентов.

Формы учебных занятий с использованием технологий проблемного обучения:

Проблемная лекция – изложение материала, предполагающее постановку проблемных и дискуссионных вопросов, освещение различных научных подходов, авторские комментарии, связанные с различными моделями интерпретации изучаемого материала.

3. Интерактивные технологии – организация образовательного процесса, которая предполагает активное и нелинейное взаимодействие всех участников, достижение на этой основе личностно-значимого для них образовательного результата.

Формы учебных занятий с использованием специализированных интерактивных технологий:

Лекция «обратной связи» – лекция-провокация (изложение материала с заранее запланированными ошибками), лекция-беседа, лекция-дискуссия, лекция-прессконференция.

Практическое занятие в форме практикума – организация учебной работы, направленная на решение комплексной учебно-познавательной задачи, требующей от студента применения как научно-теоретических знаний, так и практических навыков.

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 Учебно-методическое и информационное обеспечение дисциплины (модуля)

а) Основная литература:

1. Приемы объектно ориентированного проектирования. Паттерны проектирования. [Электронный ресурс] : справ. / Э. Гамма [и др.]. — Электрон. дан. — М. : ДМК Пресс, 2014. — 368 с. — Режим доступа: http://e.lanbook.com/book/1220 — Загл. с экрана.

2. Липпман, С. Язык программирования C++. Полное руководство. [Электронный ресурс] : рук. / С. Липпман, Ж. Лажойе. — Электрон. дан. — М. : ДМК Пресс, 2016. — 1105 с. — Режим доступа: http://e.lanbook.com/book/1216 — Загл. с экрана.

б) Дополнительная литература:

1. Страуструп, Б. Дизайн и эволюция C++. [Электронный ресурс] — Электрон. дан. — М. : ДМК Пресс, 2015. — 448 с. — Режим доступа:

http://e.lanbook.com/book/1222 — Загл. с экрана.

2. Аммерааль, Л. STL для программистов на C++. [Электронный ресурс] — Электрон. дан. — М. : ДМК Пресс, 2006. — 240 с. — Режим доступа: http://e.lanbook.com/book/1218 — Загл. с экрана.

в) Методические указания:

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

Наименование ПО	№ договора	Срок действия лицензии
MS Windows 7 Professional(для классов)	Д-1227-18 от 08.10.2018	11.10.2021
MS Office 2007 Professional	№ 135 от 17.09.2007	бессрочно
Borland Turbo C++	№112301 от 23.11.2005	бессрочно
MS Visual Studio 2013 Professional(для класса)	Д-1227-18 от 08.10.2018	11.10.2021
MS Visual Studio 2017 Community Edition	свободно распространяемое ПО	бессрочно

Профессиональные базы данных и информационные справочные системы

Название курса	Ссылка

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

Лекционная аудитория — мультимедийные средства хранения, передачи и представления информации.

Компьютерный класс — персональные компьютеры с компиляторами C++, пакетом Office, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета.

Аудитории для самостоятельной работы: компьютерные классы; читальные залы библиотеки — все классы УИТиАСУ с персональными компьютерами, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета.

Аудитории для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации — ауд. 282 и классы УИТиАСУ.

Помещения для самостоятельной работы обучающихся, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и наличием доступа в электрон-ную информационно-образовательную среду организации — классы УИТиАСУ.

Помещения для хранения и профилактического обслуживания учебного оборудования — ауд. 379.

Приложение 1

По дисциплине «Паттерное программирование» предусмотрена аудиторная и внеаудиторная самостоятельная работа обучающихся.

Аудиторная самостоятельная работа студентов предполагает решение контрольных задач на лабораторно-практических занятиях.

Примерные аудиторные контрольные работы (АКР):

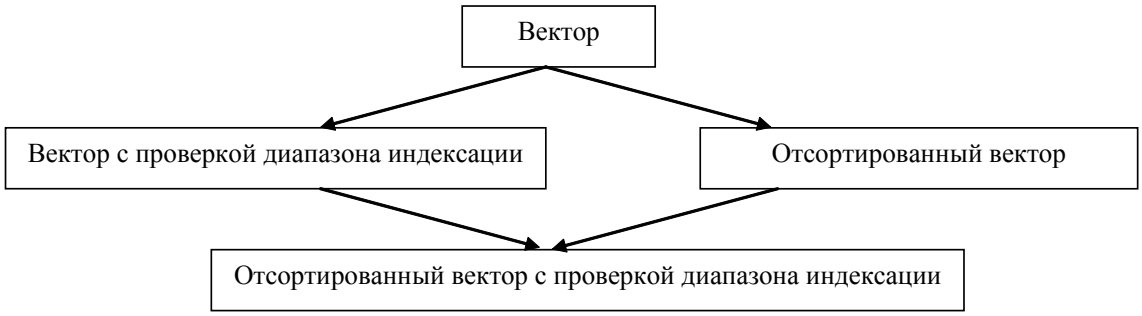
Раздел 1.

1. Реализовать класс «Длинное целое». Обеспечить возможность выполнения арифметических операций с экземплярами класса. Протестировать корректность программы на случайных числах.

2. Разработать информационную систему для моделирования геобиоценоза. Использовать полиморфизм. Обеспечить слабую связь между классом "Объект карты" и классом "Карта".

Раздел 2.

1. Реализовать иерархию классов согласно следующей схеме:



Раздел 3.

1. Спроектировать и реализовать иерархию классов для игровых персонажей и разных типов вооружения. Каждый персонаж в любой момент времени использует только один вид оружия, но может свободно менять оружие в ходе игры. Использовать паттерн Стратегия.

2. Промоделировать чат на основе паттерна Observer.

3. Адаптировать лекционный пример из темы паттерн «Декоратор» в соответствии с новыми требованиями: Теперь кофе можно заказать в маленькой, средней или большой чашке. Starbuzz считает размер порции неотъемлемой частью класса кофе, поэтому в класс Beverage были добавлены два новых метода: setSize() и getSize(). Стоимость дополнений также зависит от размера порции, так что, скажем, добавка сои должна стоить 10, 15 или 20 центов для маленькой, средней или большой порции соответственно.

```
#include <string>

#include <iostream>
```

```
using namespace std;

class Beverage

{

protected:

    string description;

public:

    virtual string getDescription()    {return description;}

    virtual double cost()=0;

};



class Espresso : public Beverage

{

public:

    Espresso()  {description="Espresso";}

    virtual double cost()   {return 1.99;}
```

```
};



class HouseBlend : public Beverage

{

public:

    HouseBlend()      {description="House Blend coffee";}

    virtual double cost()   {return 0.99;}

};



class DarkRoast : public Beverage

{

public:

    DarkRoast() {description="Dark Roast coffee";}

    virtual double cost()   {return 1.39;}
```

```
};



class CondimentDecorator : public Beverage

{

protected:

    Beverage *beverage;

};



class Soy : public CondimentDecorator

{

public:

    Soy(Beverage *b)  {beverage=b;}

    virtual string getDescription()    {return beverage->getDescription()+", Soy";}

    virtual double cost()             {return 0.20+beverage->cost();}

    ~Soy()                        {delete beverage;}
```

```
};



class Whip : public CondimentDecorator

{

public:

    Whip(Beverage *b) {beverage=b;}

    virtual string getDescription()    {return beverage->getDescription()+", Whip";}

    virtual double cost()            {return 0.15+beverage->cost();}

    ~Whip()                        {delete beverage;}

};



int _tmain(int argc, _TCHAR* argv[])

{

    Beverage *b=new Espresso;
```

```
    cout<<b->getDescription()<<", $"<<b->cost()<<endl;

    delete b;

     Beverage *b2=new HouseBlend;

    b2=new Soy(b2);

    b2=new Whip(b2);

    b2=new Whip(b2);

    cout<<b2->getDescription()<<", $"<<b2->cost()<<endl;

    delete b2;

 return 0;

}
```

4. Доработать лекционный пример из темы паттерн «Команда»: Добавить устройство — трехскоростной вентилятор и реализовать функцию отмены последней операции.

```
#include <iostream>

#include <string>

using namespace std;

 class Light
```

```
{
    string descr;
public:
    Light(string d)   {descr=d;}
    void on()   {cout<<descr<<": Light is on"<<endl;}
    void off()  {cout<<descr<<": Light is off"<<endl;}
};

class GarageDoor
{
public:
    void up()   {cout<<"Garage door is open"<<endl;}
    void down() {cout<<"Garage door is close"<<endl;}
};
```

```
class Stereo

{

public:

    void on()                {cout<<"Stereo is on"<<endl;}

    void off()               {cout<<"Stereo is off"<<endl;}

    void setCd()             {cout<<"Stereo is set for CD input"<<endl;}

    void setDvd()            {cout<<"Stereo is set for DVD input"<<endl;}

    void setRadio()          {cout<<"Stereo is set for radio"<<endl;}

    void setVolume(int v)    {cout<<"Stereo volume set to "<<v<<endl;}

};



class Command

{

public:
```

```
    virtual void execute()=0;

};



class NoCommand : public Command

{

public:

    virtual void execute() {}

};



class LightOnCommand : public Command

{

    Light *light;

public:

    LightOnCommand(Light *l)     {light=l;}

    virtual void execute()       {light->on();}
```

```
};

class LightOffCommand : public Command

{

    Light *light;

public:

    LightOffCommand(Light *l)    {light=l;}

    virtual void execute()       {light->off();}

};



class GarageDoorUpCommand : public Command

{

    GarageDoor *garageDoor;

public:

    GarageDoorUpCommand(GarageDoor *gd) {garageDoor=gd;}
```

```
    virtual void execute()          {garageDoor->up();}

};

class GarageDoorDownCommand : public Command

{

    GarageDoor *garageDoor;

public:

    GarageDoorDownCommand(GarageDoor *gd)   {garageDoor=gd;}

    virtual void execute()              {garageDoor->down();}

};



class StereoOnWithCDCommand : public Command

{

    Stereo *stereo;

public:

    StereoOnWithCDCommand(Stereo *s)   {stereo=s;}
```

```
    virtual void execute()
    {
        stereo->on();
        stereo->setCd();
        stereo->setVolume(11);
    }
};
class StereoOffCommand : public Command
{
    Stereo *stereo;
public:
    StereoOffCommand(Stereo *s)  {stereo=s;}
    virtual void execute()  {stereo->off();}
};
```

```
//Пульт

class RemoteControl

{

    int countButtons;

    Command **onCommands;

    Command **offCommands;

    NoCommand *noCommand;

public:

    RemoteControl(int c)

    {

        countButtons=c;

        onCommands=new Command*[c];

        offCommands=new Command*[c];

        noCommand=new NoCommand;
```

```
    for(int i=0; i<c; i++)
    {
        onCommands[i]=noCommand;
        offCommands[i]=noCommand;
    }
}
~RemoteControl()
{
    delete []onCommands;
    delete []offCommands;
    delete noCommand;
}
void setCommand(int slot, Command *onCommand, Command *offCommand)
{
```

```
        onCommands[slot]=onCommand;

        offCommands[slot]=offCommand;

    }

    void onButtonWasPushed(int slot)   {onCommands[slot]->execute();}

    void offButtonWasPushed(int slot)  {offCommands[slot]->execute();}

};



int _tmain(int argc, _TCHAR* argv[])

{

    RemoteControl rc(7);

    Light *livingRoomLight = new Light("Living Room");

    Light *kitchenLight = new Light("Kitchen");

    GarageDoor *garageDoor = new GarageDoor;

    Stereo *stereo = new Stereo;
```

```
    LightOnCommand *livingRoomLightOn = new LightOnCommand(livingRoomLight);

    LightOffCommand *livingRoomLightOff = new LightOffCommand(livingRoomLight);

    LightOnCommand *kitchenLightOn = new LightOnCommand(kitchenLight);

    LightOffCommand *kitchenLightOff = new LightOffCommand(kitchenLight);

    GarageDoorUpCommand *garageDoorUp = new GarageDoorUpCommand(garageDoor);

    GarageDoorDownCommand *garageDoorDown = new
GarageDoorDownCommand(garageDoor);

    StereoOnWithCDCommand *stereoOnWithCD = new StereoOnWithCDCommand(stereo);

    StereoOffCommand *stereoOff = new StereoOffCommand(stereo);



    rc.setCommand(0, livingRoomLightOn, livingRoomLightOff);

    rc.setCommand(1, kitchenLightOn, kitchenLightOff);

    rc.setCommand(2, garageDoorUp, garageDoorDown);

    rc.setCommand(3, stereoOnWithCD, stereoOff);
```

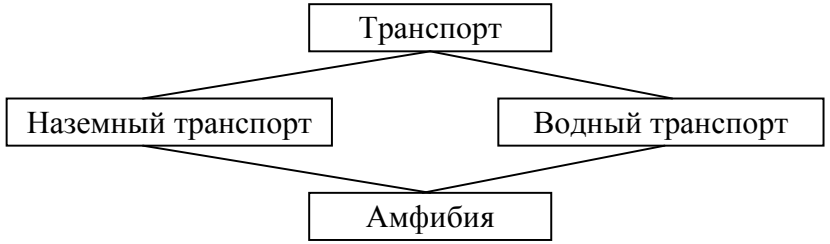
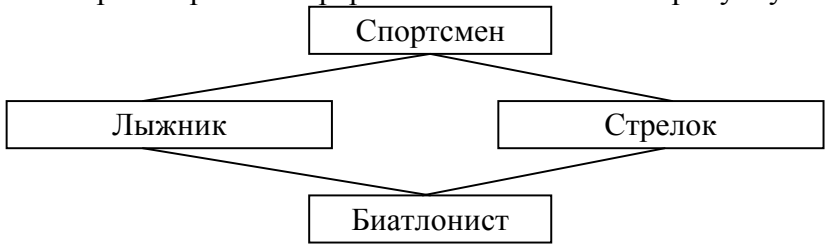
```
for(int i=0; i<7; i++)

{

    rc.onButtonWasPushed(i);

    rc.offButtonWasPushed(i);

}



delete livingRoomLightOn;

delete livingRoomLightOff;

delete kitchenLightOn;

delete kitchenLightOff;

delete garageDoorUp;

delete garageDoorDown;

delete stereoOnWithCD;

delete stereoOff;
```

```
    delete livingRoomLight;

    delete kitchenLight;

    delete garageDoor;

    delete stereo;



    return 0;

}
```

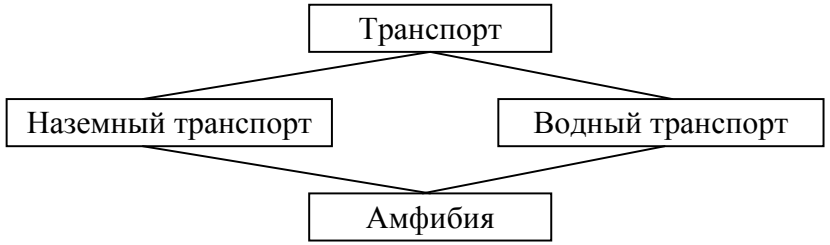
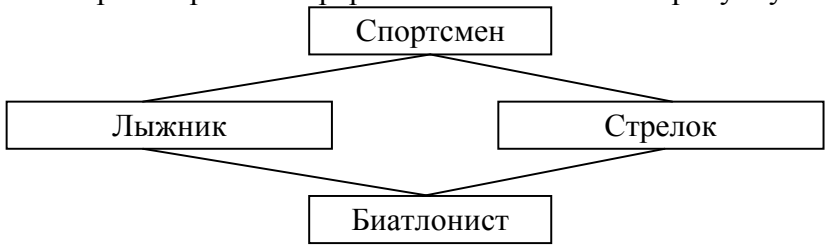
Приложение 2

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

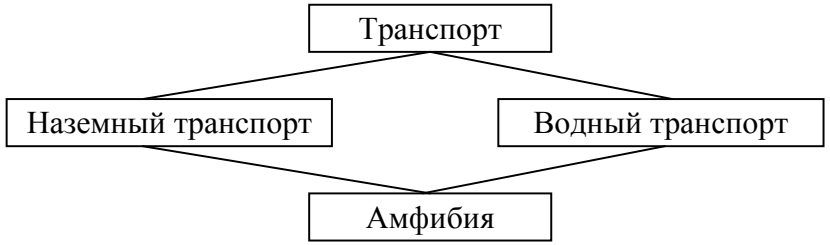
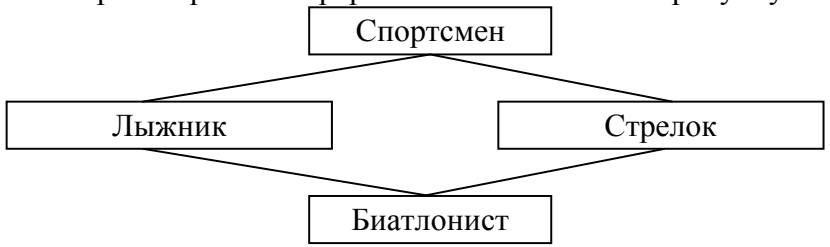
Код индикатора	Индикатор достижения компетенции	Оценочные средства
ПК-6: Способность к формализации и алгоритмизации поставленных задач, к написанию программного кода с использованием языков программирования, определения и манипулирования данными и оформлению программного кода в соответствии установленными требованиями		
ПК-6.1	Оценивает качество математической модели при формализации задачи предметной области	*Перечень теоретических вопросов* 1. Статическое или раннее связывание (static/early binding). Позднее/динамическое связывание (late/dynamic binding). Таблица виртуальных функций (virtual function table). 2. Виртуальные функции/методы (virtual functions/methods). Абстрактные классы (abstract classes) и чистые виртуальные функции (pure virtual functions). 3. Множественное наследование. Разрешение противоречий при наследовании одноименных членов класса. 4. Влияние множественного наследования на механизм виртуальных функций. Область видимости класса при множественном наследовании. 5. Виртуальное наследование. 6. Исключения и наследование. 7. Шаблоны (паттерны) проектирования. Основные понятия. Каталог паттернов проектирования. 8. Паттерн «Стратегия» (Strategy). 9. Паттерн «Наблюдатель» (Observer). 10. Паттерн «Декоратор» (Decorator). 11. Паттерн «Одиночка» (Singleton).

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		12. Паттерн «Команда» (Command) *Практические задания* 1. Спроектировать иерархию классов согласно рисунку:  В числе других должен быть определен метод способПередвижения(). 2. Спроектировать иерархию классов для моделирования сети Bluetooth. Сетевые устройства могут объединяться в «пикосеть» (piconet). В каждой пикосети одно устройство работает как master, а остальные как slave. Несколько пикосетей могут объединяться в «рассыпчатую» (scatternet) сеть. Для этого каждая пара пикосетей должна иметь общее устройство, которое будет master'ом в одной и slave'ом в другой 3. Спроектировать иерархию классов согласно рисунку:  В числе других должен быть определен метод используемыйИнвентарь(). 4. Спроектировать иерархию классов для расчета многослойной брони.

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		Для каждого материала известно, сколько энергии снаряда на миллиметр толщины он поглощает
ПК-6.2	Оценивает качество разработанных алгоритмов для последующего кодирования	*Перечень теоретических вопросов* 1. Статическое или раннее связывание (static/early binding). Позднее/динамическое связывание (late/dynamic binding). Таблица виртуальных функций (virtual function table). 2. Виртуальные функции/методы (virtual functions/methods). Абстрактные классы (abstract classes) и чистые виртуальные функции (pure virtual functions). 3. Множественное наследование. Разрешение противоречий при наследовании одноименных членов класса. 4. Влияние множественного наследования на механизм виртуальных функций. Область видимости класса при множественном наследовании. 5. Виртуальное наследование. 6. Исключения и наследование. 7. Шаблоны (паттерны) проектирования. Основные понятия. Каталог паттернов проектирования. 8. Паттерн «Стратегия» (Strategy). 9. Паттерн «Наблюдатель» (Observer). 10. Паттерн «Декоратор» (Decorator). 11. Паттерн «Одиночка» (Singleton). 12. Паттерн «Команда» (Command) *Практические задания* 1. Спроектировать иерархию классов согласно рисунку:

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		 В числе других должен быть определен метод способПередвижения(). 2. Спроектировать иерархию классов для моделирования сети Bluetooth. Сетевые устройства могут объединяться в «пикосеть» (piconet). В каждой пикосети одно устройство работает как master, а остальные как slave. Несколько пикосетей могут объединяться в «рассыпчатую» (scatternet) сеть. Для этого каждая пара пикосетей должна иметь общее устройство, которое будет master'ом в одной и slave'ом в другой 3. Спроектировать иерархию классов согласно рисунку:  В числе других должен быть определен метод используемыйИнвентарь(). 4. Спроектировать иерархию классов для расчета многослойной брони. Для каждого материала известно, сколько энергии снаряда на миллиметр толщины он поглощает
ПК-6.3	Оценивает выбор программных средств для программирования и манипулирования	*Перечень теоретических вопросов*

Код индикатора	Индикатор достижения компетенции	Оценочные средства
	данными в соответствии установленными требованиями	1. Статическое или раннее связывание (static/early binding). Позднее/динамическое связывание (late/dynamic binding). Таблица виртуальных функций (virtual function table). 2. Виртуальные функции/методы (virtual functions/methods). Абстрактные классы (abstract classes) и чистые виртуальные функции (pure virtual functions). 3. Множественное наследование. Разрешение противоречий при наследовании одноименных членов класса. 4. Влияние множественного наследования на механизм виртуальных функций. Область видимости класса при множественном наследовании. 5. Виртуальное наследование. 6. Исключения и наследование. 7. Шаблоны (паттерны) проектирования. Основные понятия. Каталог паттернов проектирования. 8. Паттерн «Стратегия» (Strategy). 9. Паттерн «Наблюдатель» (Observer). 10. Паттерн «Декоратор» (Decorator). 11. Паттерн «Одиночка» (Singleton). 12. Паттерн «Команда» (Command) *Практические задания* 1. Спроектировать иерархию классов согласно рисунку:

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		 В числе других должен быть определен метод способПередвижения(). 2. Спроектировать иерархию классов для моделирования сети Bluetooth. Сетевые устройства могут объединяться в «пикосеть» (piconet). В каждой пикосети одно устройство работает как master, а остальные как slave. Несколько пикосетей могут объединяться в «рассыпчатую» (scatternet) сеть. Для этого каждая пара пикосетей должна иметь общее устройство, которое будет master'ом в одной и slave'ом в другой 3. Спроектировать иерархию классов согласно рисунку:  В числе других должен быть определен метод используемыйИнвентарь(). 4. Спроектировать иерархию классов для расчета многослойной брони. Для каждого материала известно, сколько энергии снаряда на миллиметр толщины он поглощает
ПК-7: Владеет способами разработки процедур интеграции программных модулей, компонент и верификации выпусков		

Код индикатора	Индикатор достижения компетенции	Оценочные средства
программного продукта, включая базы данных		
ПК-7.1	Оценивает выбор программных средств для разработки и верификации интеграционного слоя автоматизированных систем	*Перечень теоретических вопросов* 1. Шаблоны (паттерны) проектирования. Основные понятия. Каталог паттернов проектирования. 2. Паттерн «Стратегия» (Strategy). 3. Паттерн «Наблюдатель» (Observer). 4. Паттерн «Декоратор» (Decorator). 5. Паттерн «Одиночка» (Singleton). *6.* Паттерн «Команда» (Command) *Практические задания* 1. Спроектировать иерархию классов для моделирования игры в шахматы. Учесть, что пешка может превращаться в фигуру. Обеспечить смену поведения без замены объекта. 2. Спроектировать иерархию классов для расчета гидравлического сопротивления участка трубопровода. Для каждого конструктивного элемента трубопровода известна характеристика потери давления, либо удельная (например, для прямого участка в Н/м), либо абсолютная (например, для поворота на 90 градусов в Н). 3. Спроектировать иерархию классов для моделирования игры в шахматы. Учесть, что пешка может превращаться в фигуру. Обеспечить смену поведения без замены объекта. 4. Спроектировать иерархию классов для расчета гидравлического сопротивления участка трубопровода. Для каждого конструктивного элемента трубопровода известна характеристика потери давления, либо удельная (например, для прямого участка в Н/м), либо

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		абсолютная (например, для поворота на 90 градусов в Н) 5. Спроектировать иерархию классов для моделирования штатного состава предприятия. Учесть возможность перевода работника с должности на должность. 6. Спроектировать иерархию классов для моделирования системы ролей пользователей в СУБД. Комбинация разрешений для объекта БД индивидуальна для каждой роли. Определить метод в классе ОбъектБД, возвращающий битовую маску разрешений для роли

б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Промежуточная аттестация по дисциплине «Паттерное программирование» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме экзамена.

Экзамен по данной дисциплине проводится в устной форме по экзаменационным билетам, каждый из которых включает 1 теоретический вопрос и одно практическое задание.

Показатели и критерии оценивания экзамена:

– на оценку **«отлично»** (5 баллов) – обучающийся демонстрирует высокий уровень сформированности компетенций, всестороннее, систематическое и глубокое знание учебного материала, свободно выполняет практические задания, свободно оперирует знаниями, умениями, применяет их в ситуациях повышенной сложности.

– на оценку **«хорошо»** (4 балла) – обучающийся демонстрирует средний уровень сформированности компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

– на оценку **«удовлетворительно»** (3 балла) – обучающийся демонстрирует пороговый уровень сформированности компетенций: в ходе контрольных мероприятий допускаются ошибки, проявляется отсутствие отдельных знаний, умений, навыков, обучающийся испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

– на оценку **«неудовлетворительно»** (2 балла) – обучающийся демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач.

– на оценку **«неудовлетворительно»** (1 балл) – обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач.